



PERÚ

Ministerio
de Educación

Libro de Sílabos

Carrera Técnica en Ciencia de la Computación

– 2021-I –

Lima: 7 de julio de 2020

Equipo de Trabajo

Ernesto Cuadros-Vargas (Editor)

Presidente de la Sociedad Peruana de Computación (SPC) 2001-2007, 2009
Miembro del *Steering Committee de ACM/IEEE-CS Computing Curricula for
Computer Science (CS2013)*

Miembro del *Steering Committee de ACM/IEEE-CS Computing Curricula
2020 (CS2020)*

email: ecuadros@spc.org.pe

<http://socios.spc.org.pe/ecuadros>

Índice general

Primer Semestre	5
1.1. CS111. Introducción a la Ciencia de la Computación	5
1.2. CS1D1. Estructuras Discretas I	12
1.3. MA100. Matemática I	17
1.4. FG101. Comunicación	20
1.5. FG102. Metodología del Estudio	22
1.6. ID101. Inglés I	26
Segundo Semestre	30
2.1. CS112. Ciencia de la Computación I	30
2.2. CS1D2. Estructuras Discretas II	38
2.3. MA101. Matemática II	42
2.4. FG106. Teatro	46
2.5. ID102. Inglés II	50
Tercer Semestre	54
3.1. CS113. Ciencia de la Computación II	54
3.2. CS221. Arquitectura de Computadores	65
3.3. CS2B1. Desarrollo Basado en Plataformas	73
3.4. MA203. Estadística y Probabilidades	78
3.5. FG203. Oratoria	80
3.6. ID201. Inglés III	82
Cuarto Semestre	86
4.1. CS210. Algoritmos y Estructuras de Datos	86
4.2. CS271. Gerenciamiento de Datos I	89
4.3. CS2S1. Sistemas Operativos	95
4.4. FG350. Liderazgo y Desempeño	103
4.5. ID202. Inglés IV	106
Quinto Semestre	110
5.1. CS212. Análisis y Diseño de Algoritmos	110
5.2. CS231. Redes y Comunicaciones	115
5.3. CS261. Sistemas Inteligentes	120
5.4. CS291. Ingeniería de Software I	129
5.5. CS2H1. Experiencia de Usuario (UX)	135
5.6. CB111. Física Computacional	142
5.7. ID203. Inglés V	146

Sexto Semestre	150
6.1. CS292. Ingeniería de Software II	150
6.2. CS311. Programación Competitiva	156
6.3. CS312. Estructuras de Datos Avanzadas	160
6.4. CS393. Sistemas de Infomación	164
6.5. CS3I1. Seguridad en Computación	166
6.6. CS3P1. Computación Paralela y Distribuída	176

Ministerio de Educación (MINEDU)
Sílabo 2021-I

1. CURSO

CS111. Introducción a la Ciencia de la Computación (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	4
2.2 Horas de teoría	:	2 (Semanal)
2.3 Horas de práctica	:	4 (Semanal)
2.4 Duración del periodo	:	16 semanas
2.5 Condición	:	Obligatorio
2.6 Modalidad	:	Presencial
2.7 Prerrequisitos	:	Ninguno

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Este es el primer curso en la secuencia de los cursos introductorios a la Ciencia de la Computación. En este curso se pretende cubrir los conceptos señalados por la Computing Curricula IEEE-CS/ACM 2013. La programación es uno de los pilares de la Ciencia de la Computación; cualquier profesional del Área, necesitará programar para concretizar sus modelos y propuestas. Este curso introducción a los participantes en los conceptos fundamentales de este arte. Lo tópicos incluyen tipos de datos, estructuras de control, funciones, listas, recursividad y la mecánica de la ejecución, prueba y depuración.

5. OBJETIVOS

- Introducir los conceptos fundamentales de programación.
- Desarrollar su capacidad de abstracción utilizar un lenguaje de programación.

6. COMPETENCIAS

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Usar**)
- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Usar**)
- d) Trabajar efectivamente en equipos para cumplir con un objetivo común. (**Usar**)
- a10) Evaluar y aplicar pensamiento computacional para resolver problemas cotidianos. (10)
- a11) Utilizar de forma eficiente estructuras de control condicionales, repetitivas, funciones, recursividad, ordenamiento y búsqueda. (11)
- b1) Identificar y aplicar de forma eficiente diversas estrategias algorítmicas y estructuras de datos para la solución de un problema dadas ciertas restricciones de espacio y tiempo. (1)
- d1) Desarrollo colaborativo de software utilizando repositorios de código y gestión de versiones (ej. Git, Bitbucket, SVN). (1)

7. TEMAS

Unidad 1: Historia (5)	
Competencias esperadas: a	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Pre-historia – El mundo antes de 1946. • Historia del hardware, software, redes. • Pioneros de la Computación. • Historia de Internet. 	<ul style="list-style-type: none"> • Identificar importantes tendencias en la historia del campo de la computación [Familiarizarse] • Identificar las contribuciones de varios pioneros en el campo de la computación [Familiarizarse] • Discutir el contexto histórico de los paradigmas de diversos lenguajes de programación [Familiarizarse] • Comparar la vida diaria antes y después de la llegada de los ordenadores personales y el Internet [Evaluar]
Lecturas : [BB19], [Gut13], [Zel10]	

Unidad 2: Sistemas de tipos básicos (2)	
Competencias esperadas: a	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Tipos como conjunto de valores junto con un conjunto de operaciones. <ul style="list-style-type: none"> – Tipos primitivos (p.e. números, booleanos) – Composición de tipos contruídos de otros tipos (p.e., registros, uniones, arreglos, listas, funciones, referencias) • Asociación de tipos de variables, argumentos, resultados y campos. • Tipo de seguridad y los errores causados por el uso de valores de manera incompatible dadas sus tipos previstos. 	<ul style="list-style-type: none"> • Tanto para tipo primitivo y un tipo compuesto, describir de manera informal los valores que tiene dicho tipo [Familiarizarse] • Para un lenguaje con sistema de tipos estático, describir las operaciones que están prohibidas de forma estática, como pasar el tipo incorrecto de valor a una función o método [Familiarizarse] • Describir ejemplos de errores de programa detectadas por un sistema de tipos [Familiarizarse] • Para múltiples lenguajes de programación, identificar propiedades de un programa con verificación estática y propiedades de un programa con verificación dinámica [Usar] • Usar tipos y mensajes de error de tipos para escribir y depurar programas [Usar] • Definir y usar piezas de programas (tales como, funciones, clases, métodos) que usan tipos genéricos, incluyendo para colecciones [Usar]
Lecturas : [Gut13], [Zel10]	

Unidad 3: Conceptos Fundamentales de Programación (9)**Competencias esperadas: a**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Sintaxis y semántica básica de un lenguaje de alto nivel.• Variables y tipos de datos primitivos (ej., números, caracteres, booleanos)• Expresiones y asignaciones.• Operaciones básicas I/O incluyendo archivos I/O.• Estructuras de control condicional e iterativas.• Paso de funciones y parámetros.• Concepto de recursividad.	<ul style="list-style-type: none">• Analiza y explica el comportamiento de programas simples que involucran estructuras fundamentales de programación variables, expresiones, asignaciones, E/S, estructuras de control, funciones, paso de parámetros, y recursividad [Evaluar]• Identifica y describe el uso de tipos de datos primitivos [Familiarizarse]• Escribe programas que usan tipos de datos primitivos [Usar]• Modifica y expande programas cortos que usen estructuras de control condicionales e iterativas así como funciones [Usar]• Diseña, implementa, prueba, y depura un programa que usa cada una de las siguientes estructuras de datos fundamentales: cálculos básicos, E/S simple, condicional estándar y estructuras iterativas, definición de funciones, y paso de parámetros [Usar]• Escribe un programa que usa E/S de archivos para brindar persistencia a través de ejecuciones múltiples [Usar]• Escoje estructuras de condición y repetición adecuadas para una tarea de programación dada [Familiarizarse]• Describe el concepto de recursividad y da ejemplos de su uso [Evaluar]• Identifica el caso base y el caso general de un problema basado en recursividad [Familiarizarse]
Lecturas : [Gut13], [Zel10]	

Unidad 4: Análisis Básico (2)**Competencias esperadas: a,b**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Diferencias entre el mejor, el esperado y el peor caso de un algoritmo.• Definición formal de la Notación Big O.• Clases de complejidad como constante, logarítmica, lineal, cuadrática y exponencial.• Uso de la notación Big O.• Análisis de algoritmos iterativos y recursivos.	<ul style="list-style-type: none">• Explique a que se refiere con “mejor”, “esperado” y “peor” caso de comportamiento de un algoritmo [Familiarizarse]• En el contexto de a algoritmos específicos, identifique las características de data y/o otras condiciones o suposiciones que lleven a diferentes comportamientos [Familiarizarse]• Indique la definición formal de Big O [Familiarizarse]• Use la notación formal de la Big O para dar límites superiores asintóticos en la complejidad de tiempo y espacio de los algoritmos [Usar]• Usar la notación formal Big O para dar límites de casos esperados en el tiempo de complejidad de los algoritmos [Usar]
Lecturas : [Gut13], [Zel10]	

Unidad 5: Algoritmos y Estructuras de Datos fundamentales (8)**Competencias esperadas: a,b**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Algoritmos numéricos simples, tales como el cálculo de la media de una lista de números, encontrar el mínimo y máximo.• Algoritmos de búsqueda secuencial y binaria.• Algoritmos de ordenamiento de peor caso cuadrático (selección, inserción)• Algoritmos de ordenamiento con peor caso o caso promedio en $O(N \lg N)$ (Quicksort, Heapsort, Mergesort)• Tablas Hash, incluyendo estrategias para evitar y resolver colisiones.• Árboles de búsqueda binaria:<ul style="list-style-type: none">– Operaciones comunes en árboles de búsqueda binaria como seleccionar el mínimo, máximo, insertar, eliminar, recorrido en árboles.• Grafos y algoritmos en grafos:<ul style="list-style-type: none">– Representación de grafos (ej., lista de adyacencia, matriz de adyacencia)– Recorrido en profundidad y amplitud• Montículos (Heaps)• Grafos y algoritmos en grafos:<ul style="list-style-type: none">– Problema de corte máximo y mínimo– Búsqueda local• Búsqueda de patrones y algoritmos de cadenas/texto (ej. búsqueda de subcadena, búsqueda de expresiones regulares, algoritmos de subsecuencia común más larga)	<ul style="list-style-type: none">• Implementar algoritmos numéricos básicos [Usar]• Implementar algoritmos de búsqueda simple y explicar las diferencias en sus tiempos de complejidad [Evaluar]• Ser capaz de implementar algoritmos de ordenamiento comunes cuadráticos y $O(N \log N)$ [Usar]• Describir la implementación de tablas hash, incluyendo resolución y el evitamiento de colisiones [Familiarizarse]• Discutir el tiempo de ejecución y eficiencia de memoria de los principales algoritmos de ordenamiento, búsqueda y hashing [Familiarizarse]• Discutir factores otros que no sean eficiencia computacional que influyan en la elección de algoritmos, tales como tiempo de programación, mantenibilidad, y el uso de patrones específicos de la aplicación en los datos de entrada [Familiarizarse]• Explicar como el balanceamiento del arbol afecta la eficiencia de varias operaciones de un arbol de búsqueda binaria [Familiarizarse]• Resolver problemas usando algoritmos básicos de grafos, incluyendo búsqueda por profundidad y búsqueda por amplitud [Usar]• Demostrar habilidad para evaluar algoritmos, para seleccionar de un rango de posibles opciones, para proveer una justificación por esa selección, y para implementar el algoritmo en un contexto en específico [Evaluar]• Describir la propiedad del heap y el uso de heaps como una implementación de colas de prioridad [Familiarizarse]• Resolver problemas usando algoritmos de grafos, incluyendo camino más corto de una sola fuente y camino más corto de todos los pares, y como mínimo un algoritmo de arbol de expansion minima [Usar]• Trazar y/o implementar un algoritmo de comparación de string [Usar]
Lecturas : [Gut13], [Zel10]	

Unidad 6: Algoritmos y Diseño (9)	
Competencias esperadas: a,b	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Conceptos y propiedades de los algoritmos <ul style="list-style-type: none"> – Comparación informal de la eficiencia de los algoritmos (ej., conteo de operaciones) • Rol de los algoritmos en el proceso de solución de problemas • Estrategias de solución de problemas <ul style="list-style-type: none"> – Funciones matemáticas iterativas y recursivas – Recorrido iterativo y recursivo en estructura de datos – Estrategias Divide y Conquistar • Conceptos y principios fundamentales de diseño <ul style="list-style-type: none"> – Abstracción – Descomposición de Program – Encapsulamiento y camuflaje de información – Separación de comportamiento y aplicación 	<ul style="list-style-type: none"> • Discute la importancia de los algoritmos en el proceso de solución de un problema [Familiarizarse] • Discute como un problema puede ser resuelto por múltiples algoritmos, cada uno con propiedades diferentes [Familiarizarse] • Crea algoritmos para resolver problemas simples [Usar] • Usa un lenguaje de programación para implementar, probar, y depurar algoritmos para resolver problemas simples [Usar] • Implementa, prueba, y depura funciones recursivas simples y sus procedimientos [Usar] • Determina si una solución iterativa o recursiva es la más apropiada para un problema [Evaluar] • Implementa un algoritmo de divide y vencerás para resolver un problema [Usar] • Aplica técnicas de descomposición para dividir un programa en partes más pequeñas [Usar] • Identifica los componentes de datos y el comportamiento de múltiples tipos de datos abstractos [Usar] • Implementa un tipo de dato abstracto coherente, con la menor pérdida de acoplamiento entre componentes y comportamientos [Usar] • Identifica las fortalezas y las debilidades relativas entre múltiples diseños e implementaciones de un problema [Evaluar]
Lecturas : [Gut13], [Zel10]	

Unidad 7: Métodos de Desarrollo (1)	
Competencias esperadas: a,b	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Entornos modernos de programación: <ul style="list-style-type: none"> – Búsqueda de código. – Programación usando librería de componentes y sus APIs. 	<ul style="list-style-type: none"> • Construir y depurar programas que utilizan las bibliotecas estándar disponibles con un lenguaje de programación elegido [Familiarizarse]
Lecturas : [Gut13], [Zel10]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

***** EVALUATION MISSING *****

10. BIBLIOGRAFÍA BÁSICA

- [BB19] J. Glenn Brookshear and Dennis Brylow. *Computer Science: An Overview*. Ed. by PEARSON. Global Edition. Pearson, 2019. ISBN: 1292263423. URL: <http://www.pearsonhighered.com/brookshear>.
- [Gut13] John V Guttag. . *Introduction To Computation And Programming Using Python*. MIT Press, 2013.
- [Zel10] John Zelle. *Python Programming: An Introduction to Computer Science*. Franklin, Beedle & Associates Inc, 2010.

Ministerio de Educación (MINEDU)
Sílabo 2021-I

1. CURSO

CS1D1. Estructuras Discretas I (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	4
2.2 Horas de teoría	:	2 (Semanal)
2.3 Horas de práctica	:	-
2.4 Duración del periodo	:	16 semanas
2.5 Condición	:	Obligatorio
2.6 Modalidad	:	Presencial
2.7 Prerrequisitos	:	Ninguno

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Las estructuras discretas proporcionan los fundamentos teóricos necesarios para la computación. Estos fundamentos no sólo son útiles para desarrollar la computación desde un punto de vista teórico como sucede En el curso de la teoría computacional, pero también es útil para la práctica de la informática; En particular en aplicaciones tales como verificación, Criptografía, métodos formales, etc.

5. OBJETIVOS

- Aplicar Correctamente conceptos de matemáticas finitas (conjuntos, relaciones, funciones) para representar datos de problemas reales.
- Modelar situaciones reales descritas en lenguaje natural, usando lógica proposicional y lógica predicada.
- Determinar las propiedades abstractas de las relaciones binarias.
- Elegir el método de demostración más apropiado para determinar la veracidad de una propuesta y construir argumentos matemáticos correctos.
- Interpretar soluciones matemáticas a un problema y determinar su fiabilidad, ventajas y desventajas.
- Expresar el funcionamiento de un circuito electrónico simple usando álgebra booleana.

6. COMPETENCIAS

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Usar**)
- j) Aplicar la base matemática, principios de algoritmos y la teoría de la CS en el modelamiento y diseño de sistemas. (**Usar**)
- a1) Aplicar técnicas de demostraciones (método directo, contrapositiva, inducción y contradicción) para demostrar propiedades en estructuras discretas y algoritmos. (**1**)
- a2) Usar de manera ordenada proposiciones lógicas. (**2**)
- a3) Aplicar técnicas de conteo en resolución de problemas computacionales. (**3**)
- j1) Solucionar problemas de recurrencia para simplificar la complejidad algorítmica. (**1**)
- j2) Aplicar teoría de grafos y árboles para la optimización y resolución de problemas. (**2**)

7. TEMAS

Unidad 1: Funciones, relaciones y conjuntos (22)**Competencias esperadas: a,j**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Conjuntos:<ul style="list-style-type: none">– Diagramas de Venn– Unión, intersección, complemento– Producto Cartesiano– Potencia de conjuntos– Cardinalidad de Conjuntos finitos• Relaciones:<ul style="list-style-type: none">– Reflexividad, simetria, transitividad– Relaciones de equivalencia– Relación de orden parcial y conjuntos parcialmente ordenados– Elementos extremos de un conjunto parcialmente ordenado• Funciones:<ul style="list-style-type: none">– Suryecciones, inyecciones, biyecciones– Inversas– Composición	<ul style="list-style-type: none">• Explicar con ejemplos la terminología básica de funciones, relaciones y conjuntos [Evaluar]• Realizar las operaciones asociadas con conjuntos, funciones y relaciones [Evaluar]• Relacionar ejemplos prácticos para conjuntos funciones o modelos de relación apropiados e interpretar la asociación de operaciones y terminología en contexto [Evaluar]
Lecturas : [Gri03], [Ros07], [Vel06]	

Unidad 2: Lógica básica (14)**Competencias esperadas: a, j**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Lógica proposicional.• Conectores lógicos.• Tablas de verdad.• Forma normal (conjuntiva y disyuntiva)• Validación de fórmula bien formada.• Reglas de inferencia proposicional (conceptos de modus ponens y modus tollens)• Logica de predicados:<ul style="list-style-type: none">– Cuantificación universal y existencial• Limitaciones de la lógica proposicional y de predicados (ej. problemas de expresividad)	<ul style="list-style-type: none">• Convertir declaraciones lógicas desde el lenguaje informal a expresiones de lógica proposicional y de predicados [Usar]• Aplicar métodos formales de simbolismo proposicional y lógica de predicados, como el cálculo de la validez de formulas y cálculo de formas normales [Usar]• Usar reglas de inferencia para construir demostraciones en lógica proposicional y de predicados [Usar]• Describir como la lógica simbólica puede ser usada para modelar situaciones o aplicaciones de la vida real, incluidos aquellos planteados en el contexto computacional como análisis de software (ejm. programas correctores), consulta de base de datos y algoritmos [Familiarizarse]• Aplicar demostraciones de lógica formal y/o informal, pero rigurosa, razonamiento lógico para problemas reales, como la predicción del comportamiento de software o solución de problemas tales como rompecabezas [Usar]• Describir las fortalezas y limitaciones de la lógica proposicional y de predicados [Usar]
Lecturas : [Ros07], [Gri03], [Vel06]	

Unidad 3: Técnicas de demostración (14)	
Competencias esperadas: a,j	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Nociones de implicancia, equivalencia, conversión, inversa, contrapositivo, negación, y contradicción • Estructura de pruebas matemáticas. • Demostración directa. • Refutar por contraejemplo. • Demostración por contradicción. • Inducción sobre números naturales. • Inducción estructural. • Inducción leve y fuerte (Ej. Primer y Segundo principio de la inducción) • Definiciones matemáticas recursivas. • Conjuntos bien ordenados. 	<ul style="list-style-type: none"> • Identificar la técnica de demostración utilizada en una demostración dada [Evaluar] • Describir la estructura básica de cada técnica de demostración (demostración directa, demostración por contradicción e inducción) descritas en esta unidad [Usar] • Aplicar las técnicas de demostración (demostración directa, demostración por contradicción e inducción) correctamente en la construcción de un argumento solido [Usar] • Determine que tipo de demostración es la mejor para un problema dado [Evaluar] • Explicar el paralelismo entre ideas matemáticas y/o inducción estructural para la recursión y definir estructuras recursivamente [Familiarizarse] • Explicar la relación entre inducción fuerte y débil y dar ejemplos del apropiado uso de cada uno [Evaluar] • Enunciar el principio del buen-orden y su relación con la inducción matemática [Familiarizarse]
Lecturas : [Ros07], [Vel06], [Sch12], [Vel06]	

Unidad 4: Representación de Datos (10)	
Competencias esperadas: a,j	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Representaciones numéricas: signo magnitud, punto flotante. • Representaciones de otros objetos: conjuntos, relaciones, funciones 	<ul style="list-style-type: none"> • Conocer las formas de representación numérica como signo magnitud y punto flotante. [Evaluar]. • Llevar a cabo operaciones aritméticas utilizando las distintas formas de representación. [Evaluar]. • Conocer el estándar de punto flotante IEEE-754 [Familiarizarse].
Lecturas : [Ros07], [Gri03], [Vel06]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

***** EVALUATION MISSING *****

10. BIBLIOGRAFÍA BÁSICA

- [Gri03] R. Grimaldi. *Discrete and Combinatorial Mathematics: An Applied Introduction*. 5 ed. Pearson, 2003.
- [Ros07] Kenneth H. Rosen. *Discrete Mathematics and Its Applications*. 7 ed. Mc Graw Hill, 2007.
- [Sch12] Edward R. Scheinerman. *Mathematics: A Discrete Introduction*. 3 ed. Brooks Cole, 2012.
- [Vel06] Daniel J. Velleman. *How to Prove It: A Structured Approach*. Ed. by Cambridge University Pres. 2nd. 2006. ISBN: 978-0521675994.

Ministerio de Educación (MINEDU)
Sílabo 2021-I

1. CURSO

MA100. Matemática I (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	: 5
2.2 Horas de teoría	: 2 (Semanal)
2.3 Horas de práctica	: -
2.4 Duración del periodo	: 16 semanas
2.5 Condición	: Obligatorio
2.6 Modalidad	: Presencial
2.7 Prerrequisitos	: Ninguno

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

The course aims to develop in students the skills to deal with models in science and engineering related to single variable differential calculus skills. In the course it is studied and applied concepts related to calculation limits, derivatives and integrals of real and vector functions of single real variables to be used as base and support for the study of new contents and subjects. Also seeks to achieve reasoning capabilities and applicability to interact with real-world problems by providing a mathematical basis for further professional development activities.

5. OBJETIVOS

- .
- .
- .

6. COMPETENCIAS

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Evaluar**)
- j) Aplicar la base matemática, principios de algoritmos y la teoría de la CS en el modelamiento y diseño de sistemas. (**Evaluar**)

■NoSpecificOutcomes■

7. TEMAS

Unidad 1: Números complejos (20)	
Competencias esperadas: C1	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• .• .	<ul style="list-style-type: none">• .• .
Lecturas : [Ste12], [ión14]	

Unidad 2: Functions of a single variable (10)	
Competencias esperadas: C20	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • . • . • . • . • . • . 	<ul style="list-style-type: none"> • . • . • . • . • . • .
Lecturas : [Ste12], [ión14]	

Unidad 3: Límites y derivadas (20)	
Competencias esperadas: C1	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • . • . • . • . • . 	<ul style="list-style-type: none"> • . • . • . • . • . • . • . • . • . • . • .
Lecturas : [Ste12], [ión14]	

Unidad 4: Integrales (22)	
Competencias esperadas: C20	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • . • . • . • . 	<ul style="list-style-type: none"> • . • . • . • . • . • . • . • . • . • . • .
Lecturas : [Ste12], [ión14]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

***** EVALUATION MISSING *****

10. BIBLIOGRAFÍA BÁSICA

[ión14] ROn Larson íon. *Calculus*. 10th. 2014.

[Ste12] James Stewart. *Calculus*. 7th. 2012.

Ministerio de Educación (MINEDU)
Sílabo 2021-I

1. CURSO

FG101. Comunicación (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	3
2.2 Horas de teoría	:	2 (Semanal)
2.3 Horas de práctica	:	-
2.4 Duración del periodo	:	16 semanas
2.5 Condición	:	Obligatorio
2.6 Modalidad	:	Presencial
2.7 Prerrequisitos	:	Ninguno

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Para lograr una eficaz comunicación en el ámbito personal y profesional, es prioritario el manejo adecuado de la Lengua en forma oral y escrita. Se justifica, por lo tanto, que los alumnos de la Universidad Católica San Pablo conozcan, comprendan y apliquen los aspectos conceptuales y operativos de su idioma, para el desarrollo de sus habilidades comunicativas fundamentales: Escuchar, hablar, leer y escribir. En consecuencia el ejercicio permanente y el aporte de los fundamentos contribuyen grandemente en la formación académica y, en el futuro, en el desempeño de su profesión

5. OBJETIVOS

- Desarrollar capacidades comunicativas a través de la teoría y práctica del lenguaje que ayuden al estudiante a superar las exigencias académicas del pregrado y contribuyan a su formación humanística y como persona humana.

6. COMPETENCIAS

- f) Comunicarse efectivamente con audiencias diversas. (**Usar**)
- h) Incorporarse a un proceso de aprendizaje profesional continuo. (**Usar**)
- n) Aplicar conocimientos de humanidades en su labor profesional. (**Usar**)

■NoSpecificOutcomes■

7. TEMAS

Unidad 1: Primera Unidad (16)	
Competencias esperadas: C17,C20	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • La comunicación, definición, relevancia. Elementos. Proceso. Funciones. Clasificación. Comunicación oral y escrita. • El lenguaje: definición. Características y funciones. Lengua: niveles. Sistema. Norma. Habla. El signo lingüístico: definición, características. • Multilingüismo en el Perú. Variaciones dialectales en el Perú. • La palabra: definición, clases y estructura. Los monemas: lexema y morfema. El morfema: clases. La etimología. • El Artículo académico: Definición, estructura, elección del tema, delimitación del tema. 	<ul style="list-style-type: none"> • Reconocer y valorar la comunicación como un proceso de comprensión e intercambio de mensajes, diferenciando sus elementos, funciones y clasificación [Usar]. • Analizar las características, funciones y elementos del lenguaje y de la lengua [Usar]. • Identificar las características del multilingüismo en el Perú, valorando su riqueza idiomática [Usar]. • Identificar las cualidades de la palabra y sus clases [Usar].
Lecturas : [Len10]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

***** EVALUATION MISSING *****

10. BIBLIOGRAFÍA BÁSICA

[Len10] Real Academia de la Lengua Española. *Nueva gramática de la lengua española, morfología y sintaxis*. Madrid, España: Ed. Espasa, 2010.

Ministerio de Educación (MINEDU)
Sílabo 2021-I

1. CURSO

FG102. Metodología del Estudio (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	: 3
2.2 Horas de teoría	: 2 (Semanal)
2.3 Horas de práctica	: -
2.4 Duración del periodo	: 16 semanas
2.5 Condición	: Obligatorio
2.6 Modalidad	: Presencial
2.7 Prerrequisitos	: Ninguno

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Los alumnos en formación profesional necesitan mejorar su actitud frente al trabajo y exigencia académicos. Además conviene que entiendan el proceso mental que se da en el ejercicio del estudio para lograr el aprendizaje; así sabrán dónde y cómo hacer los ajustes más convenientes a sus necesidades. Asimismo, requieren dominar variadas formas de estudiar, para que puedan seleccionar las estrategias más convenientes a su personal estilo de aprender y a la naturaleza de cada asignatura. De igual modo conocer y usar maneras de buscar información académica y realizar trabajos creativos de tipo académico formal, así podrán aplicarlos a su trabajo universitario, haciendo exitoso su esfuerzo.

5. OBJETIVOS

- Desarrollar en el estudiante actitudes y habilidades que promuevan la autonomía en el aprendizaje, el buen desempeño académico y su formación como persona y profesional.

6. COMPETENCIAS

- d) Trabajar efectivamente en equipos para cumplir con un objetivo común. (**Usar**)

- h) Incorporarse a un proceso de aprendizaje profesional continuo. (**Usar**)

- l) Desarrollar principios investigación en el área de computación con niveles de competitividad internacional. (**Familiarizarse**)

■NoSpecificOutcomes■

7. TEMAS

Unidad 1: (12)	
Competencias esperadas: C19, C24	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • El subrayado. • Toma de puntas. • La vocación, hábitos de la vida universitaria. • Interacción humana. • La voluntad como requisito para el aprendizaje. • La planificación y el tiempo 	<ul style="list-style-type: none"> • Analizar la documentación normativa de la Universidad valorando su importancia para la convivencia y desempeño académico. [Usar] • Comprender y valorar la exigencia de la vida universitaria como parte de la formación personal y profesional.[Usar] • Planificar adecuadamente el tiempo en función de sus metas personales y académicas.[Usar] • Elaborar un plan de mejora personal a partir del conocimiento de sí mismo.[Usar]
Lecturas : [Bibliograf\IeC {\'\i }a]	

Unidad 2: (12)	
Competencias esperadas: C19,C24	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Resumen. Notas al margen. Nemotecnias. • Procesos mentales: Simples, complejos. Fundamentos del aprendizaje significativo. • Los pasos o factores para el aprendizaje. Leyes del aprendizaje. Cuestionario de estilos de aprendizaje Identificación del estilo de aprendizaje personal • La lectura académica. Niveles de análisis de un texto: idea central, idea principal e ideas secundarias. El modelo de Meza de Vernet. • Exámenes: Preparación. Pautas y estrategias para antes, durante y después de un examen. Inteligencia emocional y exámenes. • Las fuentes de información. Aparato crítico: concepto y finalidad. Normas Vancouver. Referencias y citas. 	<ul style="list-style-type: none"> • Identificar los procesos mentales relacionándolos con el aprendizaje [Usar]. • Comprender el proceso del aprendizaje para determinar el estilo propio e incorporarlo en su actividad académica [Usar]. • Desarrollar estrategias para el análisis de textos potenciando la comprensión lectora [Usar]. • Diseñar un programa estratégico para afrontar con éxito los exámenes[Usar].
Lecturas : [Rod07], [Per10], [Qui07]	

Unidad 3: (12)	
Competencias esperadas: C24	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Los mapas conceptuales. Características y elementos. • Los derechos de autor y el plagio. Derechos personales o morales. Derechos patrimoniales. “Copyright”. • Autoestima, Inteligencia Emocional, Asertividad y Resiliencia. Conceptos, desarrollo y fortalecimiento. • Aparato crítico: Normas Vancouver. Aplicación práctica. • Generación de ideas. Estrategias para organizar las ideas, redacción y revisión. 	<ul style="list-style-type: none"> • Aplicar las técnicas de estudio atendiendo a sus particularidades y adecuándolas a las distintas situaciones que demanda el aprendizaje [Usar]. • Reconocer la importancia del respeto a la propiedad Intelectual [Usar]. • Reconocer la importancia de la Inteligencia Emocional, la conducta asertiva, la autoestima y la resiliencia valorándolas como fortalezas para el desempeño universitario [Usar].
Lecturas : [Chá11], [Vel99]	

Unidad 4: (12)	
Competencias esperadas: C19	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Cuadro Sinóptico. Los mapas mentales. Practicas con la temática del curso. • El método personal de estudio. • El aprendizaje cooperativo: definición, los grupos de estudio, organización, roles de los miembros. • Pautas para conformar grupos eficientes y armónicos. • El método personal de estudio.Reforzamiento de técnicas de estudio. • Presentación y exposición de trabajos de producción intelectual. • El debate y la argumentación. 	<ul style="list-style-type: none"> • Aplicar las técnicas de estudio atendiendo a sus particularidades y adecuándolas a las distintas situaciones que demanda el aprendizaje [Usar]. • Asumir el manejo de conductas y actitudes para el aprendizaje cooperativo y el desempeño en los equipos de trabajo [Usar]. • Formular un proyecto de método personal de estudio, de acuerdo a su estilo y necesidades, que incluya técnicas y estrategias [Usar].
Lecturas : [Rod07], [Chá11]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

***** EVALUATION MISSING *****

10. BIBLIOGRAFÍA BÁSICA

- [Chá11] A. Chávez. *Se necesita un tutor*. UCSP, 2011.
- [Per10] A.E. Perez. *Teoría del Derecho*. Editorial Madrid, 2010.
- [Qui07] V. Quintana. *El estudio Universitario y elementos de investigación científica*. Editorial universitaria, 2007.
- [Rod07] J. Rodríguez. *Guía para el método de estudio universitario*. Educa, 2007.
- [Vel99] Marco Flores Velazco. *Mapas conceptuales en el aula*. Ed. San Marcos, 1999.

Ministerio de Educación (MINEDU)
Sílabo 2021-I

1. CURSO

ID101. Inglés I (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	: 3
2.2 Horas de teoría	: -
2.3 Horas de práctica	: -
2.4 Duración del periodo	: 16 semanas
2.5 Condición	: Obligatorio
2.6 Modalidad	: Presencial
2.7 Prerrequisitos	: Ninguno

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Parte fundamental de la formación integral de un profesional es la habilidad de comunicarse en un idioma extranjero además del propio idioma nativo. No solamente amplía su horizonte cultural sino que permite una visión más humana y comprensiva de la vida de las personas. En el caso de los idiomas extranjeros, indudablemente el Inglés es el más práctico porque es hablado alrededor de todo el mundo. No hay país alguno donde éste no sea hablado. En las carreras relacionadas con los servicios al turista el Inglés es tal vez la herramienta práctica más importante que el alumno debe dominar desde el primer momento, como parte de su formación integral.

5. OBJETIVOS

- Conocer el idioma Inglés y su estructura gramatical.
- Identificar situaciones y emplear diálogos relacionados a ellas.

6. COMPETENCIAS

f) Comunicarse efectivamente con audiencias diversas. (**Usar**)

■NoSpecificOutcomes■

7. TEMAS

Unidad 1: Hello everybody (0)	
Competencias esperadas: C25	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Verbo To Be.• Oraciones Afirmativas, Negativas y Preguntas.• Expresiones Numéricas.• Objetos y Países.• Expresiones para saludar y hacer presentaciones.	<ul style="list-style-type: none">• Al terminar la primera unidad, cada uno de los alumnos, comprendiendo la gramática del tiempo presente es capaz de expresar una mayor cantidad de expresiones de tiempo y además usar oraciones con el verbo To Be para expresar situación y estado.• Que el alumno sea capaz de analizar y expresar ideas acerca de fechas y números en orden.
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 2: Meeting people (0)	
Competencias esperadas: C25	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Adjetivos Posesivos. • Expresiones para averiguar precios. • Expresiones de Posesión. • Vocabulario de Familia, Comidas y Bebidas. • Pedidos formales. • Cartas informales. 	<ul style="list-style-type: none"> • Al terminar la segunda unidad, los alumnos habiendo identificado la forma de expresar pedidos y hacer ofrecimientos en restaurantes los utilizan en situaciones varias. Explica y aplica vocabulario de comidas y bebidas.
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 3: The world of work (0)	
Competencias esperadas: C25	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Tiempo Presente Simple. Auxiliares. • Oraciones Afirmativas, Negativas y Preguntas. • Verbos comunes y Ocupaciones. • Indicaciones para expresar la hora. 	<ul style="list-style-type: none"> • Al terminar la tercera unidad, los alumnos habiendo reconocido las características del presente simple, lo utiliza para hacer descripciones de diversos tipos. Describen personas y lugares y dan indicaciones de dirección. Expresa la hora.
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 4: Take it easy (0)	
Competencias esperadas: C25	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Presente Simple 2. • Oraciones Afirmativas, Negativas y Preguntas. • Uso de Verbos de entretenimiento. • Tiempo Libre. • Las estaciones del año. • Expresiones de actividades sociales. 	<ul style="list-style-type: none"> • Al terminar la cuarta unidad, los alumnos habiendo identificado la idea de expresar ideas de acciones de tiempo libre en Presente Simple y Continuo. Expresan ideas de estaciones y actividades relacionadas.
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 5: Where do you live ? (0)	
Competencias esperadas: C25	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Uso There is/There are. • Oraciones con Preposiciones. • Expresiones de Cantidad. • Vocabulario de aviones y lugares. • Expresiones de indicaciones de dirección. 	<ul style="list-style-type: none"> • Al finalizar la quinta unidad, los alumnos, a partir de la comprensión del tiempo presente continuo, elaborarán oraciones utilizando ideas de ubicación y de lugar. Asimilarán además la necesidad de expresar objetos de uso común. Adquirirán vocabulario para describir las partes de una casa usan expresiones para pedir indicaciones de dirección.
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 6: Can you speak English? (0)	
Competencias esperadas: C25	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Can/cant. • Pasado del verbo To Be. Uso de Could. • Vocabulario de países e idiomas. • Expresiones para el uso del teléfono. • Redacción de cartas formales. • Lecturas. 	<ul style="list-style-type: none"> • Al finalizar la sexta unidad, los alumnos habiendo conocido los fundamentos del uso de auxiliares de modo, crearan oraciones aplicadas al contexto adecuado. Enfatizan la diferencia entre idiomas y nacionalidades. Describen sentimientos. Utilizan expresiones en el teléfono.
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 7: Then and now (0)	
Competencias esperadas: C25	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Pasado Simple. • Expresiones de tiempo pasado. • Vocabulario verbos regulares e irregulares. • Expresiones para describir el clima. • Redacción de párrafos descriptivos. • Ocasiones Especiales. 	<ul style="list-style-type: none"> • Al finalizar la séptima unidad, los alumnos habiendo conocido los fundamentos de la estructuración del Pasado Simple experimentan la necesidad de poder expresar este tipo de tiempo en acciones. Realizarán prácticas en contextos adecuados. Enfatizan la diferencia entre verbos irregulares y regulares. Describen acciones con verbos varios. Utilizan expresiones para describir el clima.
Lecturas : [SJ02], [Cam06], [Mac99]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

***** EVALUATION MISSING *****

10. BIBLIOGRAFÍA BÁSICA

- [Cam06] Cambridge. *Diccionario Inglés-Espanol Cambridge*. Editorial Oxford, 2006.
[Mac99] James MacGrew. *Focus on Grammar Basic*. Editorial Oxford, 1999.
[SJ02] Liz Soars and John. *American Headway N 2 Student Book*. Editorial Oxford, 2002.

Ministerio de Educación (MINEDU)
Sílabo 2021-I

1. CURSO

CS112. Ciencia de la Computación I (Obligatorio)

2. INFORMACIÓN GENERAL

- 2.1 Créditos** : 5
- 2.2 Horas de teoría** : 2 (Semanal)
- 2.3 Horas de práctica** : 4 (Semanal)
- 2.4 Duración del periodo** : 16 semanas
- 2.5 Condición** : Obligatorio
- 2.6 Modalidad** : Presencial
- 2.7 Prerrequisitos** : CS111. Introducción a la Ciencia de la Computación. (1^{er} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Este es el segundo curso en la secuencia de los cursos introductorios a la Ciencia de la Computación. El curso introducirá a los participantes en los diversos temas del área de computación como: algoritmos, estructuras de datos, ingeniería del software, etc.

5. OBJETIVOS

- Introducir al alumno a los fundamentos del paradigma de orientación a objetos, permitiendo asimilar los conceptos necesarios para desarrollar sistemas de información.

6. COMPETENCIAS

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**E**valuar)
- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**U**sar)
- d) Trabajar efectivamente en equipos para cumplir con un objetivo común. (**U**sar)
- a10) Evaluar y aplicar pensamiento computacional para resolver problemas cotidianos. (10)
- a11) Utilizar de forma eficiente estructuras de control condicionales, repetitivas, funciones, recursividad, ordenamiento y búsqueda. (11)
- b1) Identificar y aplicar de forma eficiente diversas estrategias algorítmicas y estructuras de datos para la solución de un problema dadas ciertas restricciones de espacio y tiempo. (1)
- d1) Desarrollo colaborativo de software utilizando repositorios de código y gestión de versiones (ej. Git, Bitbucket, SVN). (1)

7. TEMAS

Unidad 1: Visión General de los Lenguajes de Programación (1)	
Competencias esperadas: a	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Breve revisión de los paradigmas de programación.• Comparación entre programación funcional y programación imperativa.• Historia de los lenguajes de programación.	<ul style="list-style-type: none">• Discutir el contexto histórico de los paradigmas de diversos lenguajes de programación [Familiarizarse]
Lecturas : [Str13], [Dei17]	

Unidad 2: Sistemas de tipos básicos (2)**Competencias esperadas: a,b,i****Temas**

- Tipos como conjunto de valores junto con un conjunto de operaciones.
 - Tipos primitivos (p.e. números, booleanos)
 - Composición de tipos contruídos de otros tipos (p.e., registros, uniones, arreglos, listas, funciones, referencias)
- Declaración de modelos (enlace, visibilidad, alcance y tiempo de vida).
- Vista general del chequeo de tipos.

Objetivos de Aprendizaje

- Tanto para tipo primitivo y un tipo compuesto, describir de manera informal los valores que tiene dicho tipo [Familiarizarse]
- Para un lenguaje con sistema de tipos estático, describir las operaciones que están prohibidas de forma estática, como pasar el tipo incorrecto de valor a una función o método [Familiarizarse]
- Describir ejemplos de errores de programa detectadas por un sistema de tipos [Familiarizarse]
- Para múltiples lenguajes de programación, identificar propiedades de un programa con verificación estática y propiedades de un programa con verificación dinámica [Usar]
- Dar un ejemplo de un programa que no verifique tipos en un lenguaje particular y sin embargo no tenga error cuando es ejecutado [Familiarizarse]
- Usar tipos y mensajes de error de tipos para escribir y depurar programas [Usar]
- Explicar como las reglas de tipificación definen el conjunto de operaciones que legales para un tipo [Familiarizarse]
- Escribir las reglas de tipo que rigen el uso de un particular tipo compuesto [Usar]
- Explicar por qué indecidibilidad requiere sistemas de tipo para conservadoramente aproximar el comportamiento de un programa [Familiarizarse]
- Definir y usar piezas de programas (tales como, funciones, clases, métodos) que usan tipos genéricos, incluyendo para colecciones [Usar]
- Discutir las diferencias entre, genéricos (*generics*), subtipo y sobrecarga [Familiarizarse]
- Explicar múltiples beneficios y limitaciones de tipificación estática en escritura, mantenimiento y depuración de un software [Familiarizarse]

Lecturas : [Str13], [Dei17]

Unidad 3: Conceptos Fundamentales de Programación (6)**Competencias esperadas: a,b,i**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Sintaxis y semántica básica de un lenguaje de alto nivel.• Variables y tipos de datos primitivos (ej., numeros, caracteres, booleanos)• Expresiones y asignaciones.• Operaciones básicas I/O incluyendo archivos I/O.• Estructuras de control condicional e iterativas.• Paso de funciones y parámetros.	<ul style="list-style-type: none">• Analiza y explica el comportamiento de programas simples que involucran estructuras fundamentales de programación variables, expresiones, asignaciones, E/S, estructuras de control, funciones, paso de parámetros, y recursividad [Evaluar]• Identifica y describe el uso de tipos de datos primitivos [Familiarizarse]• Escribe programas que usan tipos de datos primitivos [Usar]• Modifica y expande programas cortos que usen estructuras de control condicionales e iterativas así como funciones [Usar]• Diseña, implementa, prueba, y depura un programa que usa cada una de las siguientes estructuras de datos fundamentales: cálculos básicos, E/S simple, condicional estándar y estructuras iterativas, definición de funciones, y paso de parámetros [Usar]• Escribe un programa que usa E/S de archivos para brindar persistencia a través de ejecuciones múltiples [Usar]• Escoje estructuras de condición y repetición adecuadas para una tarea de programación dada [Evaluar]• Describe el concepto de recursividad y da ejemplos de su uso [Familiarizarse]• Identifica el caso base y el caso general de un problema basado en recursividad [Evaluar]
Lecturas : [Str13], [Dei17]	

Unidad 4: Programación orientada a objetos (10)	
Competencias esperadas: a,b,i	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Diseño orientado a objetos: <ul style="list-style-type: none"> – Descomposición en objetos que almacenan estados y poseen comportamiento – Diseño basado en jerarquía de clases para modelamiento • Lenguajes orientados a objetos para la encapsulación: <ul style="list-style-type: none"> – privacidad y la visibilidad de miembros de la clase – Interfaces revelan único método de firmas – clases base abstractas • Definición de las categorías, campos, métodos y constructores. • Las subclasses, herencia y método de alteración temporal. • Subtipificación: <ul style="list-style-type: none"> – Polimorfismo artículo Subtipo; upcasts implícitos en lenguajes con tipos. – Noción de reemplazo de comportamiento: los subtipos de actuar como supertipos. – Relación entre subtipos y la herencia. • Uso de colección de clases, iteradores, y otros componentes de la librería estándar. • Asignación dinámica: definición de método de llamada. 	<ul style="list-style-type: none"> • Diseñar e implementar una clase [Usar] • Usar subclase para diseñar una jerarquía simple de clases que permita al código ser reusable por diferentes subclasses [Usar] • Razonar correctamente sobre el flujo de control en un programa mediante el envío dinámico [Usar] • Comparar y contrastar (1) el enfoque procedurador/funcional- definiendo una función por cada operación con el cuerpo de la función proporcionando un caso por cada variación de dato - y (2) el enfoque orientado a objetos - definiendo una clase por cada variación de dato con la definición de la clase proporcionando un método por cada operación. Entender ambos enfoques como una definición de variaciones y operaciones de una matriz [Evaluar] • Explicar la relación entre la herencia orientada a objetos (código compartido y <i>overriding</i>) y subtipificación (la idea de un subtipo es ser utilizable en un contexto en el que espera al supertipo) [Familiarizarse] • Usar mecanismos de encapsulación orientada a objetos, tal como interfaces y miembros privados [Usar] • Definir y usar iteradores y otras operaciones sobre agregaciones, incluyendo operaciones que tienen funciones como argumentos, en múltiples lenguajes de programación, seleccionar la forma más natural por cada lenguaje [Usar]
Lecturas : [Str13], [Dei17]	

Unidad 5: Algoritmos y Diseño (3)**Competencias esperadas: a,b,i**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">● Estrategias de solución de problemas<ul style="list-style-type: none">– Funciones matemáticas iterativas y recursivas– Recorrido iterativo y recursivo en estructura de datos– Estrategias Divide y Conquistar● Rol de los algoritmos en el proceso de solución de problemas● Estrategias de solución de problemas<ul style="list-style-type: none">– Funciones matemáticas iterativas y recursivas– Recorrido iterativo y recursivo en estructura de datos– Estrategias Divide y Conquistar● Conceptos y principios fundamentales de diseño<ul style="list-style-type: none">– Abstracción– Descomposición de Program– Encapsulamiento y camuflaje de información– Separación de comportamiento y aplicación	<ul style="list-style-type: none">● Discute la importancia de los algoritmos en el proceso de solución de un problema [Familiarizarse]● Discute como un problema puede ser resuelto por múltiples algoritmos, cada uno con propiedades diferentes [Familiarizarse]● Crea algoritmos para resolver problemas simples [Usar]● Usa un lenguaje de programación para implementar, probar, y depurar algoritmos para resolver problemas simples [Usar]● Implementa, prueba, y depura funciones recursivas simples y sus procedimientos [Usar]● Determina si una solución iterativa o recursiva es la más apropiada para un problema [Evaluar]● Implementa un algoritmo de divide y vencerás para resolver un problema [Usar]● Aplica técnicas de descomposición para dividir un programa en partes más pequeñas [Usar]● Identifica los componentes de datos y el comportamiento de múltiples tipos de datos abstractos [Usar]● Implementa un tipo de dato abstracto coherente, con la menor pérdida de acoplamiento entre componentes y comportamientos [Usar]● Identifica las fortalezas y las debilidades relativas entre múltiples diseños e implementaciones de un problema [Evaluar]
Lecturas : [Str13], [Dei17]	

Unidad 6: Estrategias Algorítmicas (3)	
Competencias esperadas: a,b,i	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Algoritmos de fuerza bruta. • Algoritmos voraces. • Divide y vencerás. • Backtracking recursivo. • Programación Dinámica. 	<ul style="list-style-type: none"> • Para cada una de las estrategias (fuerza bruta, algoritmo goloso, divide y vencerás, recursividad en reversa y programación dinámica), identifica un ejemplo práctico en el cual se pueda aplicar [Familiarizarse] • Utiliza un enfoque voraz para resolver un problema específico y determina si la regla escogida lo guía a una solución óptima [Evaluar] • Usa un algoritmo de divide-y-vencerás para resolver un determinado problema [Usar] • Usa recursividad en reversa a fin de resolver un problema como en el caso de recorrer un laberinto [Usar] • Usa programación dinámica para resolver un problema determinado [Usar] • Determina el enfoque algorítmico adecuado para un problema [Evaluar] • Describe varios métodos basados en heurísticas para resolver problemas [Familiarizarse]
Lecturas : [Str13], [Dei17]	

Unidad 7: Análisis Básico (2)	
Competencias esperadas: a,b,i	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Diferencias entre el mejor, el esperado y el peor caso de un algoritmo. 	<ul style="list-style-type: none"> • Explique a que se refiere con “mejor”, “esperado” y “peor” caso de comportamiento de un algoritmo [Familiarizarse]
Lecturas : [Str13], [Dei17]	

Unidad 8: Algoritmos y Estructuras de Datos fundamentales (6)	
Competencias esperadas: a,b,i	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Algoritmos numéricos simples, tales como el cálculo de la media de una lista de números, encontrar el mínimo y máximo. • Algoritmos de búsqueda secuencial y binaria. • Algoritmos de ordenamiento de peor caso cuadrático (selección, inserción) • Algoritmos de ordenamiento con peor caso o caso promedio en $O(N \lg N)$ (Quicksort, Heapsort, Mergesort) 	<ul style="list-style-type: none"> • Implementar algoritmos numéricos básicos [Usar] • Implementar algoritmos de búsqueda simple y explicar las diferencias en sus tiempos de complejidad [Evaluar] • Ser capaz de implementar algoritmos de ordenamiento comunes cuadráticos y $O(N \log N)$ [Usar] • Discutir el tiempo de ejecución y eficiencia de memoria de los principales algoritmos de ordenamiento, búsqueda y hashing [Familiarizarse] • Discutir factores otros que no sean eficiencia computacional que influyan en la elección de algoritmos, tales como tiempo de programación, mantenibilidad, y el uso de patrones específicos de la aplicación en los datos de entrada [Familiarizarse] • Explicar como el balanceamiento del arbol afecta la eficiencia de varias operaciones de un arbol de búsqueda binaria [Familiarizarse] • Demostrar habilidad para evaluar algoritmos, para seleccionar de un rango de posibles opciones, para proveer una justificación por esa selección, y para implementar el algoritmo en un contexto en específico [Evaluar] • Trazar y/o implementar un algoritmo de comparación de string [Usar]
Lecturas : [Str13], [Dei17]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

***** EVALUATION MISSING *****

10. BIBLIOGRAFÍA BÁSICA

[Dei17] Deitel & Deitel. *C++17 - The Complete Guide*. 10th. Pearson, 2017. ISBN: 978-0201734843.

[Str13] Bjarne Stroustrup. *The C++ Programming Language*. 4th. Addison-Wesley, 2013. ISBN: 978-0-321-56384-2.

Ministerio de Educación (MINEDU)
Sílabo 2021-I

1. CURSO

CS1D2. Estructuras Discretas II (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	4
2.2 Horas de teoría	:	2 (Semanal)
2.3 Horas de práctica	:	2 (Semanal)
2.4 Duración del periodo	:	16 semanas
2.5 Condición	:	Obligatorio
2.6 Modalidad	:	Presencial
2.7 Prerrequisitos	:	CS1D1. Estructuras Discretas I. (1 ^{er} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Para entender las técnicas computacionales avanzadas, los estudiantes deberán tener un fuerte conocimiento de las diversas estructuras discretas, estructuras que serán implementadas y usadas en laboratorio en el lenguaje de programación.

5. OBJETIVOS

- Que el alumno sea capaz de modelar problemas de ciencia de la computación usando grafos y árboles relacionados con estructuras de datos.
- Que el alumno aplique eficientemente estrategias de recorrido para poder buscar datos de una manera óptima.
- Que el alumno utilice las diversas técnicas de conteo para resolver problemas computacionales.

6. COMPETENCIAS

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Usar**)
- j) Aplicar la base matemática, principios de algoritmos y la teoría de la CS en el modelamiento y diseño de sistemas. (**Usar**)
- a3) Aplicar técnicas de conteo en resolución de problemas computacionales. (**3**)
- a9) Utilizar técnicas matemáticas que permitan acotar sumatorias y resolver recurrencias que reflejan los costos computacionales de un algoritmo. (**9**)
- a13) Utilizar definiciones de teoría de conteo para resolver problemas de ordenamiento o selección en un conjunto de elementos únicos y repetidos. (**13**)
- a14) Resolver problemas de conteo usando funciones generatrices. (**14**)
- j1) Solucionar problemas de recurrencia para simplificar la complejidad algorítmica. (**1**)
- j2) Aplicar teoría de grafos y árboles para la optimización y resolución de problemas. (**2**)

7. TEMAS

Unidad 1: Lógica Digital y Representación de Datos (10)**Competencias esperadas: a,b,i**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Retículo: Tipos y propiedades.• Álgebras booleanas.• Funciones y expresiones booleanas.• Representación de las funciones booleanas: Disjuntiva normal y conjuntiva normal.• Puertas Lógicas.• Minimización del Circuito.	<ul style="list-style-type: none">• Explicar la importancia del álgebra booleana como una unificación de la teoría de conjuntos y la lógica proposicional [Evaluar].• Explicar las estructuras algebraicas del retículo y sus tipos [Evaluar].• Explicar la relación entre el retículo y el conjunto de ordenadas y el uso prudente para demostrar que un conjunto es un retículo [Evaluar].• Explicar las propiedades que satisfacen un álgebra booleana [Evaluar].• Demostrar si una terna formada por un conjunto y dos operaciones internas es o no Álgebra booleana [Evaluar].• Encuentra las formas canónicas de una función booleana [Evaluar].• Representar una función booleana como un circuito booleano usando puertas lógicas [Evaluar].• Minimizar una función booleana [Evaluar].

Lecturas : [Ros07], [Gri03]

Unidad 2: Fundamentos de conteo (40)**Competencias esperadas: a**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Técnicas de Conteo:<ul style="list-style-type: none">– Conteo y cardinalidad de un conjunto– Regla de la suma y producto– Principio de inclusión-exclusión– Progresión geométrica y aritmética• Principio de las casillas.• Permutaciones y combinaciones:<ul style="list-style-type: none">– Definiciones básicas– Identidad de Pascal– Teorema del binomio• Resolviendo relaciones de recurrencia:<ul style="list-style-type: none">– Un ejemplo de una relación de recurrencia simple, como los números de Fibonacci– Otras ejemplos, mostrando una variedad de soluciones• Aritmetica modular basica	<ul style="list-style-type: none">• Aplicar argumentos de conteo, incluyendo las reglas del producto y de la suma, principio de inclusión-exclusión y progresiones aritméticas/geométricas [Familiarizarse]• Aplicar el principio de las casillas en el contexto de una demostración formal [Familiarizarse]• Calcular permutaciones y combinaciones en un conjunto, e interpreta su significado en el contexto de una aplicación en particular [Familiarizarse]• Mapear aplicaciones del mundo real a formalismos de conteo adecuados, como el determinar el número de formas de acomodar a un conjunto de personas alrededor de una mesa, sujeto a restricciones en la disposición de los asientos, o en el número de maneras de determinar ciertas manos en juegos de cartas (ejm. una casa llena) [Familiarizarse]• Resolver una variedad de relaciones de recurrencia básicas [Familiarizarse]• Analizar un problema para determinar las relaciones de recurrencia implícitas [Familiarizarse]• Realizar cálculos que involucran aritmética modular [Familiarizarse]
Lecturas : [Gri03]	

Unidad 3: Árboles y Grafos (40)	
Competencias esperadas: a	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Árboles. <ul style="list-style-type: none"> – Propiedades – Estrategias de recorrido • Grafos no dirigidos • Grafos dirigidos • Grafos ponderados • Árboles de expansión/bosques. • Isomorfismo en grafos. 	<ul style="list-style-type: none"> • Ilustrar mediante ejemplos la terminología básica de teoría de grafos, y de alguna de las propiedades y casos especiales de cada tipo de grafos/árboles [Familiarizarse] • Demostrar diversos métodos de recorrer árboles y grafos, incluyendo recorridos pre, post e inorden de árboles [Familiarizarse] • Modelar una variedad de problemas del mundo real en ciencia de la computación usando formas adecuadas de grafos y árboles, como son la representación de una topología de red o la organización jerárquica de un sistema de archivos [Familiarizarse] • Demostrar como los conceptos de grafos y árboles aparecen en estructuras de datos, algoritmos, técnicas de prueba (inducción estructurada), y conteos [Familiarizarse] • Explicar como construir un árbol de expansión de un grafo [Familiarizarse] • Determinar si dos grafos son isomorfos [Familiarizarse]
Lecturas : [Joh99]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

***** EVALUATION MISSING *****

10. BIBLIOGRAFÍA BÁSICA

- [Gri03] R. Grimaldi. *Discrete and Combinatorial Mathematics: An Applied Introduction*. 5 ed. Pearson, 2003.
- [Joh99] Richard Johnsonbaugh. *Matemáticas Discretas*. Prentice Hall, México, 1999.
- [Ros07] Kenneth H. Rosen. *Discrete Mathematics and Its Applications*. 7 ed. Mc Graw Hill, 2007.

Ministerio de Educación (MINEDU)
Sílabo 2021-I

1. CURSO

MA101. Matemática II (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	4
2.2 Horas de teoría	:	2 (Semanal)
2.3 Horas de práctica	:	-
2.4 Duración del periodo	:	16 semanas
2.5 Condición	:	Obligatorio
2.6 Modalidad	:	Presencial
2.7 Prerrequisitos	:	MA100. Matemática I. (1 ^{er} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

El curso desarrolla en los estudiantes las habilidades para manejar modelos de habilidades de ingeniería y ciencia. En la primera parte Del curso un estudio de las funciones de varias variables, derivadas parciales, integrales múltiples y una Introducción a campos vectoriales. Luego el estudiante utilizará los conceptos básicos de cálculo para modelar y resolver ecuaciones diferenciales ordinarias utilizando técnicas como las transformadas de Laplace y las series de Fourier.

5. OBJETIVOS

- Aplicar reglas de derivación y diferenciación parcial en funciones de varias variables.
- Aplicar técnicas para el cálculo de integrales múltiples.
- Comprender y utilizar los conceptos de cálculo vectorial.
- Comprender la importancia de las series.
- Identificar y resolver ecuaciones diferenciales de primer orden y sus aplicaciones en problemas químicos y físicos.

6. COMPETENCIAS

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Evaluar**)
- j) Aplicar la base matemática, principios de algoritmos y la teoría de la CS en el modelamiento y diseño de sistemas. (**Evaluar**)

■NoSpecificOutcomes■

7. TEMAS

Unidad 1: Multi-Variable Function Differential (24)	
Competencias esperadas: C1,C20	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Concepto de funciones multi-variables. • Derivados Direccionales • Línea tangente, plano normal a línea de curva y plano tangente, línea normal a un plano de curva. Conocer para calcular sus ecuaciones. • Concepto de valor extremo y valor extremo condicional de funciones multi-variables. • Problemas de aplicación tales como modelización de la producción total de un sistema económico, velocidad del sonido a través del océano, optimización del espesante, etc. 	<ul style="list-style-type: none"> • Comprender el concepto de funciones multi-variables. • Dominar el concepto y método de cálculo de la derivada direccional y gradiente de la guía. • Dominar el método de cálculo de la derivada parcial de primer orden y de segundo orden de las funciones compuestas. • DomEntender línea tangente, plano normal a línea de curva y plano tangente, línea normal a un plan de curva. Saber calcular sus ecuaciones.inar el método de cálculo de las derivadas parciales para funciones implícitas. • Entender línea tangente, plano normal a línea de curva y plano tangente, línea normal a un plan de curva. Saber calcular sus ecuaciones. • Aprenda el concepto de valor extremo y valor extremo condicional de funciones multi-variables; Saber para averiguar el valor extremo de la función binaria. • Ser capaz de resolver problemas de aplicaciones simples.
Lecturas : [Ste12], [Zil13]	

Unidad 2: Multi-Variable function Integral (12)	
Competencias esperadas: C1,C20	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Integral doble, integral triple y naturaleza de la integral múltiple. • Método de doble integral • Línea integral • La Divergencia, Rotación y Laplaciano 	<ul style="list-style-type: none"> • Entender la integral doble, integral triple, y entender la naturaleza de la integral múltiple. • Dominar el método de cálculo de la integral doble (coordenadas cartesianas, coordenadas polares), la integral triple (coordenadas cartesianas, coordenadas cilíndricas, coordenadas esféricas). • Entender el concepto de línea Integral, sus propiedades y relaciones. • Saber calcular la integral de línea. • Dominar el cálculo de la rotación, la divergencia y Laplacian.
Lecturas : [Ste12], [Zil13]	

Unidad 3: Series (24)	
Competencias esperadas: C1,C20	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Serie convergente. • Serie Taylor y MacLaurin. • Funciones ortogonales. 	<ul style="list-style-type: none"> • Dominio del cálculo si la serie es convergente, y si es convergente, encontrar la suma de la serie tratando de encontrar el radio de convergencia y el intervalo de convergencia de una serie de potencia. • Representa una función como una serie de potencias y encuentra la serie de Taylor y MacLaurin para estimar los valores de las funciones con la precisión deseada. • Entender los conceptos de funciones ortogonales y la expansión de una función dada f para encontrar su serie de Fourier.
Lecturas : [Ste12], [Zil13]	

Unidad 4: Ordinary Differential Equations (30)	
Competencias esperadas: C1,C20	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Concepto de ecuaciones diferenciales • Métodos para resolver ecuaciones diferenciales • Métodos para resolver las ecuaciones diferenciales lineales de segundo orden • Ecuaciones diferenciales ordinarias lineales de orden superior • Problemas de aplicaciones con las transformaciones de Laplace 	<ul style="list-style-type: none"> • Comprender ecuaciones diferenciales, soluciones, orden, solución general, condiciones iniciales y soluciones especiales, etc. • Dominar el método de cálculo para las variables ecuación separable y ecuaciones lineales de primer orden. Conocido para resolver la ecuación homogénea y las ecuaciones de Bernoulli (Bernoulli); Entender la sustitución de la variable para resolver la ecuación. • Diminuo para resolver ecuaciones diferenciales totales. • Ser capaz de utilizar el método de orden reducido para resolver ecuaciones. • Comprender la estructura de la ecuación diferencial lineal de segundo orden. • Dominio del cálculo para las ecuaciones diferenciales lineales homogéneas de coeficiente constante; Y comprender el método de cálculo para las ecuaciones diferenciales lineales homogéneas de orden superior. • Saber aplicar el método de cálculo de ecuaciones diferenciales para resolver problemas simples de aplicación geométrica y física. • Resolver correctamente ciertos tipos de ecuaciones diferenciales utilizando transformadas de Laplace.
Lecturas : [Ste12], [Zil13]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

***** EVALUATION MISSING *****

10. BIBLIOGRAFÍA BÁSICA

[Ste12] James Stewart. *Calculus*. 7th. CENGAGE Learning, 2012.

[Zil13] Dennis G. Zill. *Differential equations with Boundary value problems*. 8th. CENGAGE Learning, 2013.

Ministerio de Educación (MINEDU)
Sílabo 2021-I

1. CURSO

FG106. Teatro (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	: 2
2.2 Horas de teoría	: 1 (Semanal)
2.3 Horas de práctica	: -
2.4 Duración del periodo	: 16 semanas
2.5 Condición	: Obligatorio
2.6 Modalidad	: Presencial
2.7 Prerrequisitos	: FG101. Comunicación. (1 ^{er} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Favorece al estudiante a identificarse a la “Comunidad Académica” de la Universidad, en la medida en que le brinda canales naturales de integración a su grupo y a su Centro de Estudios y le permite, desde una visión alternativa, visualizar la valía interior de las personas a su alrededor, a la vez que puede conocer mejor la suya propia. Relaciona al universitario, a través de la experimentación, con un nuevo lenguaje, un medio de comunicación y expresión que va más allá de la expresión verbal conceptualizada. Coadyuva al estudiante en su formación integral, desarrollando en él capacidades corporales. Estimula en él, actitudes anímicas positivas, aptitudes cognitivas y afectivas. Enriquece su sensibilidad y despierta su solidaridad. Desinhibe y socializa, relaja y alegra, abriendo un camino de apertura de conocimiento del propio ser y el ser de los demás.

5. OBJETIVOS

- Contribuir a la formación personal y profesional del estudiante, reconociendo, valorando y desarrollando su lenguaje corporal, integrándolo a su grupo, afianzando su seguridad personal, enriqueciendo su intuición, su imaginación y creatividad, motivándolo a abrir caminos de búsqueda de conocimiento de sí mismo y de comunicación con los demás a través de su sensibilidad, de ejercicios de introspección y de nuevas vías de expresión.

6. COMPETENCIAS

f) Comunicarse efectivamente con audiencias diversas. (**Usar**)

ñ) Comprender que la formación de un buen profesional no se desliga ni se opone sino mas bien contribuye al auténtico crecimiento personal. Esto requiere de la asimilación de valores sólidos, horizontes espirituales amplios y una visión profunda del entorno cultural. (**Usar**)

■NoSpecificOutcomes■

7. TEMAS

Unidad 1: (6)	
Competencias esperadas: C18,C24	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">● ¿Qué es el Arte? Una experiencia vivencial y personal.● La llave maestra: la creatividad.● La importancia del teatro en la formación personal y profesional.● Utilidad y enfoque del arte teatral.	<ul style="list-style-type: none">● Reconocer la vigencia del Arte y la creatividad en el desarrollo personal y social [Usar].● Relacionar al estudiante con su grupo valorando la importancia de la comunicación humana y del colectivo social [Usar].● Reconocer nociones básicas del teatro [Usar].
Lecturas : [Maj58], [Pav98]	

Unidad 2: (6)	
Competencias esperadas: C17,C24	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Juego, luego existo. • El juego del niño y el juego dramático. • Juegos de integración grupal y juegos de creatividad. • La secuencia teatral. 	<ul style="list-style-type: none"> • Reconocer el juego como herramienta fundamental del teatro [Usar]. • Interiorizar y revalorar el juego como aprendizaje creativo [Usar]. • Acercar al estudiante de manera espontánea y natural, a la vivencia teatral [Usar].
Lecturas : [Maj58], [Pav98]	

Unidad 3: (9)	
Competencias esperadas: C17, C18, C24	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Toma de conciencia del cuerpo. • Toma de conciencia del espacio • Toma de conciencia del tiempo • Creación de secuencias individuales y colectivas: Cuerpo, espacio y tiempo. • El uso dramático del elemento: El juego teatral. • Presentaciones teatrales con el uso del elemento. 	<ul style="list-style-type: none"> • Experimentar con nuevas formas de expresión y comunicación [Usar]. • Conocer algunos mecanismos de control y manejo corporal [Usar]. • Brindar caminos para que el alumno pueda desarrollar creativamente su imaginación, su capacidad de relación y captación de estímulos auditivos, rítmicos y visuales [Usar]. • Conocer y desarrollar el manejo de su espacio propio y de sus relaciones espaciales [Usar]. • Experimentar estados emocionales diferentes y climas colectivos nuevos [Usar].
Lecturas : [Maj58], [Pav98]	

Unidad 4: (12)	
Competencias esperadas: C18, C24	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Relajación, concentración y respiración. • Desinhibición e interacción con el grupo. • La improvisación. • Equilibrio, peso, tiempo y ritmo. • Análisis del movimiento. Tipos de movimiento. • La presencia teatral. • La danza, la coreografía teatral. 	<ul style="list-style-type: none"> • Ejercitarse en el manejo de destrezas comunicativas no verbales [Usar]. • Practicar juegos y ejercicios de lenguaje corporal, individual y grupalmente [Usar]. • Expresar libre y creativamente sus emociones y sentimientos y su visión de la sociedad a través de representaciones originales con diversos lenguajes [Usar]. • Conocer los tipos de actuación [Usar].
Lecturas : [Maj58], [Pav98]	

Unidad 5: (3)	
Competencias esperadas: C24	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • El origen del teatro, el teatro griego y el teatro romano. • El teatro medieval , la comedia del arte. • De la pasión a la razón: Romanticismo e Ilustración. • El teatro realista, teatro épico. Brech y Stanislavski. • El teatro del absurdo, teatro contemporáneo y teatro total. • Teatro en el Perú: Yuyashkani, La Tarumba, pataclau, otros. 	<ul style="list-style-type: none"> • Conocer la influencia que la sociedad ha ejercido en el teatro y la respuesta de este arte ante los diferentes momentos de la historia [Usar]. • Apreciar el valor y aporte de las obras de dramaturgos importantes [Usar]. • Analizar el contexto social del arte teatral [Usar]. • Reflexionar sobre el Teatro Peruano y arequipeño [Usar].
Lecturas : [Maj58], [Pav98]	

Unidad 6: (12)	
Competencias esperadas: C17,C18, C24	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Apreciación teatral. Expectación de una o más obras teatrales. • El espacio escénico. • Construcción del personaje • Creación y montaje de una obra teatral . • Presentación en público de pequeñas obras haciendo uso de vestuario, maquillaje, escenografía, utilería y del empleo dramático del objeto. 	<ul style="list-style-type: none"> • Emplear la creación teatral, como manifestación de ideas y sentimientos propios ante la sociedad [Usar]. • Aplicar las técnicas practicadas y los conocimientos aprendidos en una apreciación y/o expresión teatral concreta que vincule el rol de la educación [Usar]. • Intercambiar experiencias y realizar presentaciones breves de ejercicios teatrales en grupo, frente a público [Usar].
Lecturas : [Maj58], [Pav98]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

***** EVALUATION MISSING *****

10. BIBLIOGRAFÍA BÁSICA

- [Maj58] Angel Majorana. *El arte de hablar en publico*. La España Moderna, 1958.
- [Pav98] Patrice Pavis. *Diccionario del Teatro*. Edit. Piados BA, 1998.

Ministerio de Educación (MINEDU)
Sílabo 2021-I

1. CURSO

ID102. Inglés II (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	: 6
2.2 Horas de teoría	: 2 (Semanal)
2.3 Horas de práctica	: -
2.4 Duración del periodo	: 16 semanas
2.5 Condición	: Obligatorio
2.6 Modalidad	: Presencial
2.7 Prerrequisitos	: ID101. Inglés I. (1 ^{er} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Parte fundamental de la formación integral de un profesional es la habilidad de comunicarse en un idioma extranjero además del propio idioma nativo. No solamente amplía su horizonte cultural sino que permite una visión más humana y comprensiva de la vida de las personas. En el caso de los idiomas extranjeros, indudablemente el Inglés es el más práctico porque es hablado alrededor de todo el mundo. No hay país alguno donde éste no sea hablado. En las carreras relacionadas con los servicios al turista el Inglés es tal vez la herramienta práctica más importante que el alumno debe dominar desde el primer momento, como parte de su formación integral.

5. OBJETIVOS

- Desarrollar la capacidad de hablar fluídamente el idioma.
- Incrementar el vocabulario y el manejo de expresiones simples.

6. COMPETENCIAS

f) Comunicarse efectivamente con audiencias diversas. (**Usar**)

■NoSpecificOutcomes■

7. TEMAS

Unidad 1: How long ago? (0)	
Competencias esperadas: 2	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Pasado Simple• Oraciones Negativas con ago.• Conjunciones• Expresiones de Tiempo en pasado• Relaciones y símbolos fonéticos• Expresiones para dar la fecha	<ul style="list-style-type: none">• Al terminar la octava unidad, cada uno de los alumnos, comprendiendo la gramática del tiempo pasado es capaz de expresar una mayor cantidad de expresiones de tiempo y además usar preposiciones para describir lugares y tiempos variados. Además es capaz de analizar y expresar ideas acerca de fechas y números en orden.
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 2: Food you like! (0)	
Competencias esperadas: 2	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Sustantivos Contables y No Contables • Expresiones con Would like y I'd like • Cuantificadores • Comidas alrededor del mundo • Pedidos formales • Cartas formales 	<ul style="list-style-type: none"> • Al terminar la novena unidad, los alumnos habiendo identificado la forma de expresar pedidos y hacer ofrecimientos, los utilizan en situaciones varias. Expresar situaciones y estados relacionados con cantidades. Explica y aplica vocabulario de comidas y bebidas.
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 3: The world of work (0)	
Competencias esperadas: 2	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Adjetivos • Oraciones con Adjetivos Comparativos. • Oraciones con Adjetivos Superlativos • Ciudades y el campo • Indicaciones de dirección 	<ul style="list-style-type: none"> • Al terminar la décima unidad, los alumnos habiendo reconocido las características de los adjetivos, utilizan éstos para hacer comparaciones de diversos tipos. Describen personas y lugares y dan indicaciones de dirección. Utilizarán conjunciones para unir ideas tipo.
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 4: Looking good! (0)	
Competencias esperadas: 2	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Presente Continuo • Oraciones Afirmativas, Negativas y Preguntas • Uso de Whose • Pronombres Posesivos • Ropa y colores • Expresiones a usar en tiendas de ropa • Símbolos fonéticos. 	<ul style="list-style-type: none"> • Al terminar la décimo primera unidad, los alumnos habiendo identificado la idea de expresar ideas de acciones que suceden en el momento o que se relacionan a cualquier tiempo estructuran oraciones en Presente Progresivo. Expresan ideas de posesión con respecto a la ropa y los colores.
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 5: Life is an adventure! (0)	
Competencias esperadas: 2	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Uso de going to • Oraciones en Tiempo Futuro • Expresiones de Cantidad. • Verbos de acción • Vocabulario del clima • Expresiones de Sugerencia • Escribir una postal 	<ul style="list-style-type: none"> • Al finalizar la décimo segunda unidad, los alumnos, a partir de la comprensión del tiempo futuro, elaborarán oraciones utilizando los elementos necesarios. Asimilarán además la necesidad de expresar infinitivos de propósito. Adquirirán vocabulario para describir el clima. Se presentará expresiones para hacer y pedir sugerencias.
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 6: You're pretty smart! (0)	
Competencias esperadas: 2	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Formas de Preguntas • Adverbios y Adjetivos • Vocabulario descripción de sentimientos • Expresiones para viajes en tren • Redacción de historias cortas • Lecturas. 	<ul style="list-style-type: none"> • Al finalizar la décimo tercera unidad, los alumnos habiendo conocido los fundamentos de la estructuración de preguntas diversas, realizarán trabajos aplicativos en contextos adecuados. Enfatizan la diferencia entre adjetivos y adverbios. Describen sentimientos. Utilizan expresiones para coger un tren. Asumen la idea se sufijos y prefijos.
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 7: Have you ever? (0)	
Competencias esperadas: 2	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Presente Perfecto • Expresiones con never, ever y yet • Vocabulario verbos en Participio pasado • Expresiones para viajes en avión • Redacción de cartas de agradecimiento • Lecturas 	<ul style="list-style-type: none"> • Al finalizar la décimo cuarta unidad, los alumnos habiendo conocido los fundamentos de la estructuración del Presente Perfecto experimentan la necesidad de poder expresar este tipo de tiempo en acciones. Realizarán prácticas en contextos adecuados. Enfatizan la diferencia entre pasado simple y presente perfecto. Describen acciones con never, ever y yet. Utilizan expresiones para utilizar en un aeropuerto.
Lecturas : [SJ02], [Cam06], [Mac99]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

***** EVALUATION MISSING *****

10. BIBLIOGRAFÍA BÁSICA

- [Cam06] Cambridge. *Diccionario Inglés-Espanol Cambridge*. Editorial Oxford, 2006.
[Mac99] James MacGrew. *Focus on Grammar Basic*. Editorial Oxford, 1999.
[SJ02] Liz Soars and John. *American Headway N 1 Student Book*. Editorial Oxford, 2002.

Ministerio de Educación (MINEDU)
Sílabo 2021-I

1. CURSO

CS113. Ciencia de la Computación II (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	4
2.2 Horas de teoría	:	2 (Semanal)
2.3 Horas de práctica	:	4 (Semanal)
2.4 Duración del periodo	:	16 semanas
2.5 Condición	:	Obligatorio
2.6 Modalidad	:	Presencial
2.7 Prerrequisitos	:	CS112. Ciencia de la Computación I. (2 ^{do} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Este es el tercer curso en la secuencia de los cursos introductorios a la informática. En este curso se pretende cubrir los conceptos señalados por la Computing Curricula IEEE(c)-ACM 2001, bajo el enfoque funcional-first. El paradigma orientado a objetos nos permite combatir la complejidad haciendo modelos a partir de abstracciones de los elementos del problema y utilizando técnicas como encapsulamiento, modularidad, polimorfismo y herencia. El dominio de estos temas permitirá que los participantes puedan dar soluciones computacionales a problemas de diseño sencillos del mundo real.

5. OBJETIVOS

- Introducir al alumno a los fundamentos del paradigma de orientación a objetos, permitiendo asimilar los conceptos necesarios para desarrollar un sistema de información

6. COMPETENCIAS

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Usar**)
- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Usar**)
- d) Trabajar efectivamente en equipos para cumplir con un objetivo común. (**Usar**)

- a10) Evaluar y aplicar pensamiento computacional para resolver problemas cotidianos. (10)
- a11) Utilizar de forma eficiente estructuras de control condicionales, repetitivas, funciones, recursividad, ordenamiento y búsqueda. (11)
- b1) Identificar y aplicar de forma eficiente diversas estrategias algorítmicas y estructuras de datos para la solución de un problema dadas ciertas restricciones de espacio y tiempo. (1)
- d1) Desarrollo colaborativo de software utilizando repositorios de código y gestión de versiones (ej. Git, Bitbucket, SVN). (1)

7. TEMAS

Unidad 1: Conceptos Fundamentales de Programación (5)**Competencias esperadas: a,b**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Sintaxis y semántica básica de un lenguaje de alto nivel.• Variables y tipos de datos primitivos (ej., numeros, caracteres, booleanos)• Expresiones y asignaciones.• Operaciones básicas I/O incluyendo archivos I/O.• Estructuras de control condicional e iterativas.• Paso de funciones y parámetros.• Concepto de recursividad.	<ul style="list-style-type: none">• Analiza y explica el comportamiento de programas simples que involucran estructuras fundamentales de programación variables, expresiones, asignaciones, E/S, estructuras de control, funciones, paso de parámetros, y recursividad [Usar]• Identifica y describe el uso de tipos de datos primitivos [Usar]• Escribe programas que usan tipos de datos primitivos [Usar]• Modifica y expande programas cortos que usen estructuras de control condicionales e iterativas así como funciones [Usar]• Diseña, implementa, prueba, y depura un programa que usa cada una de las siguientes estructuras de datos fundamentales: cálculos básicos, E/S simple, condicional estándar y estructuras iterativas, definición de funciones, y paso de parámetros [Usar]• Escribe un programa que usa E/S de archivos para brindar persistencia a través de ejecuciones múltiples [Usar]• Escoje estructuras de condición y repetición adecuadas para una tarea de programación dada [Usar]• Describe el concepto de recursividad y da ejemplos de su uso [Usar]• Identifica el caso base y el caso general de un problema basado en recursividad [Usar]
Lecturas : [stroustrup2013], [Van02], [LE13]	

Unidad 2: Programación orientada a objetos (7)**Competencias esperadas: a,b**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">● Diseño orientado a objetos:<ul style="list-style-type: none">– Descomposición en objetos que almacenan estados y poseen comportamiento– Diseño basado en jerarquía de clases para modelamiento● Definición de las categorías, campos, métodos y constructores.● Las subclases, herencia y método de alteración temporal.● Asignación dinámica: definición de método de llamada.● Subtipificación:<ul style="list-style-type: none">– Polimorfismo artículo Subtipo; upcasts implícitos en lenguajes con tipos.– Noción de reemplazo de comportamiento: los subtipos de actuar como supertipos.– Relación entre subtipos y la herencia.● Lenguajes orientados a objetos para la encapsulación:<ul style="list-style-type: none">– privacidad y la visibilidad de miembros de la clase– Interfaces revelan único método de firmas– clases base abstractas● Uso de colección de clases, iteradores, y otros componentes de la librería estándar.	<ul style="list-style-type: none">● Diseñar e implementar una clase [Usar]● Usar subclase para diseñar una jerarquía simple de clases que permita al código ser reusable por diferentes subclases [Usar]● Razonar correctamente sobre el flujo de control en un programa mediante el envío dinámico [Usar]● Comparar y contrastar (1) el enfoque procedurales/funcional- definiendo una función por cada operación con el uso de la función proporcionando un caso por cada variación de dato - y (2) el enfoque orientado a objetos - definiendo una clase por cada variación de dato con la definición de la clase proporcionando un método por cada operación. Entender ambos enfoques como una definición de variaciones y operaciones de una matriz [Usar]● Explicar la relación entre la herencia orientada a objetos (código compartido y <i>overriding</i>) y subtipificación (la idea de un subtipo es ser utilizable en un contexto en el que espera al supertipo) [Usar]● Usar mecanismos de encapsulación orientada a objetos, tal como interfaces y miembros privados [Usar]● Definir y usar iteradores y otras operaciones sobre agregaciones, incluyendo operaciones que tienen funciones como argumentos, en múltiples lenguajes de programación, seleccionar la forma más natural por cada lenguaje [Usar]
Lecturas : [Str13]	

Unidad 3: Algoritmos y Diseño (5)**Competencias esperadas: a,b,d**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Conceptos y propiedades de los algoritmos<ul style="list-style-type: none">– Comparación informal de la eficiencia de los algoritmos (ej., conteo de operaciones)• Rol de los algoritmos en el proceso de solución de problemas• Estrategias de solución de problemas<ul style="list-style-type: none">– Funciones matemáticas iterativas y recursivas– Recorrido iterativo y recursivo en estructura de datos– Estrategias Divide y Conquistar• Conceptos y principios fundamentales de diseño<ul style="list-style-type: none">– Abstracción– Descomposición de Program– Encapsulamiento y camuflaje de información– Separación de comportamiento y aplicación	<ul style="list-style-type: none">• Discute la importancia de los algoritmos en el proceso de solución de un problema [Usar]• Discute como un problema puede ser resuelto por múltiples algoritmos, cada uno con propiedades diferentes [Usar]• Crea algoritmos para resolver problemas simples [Usar]• Usa un lenguaje de programación para implementar, probar, y depurar algoritmos para resolver problemas simples [Usar]• Implementa, prueba, y depura funciones recursivas simples y sus procedimientos [Usar]• Determina si una solución iterativa o recursiva es la más apropiada para un problema [Usar]• Implementa un algoritmo de divide y vencerás para resolver un problema [Usar]• Aplica técnicas de descomposición para dividir un programa en partes más pequeñas [Usar]• Identifica los componentes de datos y el comportamiento de múltiples tipos de datos abstractos [Usar]• Implementa un tipo de dato abstracto coherente, con la menor pérdida de acoplamiento entre componentes y comportamientos [Usar]• Identifica las fortalezas y las debilidades relativas entre múltiples diseños e implementaciones de un problema [Usar]
Lecturas : [stroustrup2013], [Weert16], [LE13]	

Unidad 4: Análisis Básico (3)**Competencias esperadas: a,b**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Diferencias entre el mejor, el esperado y el peor caso de un algoritmo. • Análisis asintótico de complejidad de cotas superior y esperada. • Definición formal de la Notación Big O. • Clases de complejidad como constante, logarítmica, lineal, cuadrática y exponencial. • Medidas empíricas de desempeño. • Compensación entre espacio y tiempo en los algoritmos. • Uso de la notación Big O. • Notación Little o, Big omega y Big theta. • Relaciones recurrentes. • Análisis de algoritmos iterativos y recursivos. • Teorema Maestro y Árboles Recursivos. 	<ul style="list-style-type: none"> • Explique a que se refiere con “mejor”, “esperado” y “peor” caso de comportamiento de un algoritmo [Usar] • En el contexto de a algoritmos específicos, identifique las características de data y/o otras condiciones o suposiciones que lleven a diferentes comportamientos [Usar] • Determine informalmente el tiempo y el espacio de complejidad de diferentes algoritmos [Usar] • Indique la definición formal de Big O [Usar] • Lista y contraste de clases estándares de complejidad [Usar] • Realizar estudios empíricos para validar una hipótesis sobre runtime stemming desde un análisis matemático Ejecute algoritmos con entrada de varios tamaños y compare el desempeño [Usar] • Da ejemplos que ilustran las compensaciones entre espacio y tiempo que se dan en los algoritmos [Usar] • Use la notación formal de la Big O para dar límites superiores asintóticos en la complejidad de tiempo y espacio de los algoritmos [Usar] • Usar la notación formal Big O para dar límites de casos esperados en el tiempo de complejidad de los algoritmos [Usar] • Explicar el uso de la notación theta grande, omega grande y o pequeña para describir la cantidad de trabajo hecho por un algoritmo [Usar] • Usar relaciones recurrentes para determinar el tiempo de complejidad de algoritmos recursivamente definidos [Usar] • Resuelve relaciones de recurrencia básicas, por ejemplo. usando alguna forma del Teorema Maestro [Usar]
Lecturas : [Str13]	

Unidad 5: Sistemas de tipos básicos (5)**Competencias esperadas: a,b**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Tipos como conjunto de valores junto con un conjunto de operaciones.<ul style="list-style-type: none">– Tipos primitivos (p.e. números, booleanos)– Composición de tipos contruídos de otros tipos (p.e., registros, uniones, arreglos, listas, funciones, referencias)• Asociación de tipos de variables, argumentos, resultados y campos.• Tipo de seguridad y los errores causados por el uso de valores de manera incompatible dadas sus tipos previstos.• Metas y limitaciones de tipos estáticos<ul style="list-style-type: none">– Eliminación de algunas clases de errores sin ejecutar el programa– Indecisión significa que un análisis estatico puede aproximar el comportamiento de un programa• Tipos genéricos (polimorfismo paramétrico)<ul style="list-style-type: none">– Definición– Uso de librerías genéricas tales como colecciones.– Comparación con polimorfismo ad-hoc y polimorfismo de subtipos• Beneficios complementarios de tipos estáticos y dinámicos:<ul style="list-style-type: none">– Errores tempranos vs. errores tardíos/evitados.– Refuerzo invariante durante el desarrollo y mantenimiento del código vs. decisiones pospuestas de tipos durante la la creación de prototipos y permitir convenientemente la codificación flexible de patrones tales como colecciones heterogéneas.– Evitar el mal uso del código vs. permitir más reuso de código.– Detectar programas incompletos vs. permitir que programas incompletos se ejecuten	<ul style="list-style-type: none">• Tanto para tipo primitivo y un tipo compuesto, describir de manera informal los valores que tiene dicho tipo [Usar]• Para un lenguaje con sistema de tipos estático, describir las operaciones que están prohibidas de forma estática, como pasar el tipo incorrecto de valor a una función o método [Usar]• Describir ejemplos de errores de programa detectadas por un sistema de tipos [Usar]• Para múltiples lenguajes de programación, identificar propiedades de un programa con verificación estática y propiedades de un programa con verificación dinámica [Usar]• Dar un ejemplo de un programa que no verifique tipos en un lenguaje particular y sin embargo no tenga error cuando es ejecutado [Usar]• Usar tipos y mensajes de error de tipos para escribir y depurar programas [Usar]• Explicar como las reglas de tipificación definen el conjunto de operaciones que legales para un tipo [Usar]• Escribir las reglas de tipo que rigen el uso de un particular tipo compuesto [Usar]• Explicar por qué indecidibilidad requiere sistemas de tipo para conservadoramente aproximar el comportamiento de un programa [Usar]• Definir y usar piezas de programas (tales como, funciones, clases, métodos) que usan tipos genéricos, incluyendo para colecciones [Usar]• Discutir las diferencias entre, genéricos (<i>generics</i>), subtipo y sobrecarga [Usar]• Explicar múltiples beneficios y limitaciones de tipificación estática en escritura, mantenimiento y depuración de un software [Usar]
Lecturas : [Str13]	

Unidad 6: Algoritmos y Estructuras de Datos fundamentales (3)**Competencias esperadas: a,b,d**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Algoritmos numéricos simples, tales como el cálculo de la media de una lista de números, encontrar el mínimo y máximo.• Algoritmos de búsqueda secuencial y binaria.• Algoritmos de ordenamiento de peor caso cuadrático (selección, inserción)• Algoritmos de ordenamiento con peor caso o caso promedio en $O(N \lg N)$ (Quicksort, Heapsort, Mergesort)• Tablas Hash, incluyendo estrategias para evitar y resolver colisiones.• Árboles de búsqueda binaria:<ul style="list-style-type: none">– Operaciones comunes en árboles de búsqueda binaria como seleccionar el mínimo, máximo, insertar, eliminar, recorrido en árboles.• Grafos y algoritmos en grafos:<ul style="list-style-type: none">– Representación de grafos (ej., lista de adyacencia, matriz de adyacencia)– Recorrido en profundidad y amplitud• Montículos (Heaps)• Grafos y algoritmos en grafos:<ul style="list-style-type: none">– Problema de corte máximo y mínimo– Búsqueda local• Búsqueda de patrones y algoritmos de cadenas/texto (ej. búsqueda de subcadena, búsqueda de expresiones regulares, algoritmos de subsecuencia común más larga)	<ul style="list-style-type: none">• Implementar algoritmos numéricos básicos [Usar]• Implementar algoritmos de búsqueda simple y explicar las diferencias en sus tiempos de complejidad [Usar]• Ser capaz de implementar algoritmos de ordenamiento comunes cuadráticos y $O(N \log N)$ [Usar]• Describir la implementación de tablas hash, incluyendo resolución y el evitamiento de colisiones [Usar]• Discutir el tiempo de ejecución y eficiencia de memoria de los principales algoritmos de ordenamiento, búsqueda y hashing [Usar]• Discutir factores otros que no sean eficiencia computacional que influyan en la elección de algoritmos, tales como tiempo de programación, mantenibilidad, y el uso de patrones específicos de la aplicación en los datos de entrada [Usar]• Explicar como el balanceamiento del arbol afecta la eficiencia de varias operaciones de un arbol de búsqueda binaria [Usar]• Resolver problemas usando algoritmos básicos de grafos, incluyendo búsqueda por profundidad y búsqueda por amplitud [Usar]• Demostrar habilidad para evaluar algoritmos, para seleccionar de un rango de posibles opciones, para proveer una justificación por esa selección, y para implementar el algoritmo en un contexto en específico [Usar]• Describir la propiedad del heap y el uso de heaps como una implementación de colas de prioridad [Usar]• Resolver problemas usando algoritmos de grafos, incluyendo camino más corto de una sola fuente y camino más corto de todos los pares, y como mínimo un algoritmo de arbol de expansion minima [Usar]• Trazar y/o implementar un algoritmo de comparación de string [Usar]
Lecturas : [stroustrup2013], [PA18]	

Unidad 7: Programación reactiva y dirigida por eventos (2)	
Competencias esperadas: a,b	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Eventos y controladores de eventos. • Usos canónicos como interfaces gráficas de usuario, dispositivos móviles, robots, servidores. • Uso de frameworks reactivos. <ul style="list-style-type: none"> – Definición de controladores/oyentes (handles/listeners) de eventos. – Bucle principal de eventos no controlado por el escritor controlador de eventos (event-handler-writer) • Eventos y eventos del programa generados externamente generada. • La separación de modelo, vista y controlador. 	<ul style="list-style-type: none"> • Escribir manejadores de eventos para su uso en sistemas reactivos tales como GUIs [Usar] • Explicar porque el estilo de programación manejada por eventos es natural en dominios donde el programa reacciona a eventos externos [Usar] • Describir un sistema interactivo en términos de un modelo, una vista y un controlador [Usar]
Lecturas : [stroustrup2013], [Will1]	

Unidad 8: Árboles y Grafos (7)	
Competencias esperadas: a,b,d	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Árboles. <ul style="list-style-type: none"> – Propiedades – Estrategias de recorrido • Grafos no dirigidos • Grafos dirigidos • Grafos ponderados • Árboles de expansion/bosques. • Isomorfismo en grafos. 	<ul style="list-style-type: none"> • Ilustrar mediante ejemplos la terminología básica de teoría de grafos, y de alguna de las propiedades y casos especiales de cada tipo de grafos/árboles [Usar] • Demostrar diversos métodos de recorrer árboles y grafos, incluyendo recorridos pre, post e inorden de árboles [Usar] • Modelar una variedad de problemas del mundo real en ciencia de la computación usando formas adecuadas de grafos y árboles, como son la representación de una topología de red o la organización jerárquica de un sistema de archivos [Usar] • Demostrar como los conceptos de grafos y árboles aparecen en estructuras de datos, algoritmos, técnicas de prueba (inducción estructurada), y conteos [Usar] • Explicar como construir un árbol de expansión de un grafo [Usar] • Determinar si dos grafos son isomorfos [Usar]
Lecturas : [Nak13]	

Unidad 9: Diseño de Software (6)

Competencias esperadas: a,b

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Principios de diseño del sistema: niveles de abstracción (diseño arquitectónico y el diseño detallado), separación de intereses, ocultamiento de información, de acoplamiento y de cohesión, de reutilización de estructuras estándar. • Diseño de paradigmas tales como diseño estructurado (descomposición funcional de arriba hacia abajo), el análisis orientado a objetos y diseño, orientado a eventos de diseño, diseño de nivel de componente, centrado datos estructurada, orientada a aspectos, orientado a la función, orientado al servicio. • Modelos estructurales y de comportamiento de los diseños de software. • Diseño de patrones. • Relaciones entre los requisitos y diseños: La transformación de modelos, el diseño de los contratos, invariantes. • Conceptos de arquitectura de software y arquitecturas estándar (por ejemplo, cliente-servidor, n-capas, transforman centrados, tubos y filtros). • El uso de componentes de diseño: selección de componentes, diseño, adaptación y componentes de ensamblaje, componentes y patrones, componentes y objetos (por ejemplo, construir una GUI usando un standard widget set) • Diseños de refactorización utilizando patrones de diseño • Calidad del diseño interno, y modelos para: eficiencia y desempeño, redundancia y tolerancia a fallos, trazabilidad de los requerimientos. • Medición y análisis de la calidad de un diseño. • Compensaciones entre diferentes aspectos de la calidad. • Aplicaciones en frameworks. • Middleware: El paradigma de la orientación a objetos con middleware, requerimientos para correr y clasificar objetos, monitores de procesamiento de transacciones y el sistema de flujo de trabajo. • Principales diseños de seguridad y codificación (cross-reference IAS/Principles of secure design). <ul style="list-style-type: none"> – Principio de privilegios mínimos – Principio de falla segura por defecto – Principio de aceptabilidad psicológica 	<ul style="list-style-type: none"> • Formular los principios de diseño, incluyendo la separación de problemas, ocultación de información, acoplamiento y cohesión, y la encapsulación [Usar] • Usar un paradigma de diseño para diseñar un sistema de software básico y explicar cómo los principios de diseño del sistema se han aplicado en este diseño [Usar] • Construir modelos del diseño de un sistema de software simple los cuales son apropiado para el paradigma utilizado para diseñarlo [Usar] • En el contexto de un paradigma de diseño simple, describir uno o más patrones de diseño que podrían ser aplicables al diseño de un sistema de software simple [Usar] • Para un sistema simple adecuado para una situación dada, discutir y seleccionar un paradigma de diseño apropiado [Usar] • Crear modelos apropiados para la estructura y el comportamiento de los productos de software desde la especificaciones de requisitos [Usar] • Explicar las relaciones entre los requisitos para un producto de software y su diseño, utilizando los modelos apropiados [Usar] • Para el diseño de un sistema de software simple dentro del contexto de un único paradigma de diseño, describir la arquitectura de software de ese sistema [Usar] • Dado un diseño de alto nivel, identificar la arquitectura de software mediante la diferenciación entre las arquitecturas comunes de software, tales como 3 capas (<i>3-tier</i>), <i>pipe-and-filter</i>, y cliente-servidor [Usar] • Investigar el impacto de la selección arquitecturas de software en el diseño de un sistema simple [Usar] • Aplicar ejemplos simples de patrones en un diseño de software [Usar] • Describir una manera de refactorar y discutir cuando esto debe ser aplicado [Usar] • Seleccionar componentes adecuados para el uso en un diseño de un producto de software [Usar] • Explicar cómo los componentes deben ser adaptados para ser usados en el diseño de un producto de software [Usar] • Diseñar un contrato para un típico componente de software pequeño para el uso de un dado sistema [Usar] • Discutir y seleccionar la arquitectura de software adecuada para un sistema de software simple para un dado escenario [Usar]

Unidad 10: Ingeniería de Requisitos (1)	
Competencias esperadas: a,b	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Al describir los requisitos funcionales utilizando, por ejemplo, los casos de uso o historias de los usuarios. • Propiedades de requisitos, incluyendo la consistencia, validez, integridad y viabilidad. • Requisitos de software elicitación. • Descripción de datos del sistema utilizando, por ejemplo, los diagramas de clases o diagramas entidad-relación. • Requisitos no funcionales y su relación con la calidad del software. • Evaluación y uso de especificaciones de requisitos. • Requisitos de las técnicas de modelado de análisis. • La aceptabilidad de las consideraciones de certeza/incertidumbre sobre el comportamiento del software/sistema. • Prototipos. • Conceptos básicos de la especificación formal de requisitos. • Especificación de requisitos. • Validación de requisitos. • Rastreo de requisitos. 	<ul style="list-style-type: none"> • Enumerar los componentes clave de un caso de uso o una descripción similar de algún comportamiento que es requerido para un sistema [Usar] • Describir cómo el proceso de ingeniería de requisitos apoya la obtención y validación de los requisitos de comportamiento [Usar] • Interpretar un modelo de requisitos dada por un sistema de software simple [Usar] • Describir los retos fundamentales y técnicas comunes que se utilizan para la obtención de requisitos [Usar] • Enumerar los componentes clave de un modelo de datos (por ejemplo, diagramas de clases o diagramas ER) [Usar] • Identificar los requisitos funcionales y no funcionales en una especificación de requisitos dada por un sistema de software [Usar] • Realizar una revisión de un conjunto de requisitos de software para determinar la calidad de los requisitos con respecto a las características de los buenos requisitos [Usar] • Aplicar elementos clave y métodos comunes para la obtención y el análisis para producir un conjunto de requisitos de software para un sistema de software de tamaño medio [Usar] • Comparar los métodos ágiles y el dirigido por planes para la especificación y validación de requisitos y describir los beneficios y riesgos asociados con cada uno [Usar] • Usar un método común, no formal para modelar y especificar los requisitos para un sistema de software de tamaño medio [Usar] • Traducir al lenguaje natural una especificación de requisitos de software (por ejemplo, un contrato de componentes de software) escrito en un lenguaje de especificación formal [Usar] • Crear un prototipo de un sistema de software para reducir el riesgo en los requisitos [Usar] • Diferenciar entre el rastreo (<i>tracing</i>) hacia adelante y hacia atrás y explicar su papel en el proceso de validación de requisitos [Usar]
Lecturas : [Str13]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

***** EVALUATION MISSING *****

10. BIBLIOGRAFÍA BÁSICA

- [LE13] Stanley B. Lippman and Barbara E. Moo. *C++ Primer*. 5th. O'Reilly, 2013. ISBN: 9780133053043.
- [Nak13] S. Nakariakov. *The Boost C++ Libraries: Generic Programming*. CreateSpace Independent Publishing Platform, 2013.
- [PA18] Praseed Pai and Peter Abraham. *C++ Reactive Programming*. 1st. Packt, 2018.
- [Str13] B Stroustrup. *The C++ Programming Language, 4th edition*. Addison-Wesley, 2013.
- [Van02] David Vandervoerde. *C++ Templates: The Complete Guide*. 1st. Addison-Wesley, 2002. ISBN: 978-0134448237.
- [Wil11] Anthony Williams. *C++ Concurrency in Action*. 1st. Manning, 2011.

Ministerio de Educación (MINEDU)
Sílabo 2021-I

1. CURSO

CS221. Arquitectura de Computadores (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	3
2.2 Horas de teoría	:	2 (Semanal)
2.3 Horas de práctica	:	2 (Semanal)
2.4 Duración del periodo	:	16 semanas
2.5 Condición	:	Obligatorio
2.6 Modalidad	:	Presencial
2.7 Prerrequisitos	:	CS1D2. Estructuras Discretas II. (2 ^{do} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Un profesional en Ciencia de la Computación debe tener un conocimiento sólido de la organización y los principios de diseño de diversos sistemas de computación, al comprender las limitaciones de los sistemas modernos serán capaces de proponer nuevos paradigmas en la próxima generación. Este curso enseña los fundamentos y principios de la arquitectura de computadoras. Esta clase incluye diseño de lógica digital, conceptos básicos de arquitectura de computadora y diseño de procesador (*Instruction Set Architecture*, microarquitectura, ejecución fuera de orden, predicción de *branches*), paradigmas de ejecución (superescalar, flujo de datos, VLIW, SIMD, GPU, sistólica, multiproceso) y organización del sistema de memoria.

5. OBJETIVOS

- Proporcionar un primer enfoque en Arquitectura de Computadoras.
- Estudiar el diseño y la evolución de las arquitecturas de computador, que llevaron a las implementaciones de los sistemas modernos.
- Proporcionar un estudio profundo del hardware y su relación con la ejecución del software.
- Implementar un microprocesador simple usando el lenguaje Verilog.

6. COMPETENCIAS

- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Usar**)
- g) Analizar el impacto local y global de la computación sobre los individuos, organizaciones y sociedad. (**Usar**)
- i) Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (**Evaluar**)

■NoSpecificOutcomes■

7. TEMAS

Unidad 1: Lógica digital y sistemas digitales (18)	
Competencias esperadas: b	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Revisión e historia de la Arquitectura de Computadores. • Lógica combinacional y secuencial/<i>field programmable gate arrays</i> como bloque fundamental de construcción lógico combinacional secuencial. • Modelos de representación(abstracción) • Herramientas de diseño asistidas por computadora que procesan hardware y representaciones arquitecturales. • Registrar transferencia notación / Hardware language descriptivo (Verilog/VHDL) • Restricción física (Retrasos de Entrada, fan-in, fan-out, energía/potencia) 	<ul style="list-style-type: none"> • Describir el avance de la tecnología de dispositivos, desde los tubos de vacío hasta VLSI, desde las arquitecturas mainframe a las arquitecturas en escala warehouse [Familiarizarse] • Comprender que la tendencia de las arquitecturas modernas de computadores es hacia núcleos múltiples y que el paralelismo es inherente en todos los sistemas de hardware [Usar] • Explicar las implicancias de los límites de potencia para mejoras adicionales en el rendimiento de los procesadores y también en el aprovechamiento del paralelismo [Usar] • Relacionar las varias representaciones equivalentes de la funcionalidad de un computador, incluyendo expresiones y puertas lógicas, y ser capaces de utilizar expresiones matemáticas para describir las funciones de circuitos combinacionales y secuenciales sencillos [Familiarizarse] • Diseñar los componentes básicos de construcción de un computador: unidad aritmético lógica (a nivel de puertas lógicas), unidad central de procesamiento (a nivel de registros de transferencia), memoria (a nivel de registros de transferencia) [Usar] • Usar herramientas CAD para capturar, sistetizar, y simular bloques de construcción (como ALUs, registros, movimiento entre registros) de un computador simple [Familiarizarse] • Evaluar el comportamiento de un diagrama de tiempos y funcional de un procesador simple implementado a nivel de circuitos lógicos [Evaluar]
Lecturas : [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

Unidad 2: Representación de datos a nivel máquina (8)**Competencias esperadas: g**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Bits, Bytes y Words.• Representación de datos numérica y bases numéricas.• Sistemas de punto flotante y punto fijo.• Representaciones con signo y complemento a 2.• Representación de información no numérica (códigos de caracteres, información gráfica)• Representación de registros y arreglos.	<ul style="list-style-type: none">• Explicar porqué en computación todo es datos, inclusive las instrucciones [Evaluar]• Explicar las razones de usar formatos alternativos para representar datos numéricos [Familiarizarse]• Describir cómo los enteros negativos se almacenan con representaciones de bit de signo y complemento a 2 [Usar]• Explicar cómo las representaciones de tamaño fijo afectan en la exactitud y la precisión [Usar]• Describir la representación interna de datos no numéricos como caracteres, cadenas, registros y arreglos [Usar]• Convertir datos numéricos de un formato a otro [Usar]
Lecturas : [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

Unidad 3: Organización de la Máquina a Nivel Ensamblador (8)**Competencias esperadas: b,g**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Organización Básica de la Máquina de Von Neumann.• Unidad de Control.• <i>Instruction sets</i> y tipos (manipulación de información, control, I/O)• Assembler y Programación en Lenguaje de Máquina.• Formato de instrucciones.• Modos de direccionamiento.• Llamada a subrutinas y mecanismos de retorno.• I/O e Interrupciones.• Montículo (Heap) vs. Estático vs. Pila vs. Segmentos de código.	<ul style="list-style-type: none">• Explicar la organización de la maquina clásica de von Neumann y sus principales unidades funcionales [Familiarizarse]• Describir cómo se ejecuta una instrucción en una máquina de von Neumann con extensión para hebras, sincronización multiproceso y ejecución SIMD (máquina vectorial) [Familiarizarse]• Describir el paralelismo a nivel de instrucciones y sus peligros, y cómo es esto tratado en pipelines de proceso típicos [Familiarizarse]• Resumir cómo se representan las instrucciones, tanto a nivel de máquina bajo el contexto de un ensamblador simbólico [Familiarizarse]• Demostrar cómo se mapean los patrones de lenguajes de alto nivel en notaciones en lenguaje ensamblador o en código máquina [Usar]• Explicar los diferentes formatos de instrucciones, así como el direccionamiento por instrucción, y comparar formatos de tamaño fijo y variable [Usar]• Explicar como las llamadas a subrutinas son manejadas a nivel de ensamblador [Usar]• Explicar los conceptos básicos de interrupciones y operaciones de entrada y salida (I/O) [Familiarizarse]• Escribir segmentos de programa simples en lenguaje ensamblador [Usar]• Ilustrar cómo los bloques constructores fundamentales en lenguajes de alto nivel son implementados a nivel de lenguaje máquina [Usar]
Lecturas : [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

Unidad 4: Organización funcional (8)	
Competencias esperadas: b,g	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Implementación de <i>datapath</i>, incluyendo un <i>pipeline</i> de instrucciones, detección de <i>hazards</i> y la resolución. • Control de unidades: Microprogramada. • Instrucción (Pipelining) • Introducción al paralelismo al nivel de instrucción (PNI) 	<ul style="list-style-type: none"> • Comparar implementaciones alternativas de ruta de datos [Evaluar] • Discutir el concepto de puntos de control y la generación de señales de control usando implementaciones a nivel de circuito o microprogramadas [Familiarizarse] • Explicar el paralelismo a nivel de instrucciones básicas usando pipelining y los mayores riesgos que pueden ocurrir [Usar] • Diseñar e implementar un procesador completo, incluyendo ruta de datos y control [Usar] • Calcular la cantidad promedio de ciclos por instrucción de una implementación con procesador y sistema de memoria determinados [Evaluar]
Lecturas : [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

Unidad 5: Organización y Arquitectura del Sistema de Memoria (8)	
Competencias esperadas: b,g	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Sistemas de Almacenamiento y su Tecnología. • Jerarquía de Memoria: importancia de la localización temporal y espacial. • Organización y Operaciones de la Memoria Principal. • Latencia, ciclos de tiempo, ancho de banda e <i>inter-leaving</i>. • Memorias caché (Mapeo de direcciones, Tamaño de bloques, Reemplazo y Políticas de almacenamiento) • Multiprocesador coherencia cache / Usando el sistema de memoria para las operaciones de sincronización de memoria / atómica inter-core. • Memoria virtual (tabla de página, TLB) • Manejo de Errores y confiabilidad. • Error de codificación, compresión de datos y la integridad de datos. 	<ul style="list-style-type: none"> • Identificar las principales tecnologías de memoria (Por ejemplo: SRAM, DRAM, Flash, Disco Magnético) y su relación costo beneficio [Familiarizarse] • Explique el efecto de latencia de memoria en tiempo de ejecución [Familiarizarse] • Describir como el uso de jerarquía de memoria (caché, memoria virtual) es aplicado para reducir el atraso efectivo en la memoria [Usar] • Describir los principios de la administración de memoria [Usar] • Explique el funcionamiento de un sistema con gestión de memoria virtual [Usar] • Calcule el tiempo de acceso promedio a memoria bajo varias configuraciones de caché y memoria y para diversas combinaciones de instrucciones y referencias a datos [Evaluar]
Lecturas : [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

Unidad 6: Interfaz y comunicación (8)	
Competencias esperadas: b,g,i	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Fundamentos de I/O: Handshaking, Bbuffering, I/O programadas, interrupciones dirigidas de I/O. • Interrumpir estructuras: interrumpir reconocimiento, vectorizado y priorizado. • Almacenamiento externo, organización física y discos. • Buses: Protocolos de bus, arbitraje, acceso directo a memoria (DMA). • Introducción a Redes: comunicación de redes como otra capa de acceso remoto. • Soporte Multimedia. • Arquitecturas RAID. 	<ul style="list-style-type: none"> • Explicar como las interrupciones son aplicadas para implementar control de entrada-salida y transferencia de datos [Familiarizarse] • Identificar diversos tipos de buses en un sistema computacional [Familiarizarse] • Describir el acceso a datos desde una unidad de disco magnético [Usar] • Comparar organizaciones de red conocidas como organizaciones en bus/Ethernet, en anillo y organizaciones conmutadas versus ruteadas [Evaluar] • Identificar las interfaces entre capas necesarios para el acceso y presentación multimedia, desde la captura de la imagen en almacenamiento remoto, a través del transporte por una red de comunicaciones, hasta la puesta en la memoria local y la presentación final en una pantalla gráfica [Familiarizarse] • Describir las ventajas y limitaciones de las arquitecturas RAID [Familiarizarse]
Lecturas : [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

Unidad 7: Multiprocesamiento y arquitecturas alternativas (8)	
Competencias esperadas: i	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • <i>Power Law</i>. • Ejemplos de <i>sets</i> de instrucciones y arquitecturas SIMD y MIMD. • Redes de interconexión (Hypercube, Shuffle-exchange, Mesh, Crossbar) • Sistemas de memoria de multiprocesador compartido y consistencia de memoria. • Coherencia de cache multiprocesador. 	<ul style="list-style-type: none"> • Discutir el concepto de procesamiento paralelo mas allá del clásico modelo de von Neumann [Evaluar] • Describir diferentes arquitecturas paralelas como SIMD y MIMD [Familiarizarse] • Explicar el concepto de redes de interconexión y mostrar diferentes enfoques [Usar] • Discutir los principales cuidados en los sistemas de multiprocesamiento presentes con respecto a la gestión de memoria y describir como son tratados [Familiarizarse] • Describir las diferencias entre conectores eléctricos en paralelo backplane, interconexión memoria procesador y memoria remota via red, sus implicaciones para la latencia de acceso y el impacto en el rendimiento de un programa [Evaluar]
Lecturas : [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

Unidad 8: Mejoras de rendimiento (8)	
Competencias esperadas: g,i	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Arquitectura superescalar. • Predicción de ramificación, Ejecución especulativa, Ejecución fuera de orden. • Prefetching. • Procesadores vectoriales y GPU's • Soporte de hardware para multiprocesamiento. • Escalabilidad. • Arquitecturas alternativas, como VLIW / EPIC y aceleradores y otros tipos de procesadores de propósito especial. 	<ul style="list-style-type: none"> • Describir las arquitecturas superescalares y sus ventajas [Familiarizarse] • Explicar el concepto de predicción de bifurcaciones y su utilidad [Usar] • Caracterizar los costos y beneficios de la precarga prefetching [Evaluar] • Explicar la ejecución especulativa e identifique las condiciones que la justifican [Evaluar] • Discutir las ventajas de rendimiento ofrecida en una arquitectura de multihebras junto con los factores que hacen difícil dar el máximo beneficio de estas [Evaluar] • Describir la importancia de la escalabilidad en el rendimiento [Evaluar]
Lecturas : [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

***** EVALUATION MISSING *****

10. BIBLIOGRAFÍA BÁSICA

- [HH12] David Harris and Sarah Harris. *Digital Design and Computer Architecture*. 2nd. Morgan Kaufmann, 2012. ISBN: 978-0123944245.
- [HP06] J. L. Hennessy and D. A. Patterson. *Computer Architecture: A Quantitative Approach*. 4th. San Mateo, CA: Morgan Kaufman, 2006.
- [JAs07] Peter J. Ashenden. *Digital Design (Verilog): An Embedded Systems Approach Using Verilog*. Morgan Kaufmann, 2007. ISBN: 978-0123695277.
- [Par05] Behrooz Parhami. *Computer Architecture: From Microprocessors to Supercomputers*. New York: Oxford Univ. Press, 2005. ISBN: ISBN 0-19-515455-X.
- [PCh06] Pong P. Chu. *RTL Hardware Design Using VHDL*. 1st. Wiley-Interscience, 2006.
- [PH04] D. A. Patterson and J. L. Hennessy. *Computer Organization and Design: The Hardware/Software Interface*. 3rd ed. San Mateo, CA: Morgan Kaufman, 2004.

- [PP05] Yale N Patt and Sanjay J Patel. *Introduction to Computing Systems*. 2nd. McGraw Hill, 2005.
- [Sta10] William Stalings. *Computer Organization and Architecture: Designing for Performance*. 8th. Upper Saddle River, NJ: Prentice Hall, 2010.

Ministerio de Educación (MINEDU)
Sílabo 2021-I

1. CURSO

CS2B1. Desarrollo Basado en Plataformas (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	3
2.2 Horas de teoría	:	1 (Semanal)
2.3 Horas de práctica	:	2 (Semanal)
2.4 Duración del periodo	:	16 semanas
2.5 Condición	:	Obligatorio
2.6 Modalidad	:	Presencial
2.7 Prerrequisitos	:	CS112. Ciencia de la Computación I. (2 ^{do} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

El mundo ha cambiado debido al uso de la web y tecnologías relacionadas, el acceso rápido, oportuno y personalizado de la información, a través de la tecnología web, ubicuo y pervasiva; han cambiado la forma de ¿cómo hacemos las cosas?, ¿cómo pensamos? y ¿cómo la industria se desarrolla?.

Las tecnologías web, ubicuo y pervasivo se basan en el desarrollo de servicios web, aplicaciones web y aplicaciones móviles, las cuales son necesarias entender la arquitectura, el diseño, y la implementación de servicios web, aplicaciones web y aplicaciones móviles.

5. OBJETIVOS

- Que el alumno sea capaz de diseño e implementación de servicios, aplicaciones web utilizando herramientas y lenguajes como HTML, CSS, JavaScript (incluyendo AJAX) , back-end scripting y una base de datos, a un nivel intermedio.
- Que el alumno sea capaz de desarrollar aplicaciones móviles, administrar servidores web en sistemas basados en UNIX y aplicar técnicas de seguridad en la web a un nivel intermedio.

6. COMPETENCIAS

- c) Diseñar, implementar y evaluar un sistema, proceso, componente o programa computacional para alcanzar las necesidades deseadas. (**Usar**)
- d) Trabajar efectivamente en equipos para cumplir con un objetivo común. (**Usar**)
- g) Analizar el impacto local y global de la computación sobre los individuos, organizaciones y sociedad. (**Usar**)
- i) Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (**Usar**)
- c3) Utilizar distintas herramientas y lenguajes de programación en los componentes de software (*Full stack*). (3)
- c4) Diseñar e implementar arquitecturas de software escalables en distintas plataformas. (4)
- c5) Describir como el desarrollo basado en plataformas difiere del propósito general de programación. (5)
- c6) Aplicar las ventajas y desventajas de las restricciones de diversas plataformas. (6)
- c7) Aplicar o implementar las restricciones de las plataformas Web en el desarrollo de software. (7)
- c8) Aplicar los estándares de la web. (8)
- c9) Aplicar los estándares de desarrollo para dispositivos móviles. (9)
- c10) Implementar software como un servicio. (10)
- d1) Desarrollo colaborativo de software utilizando repositorios de código y gestión de versiones (ej. Git, Bitbucket, SVN). (1)

d2) Desarrollar presentaciones grupales e informes sobre tópicos específicos. (2)

g1) Desarrollar soluciones que resuelvan un problema existente en nuestra sociedad. (1)

i2) Utilización lenguajes y entornos de programación que permitan la implementación y depuración de las soluciones. (2)

7. TEMAS

Unidad 1: Introducción (5)	
Competencias esperadas: g	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Visión general de plataformas (ejemplo, Web, Mobil, Juegos, Industrial)• Programación a través de APIs específicos.• Visión general de lenguajes de plataforma (ejemplo, Objective C, HTML5)• Programación bajo restricciones de plataforma.	<ul style="list-style-type: none">• Describir cómo el desarrollo basado en plataforma difiere de la programación de propósito general [Familiarizarse]• Listar las características de lenguajes de plataforma [Familiarizarse]• Escribir y ejecutar un programa simple basado en plataforma [Familiarizarse]• Listar las ventajas y desventajas de la programación con restricciones de plataforma [Familiarizarse]
Lecturas : [grove2009web], [annuzzi2013introduction], [Cornez2015]	

Unidad 2: Plataformas web (5)	
Competencias esperadas: c,g,i	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Lenguajes de programación web (e.g., HTML5, Javascript, PHP, CSS)• Restricciones de las plataformas web: Client-Server, Stateless-Stateful, Caché, Uniform Interface, Layered System, Code on Demand, ReST.• Restricción de plataformas web.• Software como servicio.• Estándares web.	<ul style="list-style-type: none">• Diseñar e implementar una aplicación web sencilla [Familiarizarse]• Describir las limitaciones que la web pone a los desarrolladores [Familiarizarse]• Comparar y contrastar la programación web con la programación de propósito general [Familiarizarse]• Describir las diferencias entre software como un servicio y productos de software tradicionales [Familiarizarse]• Discutir cómo los estándares de web impactan el desarrollo de software [Familiarizarse]• Revisar una aplicación web existente con un estándar web actual [Familiarizarse]
Lecturas : [fielding2000fielding]	

Unidad 3: Desarrollo de servicios y aplicaciones web (25)	
Competencias esperadas: c,d,g,i	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Describir, identificar y depurar problemas relacionados con el desarrollo de aplicaciones web. • Diseño y desarrollo de aplicaciones web interactivas usando HTML5 y Python. • Utilice MySQL para la gestión de datos y manipular MySQL con Python. • Diseño y desarrollo de aplicaciones web asíncronos utilizando técnicas Ajax. • Uso del lado del cliente dinámico lenguaje de script Javascript y del lado del servidor lenguaje de scripting python con Ajax. • Aplicar las tecnologías XML / JSON para la gestión de datos. • Utilizar los servicios, APIs Web, Ajax y aplicar los patrones de diseño para el desarrollo de aplicaciones web. 	<ul style="list-style-type: none"> • Del lado del servidor lenguaje de scripting python: variables, tipos de datos, operaciones, cadenas, funciones, sentencias de control, matrices, archivos y el acceso a directorios, mantener el estado. [Usar] • Enfoque de programación web usando python incrustado. [Usar] • El acceso y la manipulación de MySQL. [Usar] • El enfoque de desarrollo de aplicaciones web Ajax. [Usar] • DOM y CSS utilizan en JavaScript. [Usar] • Tecnologías de actualización de contenido asíncrono. [Usar] • Objetos XMLHttpRequest utilizar para comunicarse entre clientes y servidores. [Usar] • XML y JSON. [Usar] • XSLT y XPath como mecanismos para transformar documentos XML. [Usar] • Servicios web y APIs (especialmente Google Maps). [Usar] • Marcos Ajax para el desarrollo de aplicaciones web contemporánea. [Usar] • Los patrones de diseño utilizados en aplicaciones web. [Usar]
Lecturas : [freeman2011head]	

Unidad 4: Plataformas móviles (5)	
Competencias esperadas: c,d,g,i	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Lenguajes de Programación para Móviles. • Principios de diseño: Segregación de Interfaces, Responsabilidad Única, Separación de Responsabilidades, Inversión de Dependencias. • Desafíos con movilidad y comunicación inalámbrica. • Aplicaciones Location-aware. • Rendimiento / Compensación de Potencia. • Restricciones de las Plataformas Móviles. • Tecnologías Emergentes. 	<ul style="list-style-type: none"> • Diseñar e implementar una aplicación móvil para una plataforma móvil dada [Familiarizarse] • Discutir las limitaciones que las plataformas móviles ponen a los desarrolladores [Familiarizarse] • Discutir los principios de diseño que guían la construcción de aplicaciones móviles [Familiarizarse] • Discutir el rendimiento vs pérdida de potencia [Familiarizarse] • Compare y contraste la programación móvil con la programación de propósito general [Familiarizarse]
Lecturas : [martin2017clean]	

Unidad 5: Aplicaciones Móviles para dispositivos Android (25)	
Competencias esperadas: c,d,g,i	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • The Android Platform • The Android Development Environment • Application Fundamentals • The Activity Class • The Intent Class • Permissions • The Fragment Class • User Interface Classes • User Notifications • The BroadcastReceiver Class • Threads, AsyncTask & Handlers • Alarms • Networking (http class) • Multi-touch & Gestures • Sensors • Location & Maps 	<ul style="list-style-type: none"> • Los estudiantes identifican software necesario y lo instalan en sus ordenadores personales. Los estudiantes realizan varias tareas para familiarizarse con la plataforma Android y Ambiente para el Desarrollo. [Usar] • Los estudiantes construyen aplicaciones que trazan los métodos de devolución de llamada de ciclo de vida emitidas por la plataforma Android y que demuestran el comportamiento de Android cuando los cambios de configuración de dispositivos (por ejemplo, cuando el dispositivo se mueve de vertical a horizontal y viceversa). [Usar] • Los estudiantes construyen aplicaciones que requieren iniciar múltiples actividades a través de ambos métodos estándar y personalizados. [Usar] • Los estudiantes construyen aplicaciones que requieren permisos estándar y personalizados. [Usar] • Los estudiantes construyen una aplicación que utiliza una única base de código, sino que crea diferentes interfaces de usuario dependiendo del tamaño de la pantalla de un dispositivo. [Usar] • Los estudiantes construyen un gestor de listas de tareas pendientes utilizando los elementos de la interfaz de usuario discutidos en clase. La aplicación permite a los usuarios crear nuevos elementos y para mostrarlos en un ListView. [Usar] • Los estudiantes construyen una aplicación que utiliza la información de ubicación para recoger latitud, longitud de los lugares que visitan. [Usar]
Lecturas : [annuzzi2013introduction], [Cornez2015]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

***** EVALUATION MISSING *****

10. BIBLIOGRAFÍA BÁSICA

Ministerio de Educación (MINEDU)
Sílabo 2021-I

1. CURSO

MA203. Estadística y Probabilidades (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	4
2.2 Horas de teoría	:	2 (Semanal)
2.3 Horas de práctica	:	2 (Semanal)
2.4 Duración del periodo	:	16 semanas
2.5 Condición	:	Obligatorio
2.6 Modalidad	:	Presencial
2.7 Prerrequisitos	:	MA100. Matemática I. (1 ^{er} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Provee de una introducción a la teoría de las probabilidades e inferencia estadística con aplicaciones, necesarias en el análisis de datos, diseño de modelos aleatorios y toma de decisiones.

5. OBJETIVOS

- Capacidad para diseñar y conducir experimentos, así como usar tecnología como para analizar e interpretar datos.
- Capacidad para identificar, formular y resolver problemas reales.

6. COMPETENCIAS

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Usar**)
- j) Aplicar la base matemática, principios de algoritmos y la teoría de la CS en el modelamiento y diseño de sistemas. (**Usar**)

■NoSpecificOutcomes■

7. TEMAS

Unidad 1: Tipo de variable (6)	
Competencias esperadas: C1	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Tipo de variable: Continua, discreta.	<ul style="list-style-type: none">• Clasificar las variables relevantes identificadas según su tipo: continuo (intervalo y razón), categórico (nominal, ordinario, dicotómico).• Identificar las variables relevantes de un sistema utilizando un enfoque de proceso.
Lecturas : [MRo14], [Men14]	

Unidad 2: Estadísticas descriptiva (6)	
Competencias esperadas: C1	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Tendencia Central (Media, mediana, modo) • Dispersión (Rango, desviación estándar, cuartil) • Gráficos: histograma, boxplot, etc. : Capacidad de comunicación. 	<ul style="list-style-type: none"> • Utilizar medidas de tendencia central y medidas de dispersión para describir los datos recopilados. • Utilizar gráficos para comunicar las características de los datos recopilados.
Lecturas : [MRo14], [Men14]	

Unidad 3: Estadística inferencial (6)	
Competencias esperadas: CS2	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Determinación del tamaño de la muestra • Intervalo de confianza • Tipo I y error del tipo II • Tipo de distribución • Prueba de hipótesis (t-student, medias, proporciones y ANOVA) • Relaciones entre variables: correlación, regresión. 	<ul style="list-style-type: none"> • Proponer preguntas e hipótesis de interés. • Analizar los datos recopilados utilizando diferentes herramientas estadísticas para responder preguntas de interés. • Dibujar conclusiones basadas en el análisis realizado.
Lecturas : [MRo14], [Men14]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

***** EVALUATION MISSING *****

10. BIBLIOGRAFÍA BÁSICA

- [Men14] Beaver Mendenhall. *Introducción a la probabilidad y estadística*. 13th. Cengage Learning, 2014.
- [MRo14] Sheldon M.Ross. *Introduction to Probability and Statistics for Engineers and Scientists*. 5th. Academic Press, 2014.

Ministerio de Educación (MINEDU)
Sílabo 2021-I

1. CURSO

FG203. Oratoria (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	2
2.2 Horas de teoría	:	1 (Semanal)
2.3 Horas de práctica	:	-
2.4 Duración del periodo	:	16 semanas
2.5 Condición	:	Obligatorio
2.6 Modalidad	:	Presencial
2.7 Prerrequisitos	:	FG106. Teatro. (2 ^{do} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

En la sociedad competitiva como la nuestra, se exige que la persona sea un comunicador eficaz y sepa utilizar sus potencialidades a fin de resolver problemas y enfrentar los desafíos del mundo moderno dentro de la actividad laboral, intelectual y social. Tener el conocimiento no basta, lo importante es saber comunicarlo y en la medida que la persona sepa emplear sus facultades comunicativas, derivará en éxito o fracaso aquello que tenga que realizar en su desenvolvimiento personal y profesional. Por ello es necesario para lograr un buen decir, recurrir a conocimientos, estrategias y recursos, que debe tener todo orador, para llegar con claridad, precisión y convicción al interlocutor

5. OBJETIVOS

- Al término del curso, el alumno será capaz de organizar y asumir la palabra desde la perspectiva del orador, en cualquier situación, en forma más correcta, coherente y adecuada, mediante el uso de conocimientos y habilidades lingüísticas, buscando en todo momento su realización personal y social a través de su expresión, teniendo como base la verdad y la preparación constante.

6. COMPETENCIAS

f) Comunicarse efectivamente con audiencias diversas. (**Usar**)

n) Aplicar conocimientos de humanidades en su labor profesional. (**Usar**)

ñ) Comprender que la formación de un buen profesional no se desliga ni se opone sino mas bien contribuye al auténtico crecimiento personal. Esto requiere de la asimilación de valores sólidos, horizontes espirituales amplios y una visión profunda del entorno cultural. (**Usar**)

■NoSpecificOutcomes■

7. TEMAS

Unidad 1: (3)	
Competencias esperadas: C24	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • La Oratoria • La función de la palabra. • El proceso de la comunicación. • Bases racionales y emocionales de la oratoria <ul style="list-style-type: none"> – La expresión oral en la participación. • Fuentes de conocimiento para la oratoria: niveles de cultura general. 	<ul style="list-style-type: none"> • Comprensión: interpretar, ejemplificar y generalizar las bases de la oratoria como fundamento teórico y práctico. [Usar].
Lecturas : [ME76], [Rod]	

Unidad 2: (4)	
Competencias esperadas: C17	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Cualidades de un buen orador. • Normas para primeros discursos. • El cuerpo humano como instrumento de comunicación: <ul style="list-style-type: none"> – La expresión corporal en el discurso – La voz en el discurso. • Oradores con historia y su ejemplo. 	<ul style="list-style-type: none"> • Comprensión: Interpretar, ejemplificar y generalizar conocimientos y habilidades de la comunicación oral mediante la experiencia de grandes oradores y la suya propia. [Usar]. • Aplicación: Implementar, usar, elegir y desempeñar los conocimientos adquiridos para expresarse en público en forma eficiente, inteligente y agradable. [Usar].
Lecturas : [Rod]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

***** EVALUATION MISSING *****

10. BIBLIOGRAFÍA BÁSICA

[ME76] A. Monroe and D. Ehninger. *La comunicación oral*. Hispano Europea, 1976.

[Rod] María L. Rodríguez. *Cómo manejar la información en una presentación*.

Ministerio de Educación (MINEDU)
Sílabo 2021-I

1. CURSO

ID201. Inglés III (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	3
2.2 Horas de teoría	:	-
2.3 Horas de práctica	:	-
2.4 Duración del periodo	:	16 semanas
2.5 Condición	:	Obligatorio
2.6 Modalidad	:	Presencial
2.7 Prerrequisitos	:	ID102. Inglés II. (2 ^{do} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Parte fundamental de la formación integral de un profesional es la habilidad de comunicarse en un idioma extranjero además del propio idioma nativo. No solamente amplía su horizonte cultural sino que permite una visión más humana y comprensiva de la vida. En el caso de los idiomas extranjeros, indudablemente el Inglés es el más práctico porque es hablado alrededor de todo el mundo. No hay país alguno donde este no sea hablado. En las carreras relacionadas con los servicios al turista el inglés es tal vez la herramienta práctica más importante que el alumno debe dominar desde el primer momento como parte de su formación integral.

5. OBJETIVOS

- Formar en el alumno de capacidad de comprender y retener una conversación.
- Brindar técnicas de ilación de ideas.

6. COMPETENCIAS

f) Comunicarse efectivamente con audiencias diversas. (Usar)

■NoSpecificOutcomes■

7. TEMAS

Unidad 1: Getting to know you! (0)	
Competencias esperadas: C25	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Tiempos Presente, Pasado y Futuro.• Oraciones Interrogativas con Wh-.• Palabras con más de un significado.• Partes de la oración.• Expresiones para tiempo libre.	<ul style="list-style-type: none">• Al terminar la primera unidad, cada uno de los alumnos, comprendiendo la gramática de los tiempos presente, pasado y futuro es capaz de expresar una mayor cantidad de acciones en forma de oraciones. Además es capaz de expresar ideas en forma de preguntas. Asume la idea de palabras con más de un significado. Utiliza expresiones sociales en situaciones de entretenimiento.
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 2: The way we live! (0)	
Competencias esperadas: C25	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Tiempo Presente Simple. • Tiempo Presente Continuo. • Colocaciones. • Vocabulario de países del mundo. • Expresiones de enojo. • Conectores. 	<ul style="list-style-type: none"> • Al terminar la segunda unidad, los alumnos habiendo identificado la forma de expresar presente reconocen la diferencia entre las formas del mismo y las aplican adecuadamente. Describen países acuciosamente. Asumen expresiones para demostrar interés. Utilizan conectores para unir ideas varias.
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 3: It all went wrong! (0)	
Competencias esperadas: C25	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Tiempo Pasado Simple. • Tiempo Pasado Continuo. • Verbos Irregulares. • Expresiones de Tiempo. • Conectores de tiempo. 	<ul style="list-style-type: none"> • Al terminar la tercera unidad, los alumnos habiendo reconocido las características de los tiempos en pasado los utilizan adecuadamente. Utilizan prefijos y sufijos para crear y reconocer nuevas palabras. Describen tiempo en forma amplia. Utilizarán conjunciones para unir ideas tipo.
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 4: Lets go shopping! (0)	
Competencias esperadas: C25	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Expresiones de Cantidad Indefinida • Oraciones Afirmativas, Negativas y Preguntas • Uso de Artículos • Precios de productos • Llenado de formatos y encuestas • Expresiones para ir de compras 	<ul style="list-style-type: none"> • Al terminar la cuarta primera unidad, los alumnos habiendo identificado la idea de cantidad expresan diversas situaciones que la involucran. Reconocen y aplican artículos a sustantivos. Asumen la idea de ir de compras con la ayuda de expresiones. Expresan precios e ideas de dinero. Llenan formatos varios. Expresan actitudes.
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 5: What do you want to do? (0)	
Competencias esperadas: C25	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Patrones Verbales I. • Intenciones Futuras. • Verbos de Percepción. • Vocabulario de sentimientos. • Expresiones de Planes y Ambiciones. 	<ul style="list-style-type: none"> • Al finalizar la quinta unidad, los alumnos, a partir de la comprensión de la idea de patrones verbales elaborarán oraciones utilizando los elementos necesarios. Asimilarán además la necesidad de expresar intenciones futuras. Adquirirán vocabulario para describir sentimientos. Se presentará expresiones para describir planes y ambiciones.
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 6: The best in the world! (0)	
Competencias esperadas: C25	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Whats it like?. • Adjetivos. • Comparativos y Superlativos. • Sinónimos y Antónimos. • Indicaciones de Dirección. • Lecturas. 	<ul style="list-style-type: none"> • Al finalizar la sexta unidad, los alumnos habiendo conocido los fundamentos del uso de adjetivos, estructuran oraciones con diversas formas de los mismos en contextos adecuados. Enfatizan la diferencia entre tipos de ciudades y pueblos y estilos de vida. Utilizan expresiones indicación de direcciones.
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 7: Fame! (0)	
Competencias esperadas: C25	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Presente Perfecto y Pasado Simple • Expresiones for, ever, since • Adverbios • Expresiones que vienen en pares • Respuestas cortas • Celebridades 	<ul style="list-style-type: none"> • Al finalizar la séptima unidad, los alumnos habiendo conocido los fundamentos de la estructuración del tiempo presente perfecto y lo diferencian del pasado simple. Enfatizan la diferencia entre formas de adjetivos. Describen ideas de la música. Utilizan expresiones para dar respuestas cortas. Asumen la idea de dar explicaciones extra de los elementos de una oración.
Lecturas : [SJ02], [Cam06], [Mac99]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

***** EVALUATION MISSING *****

10. BIBLIOGRAFÍA BÁSICA

- [Cam06] Cambridge. *Diccionario Inglés-Español Cambridge*. Editorial Oxford, 2006.
[Mac99] James MacGrew. *Focus on Grammar Basic*. Editorial Oxford, 1999.
[SJ02] Liz Soars and John. *American Headway N 2 Student Book*. Editorial Oxford, 2002.

Ministerio de Educación (MINEDU)
Sílabo 2021-I

1. CURSO

CS210. Algoritmos y Estructuras de Datos (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	4
2.2 Horas de teoría	:	2 (Semanal)
2.3 Horas de práctica	:	2 (Semanal)
2.4 Duración del periodo	:	16 semanas
2.5 Condición	:	Obligatorio
2.6 Modalidad	:	Presencial
2.7 Prerrequisitos	:	CS113. Ciencia de la Computación II. (3 ^{er} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

El fundamento teórico de todas las ramas de la informática descansa sobre los algoritmos y estructuras de datos, este curso brindará a los participantes una introducción a estos temas, formando así una base que servirá para los siguientes cursos en la carrera.

5. OBJETIVOS

- Hacer que el alumno entienda la importancia de los algoritmos para la solución de problemas.
- Introducir al alumno hacia el campo de la aplicación de las estructuras de datos.

6. COMPETENCIAS

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Usar**)
 - b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Usar**)
 - c) Diseñar, implementar y evaluar un sistema, proceso, componente o programa computacional para alcanzar las necesidades deseadas. (**Usar**)
-
- a4) Aplicar técnicas eficientes de resolución de problemas computacionales. (4)
 - b1) Identificar y aplicar de forma eficiente diversas estrategias algorítmicas y estructuras de datos para la solución de un problema dadas ciertas restricciones de espacio y tiempo. (1)
 - c1) Identificar e implementar estructuras de datos para la solución de un problema computacional. (1)

7. TEMAS

Unidad 1: Grafos (12)	
Competencias esperadas: a,b,c	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Concepto de Grafos. • Grafos Dirigidos y Grafos no Dirigidos. • Utilización de los Grafos. • Medida de la Eficiencia. En tiempo y espacio. • Matrices de Adyacencia. • Matrices de Adyacencia etiquetada. • Listas de Adyacencia. • Implementación de Grafos usando Matrices de Adyacencia. • Implementación de Grafos usando Listas de Adyacencia. • Inserción, Búsqueda y Eliminación de nodos y aristas. • Algoritmos de búsqueda en grafos. 	<ul style="list-style-type: none"> • Adquirir destreza para realizar una implementación correcta. [Usar] • Desarrollar los conocimientos para decidir cuando es mejor usar una técnica de implementación que otra. [Usar]
Lecturas : [Cor+09], [Fag+14], [Knu97], [Knu98]	

Unidad 2: Matrices Esparzas (8)	
Competencias esperadas: a,b,c	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Conceptos Iniciales. • Matrices poco densas • Medida de la Eficiencia en Tiempo y en Espacio • Creación de la matriz esparza estática vs Dinámicas. • Métodos de inserción, búsqueda y eliminación 	<ul style="list-style-type: none"> • Comprender el uso y implementación de matrices esparzas.[Evaluar]
Lecturas : [Cor+09], [Fag+14], [Knu97], [Knu98]	

Unidad 3: Árboles Equilibrados (16)	
Competencias esperadas: a,b,c	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Árboles AVL. • Medida de la Eficiencia. • Rotaciones Simples y Compuestas • Inserción, Eliminación y Búsqueda. • Árboles B , B+ B* y Patricia. 	<ul style="list-style-type: none"> • Comprender las funciones básicas de estas estructuras complejas con el fin de adquirir la capacidad para su implementación. [Evaluar]
Lecturas : [Cor+09], [Fag+14], [Knu97], [Knu98]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

***** EVALUATION MISSING *****

10. BIBLIOGRAFÍA BÁSICA

- [Cor+09] Thomas H. Cormen et al. *Introduction to Algorithms*. Third Edition. ISBN: 978-0-262-53305-8. MIT Press, 2009.
- [Fag+14] José Fager et al. *Estructura de datos*. First Edition. Iniciativa Latinoamericana de Libros de Texto Abiertos (LATIN), 2014.
- [Knu97] Donald E. Knuth. *The Art of Computer Programming, Vol. 1: Fundamental Algorithms*. 3rd. Addison-Wesley Professional, 1997.
- [Knu98] Donald E. Knuth. *The art of computer programming, volume 3:Sorting and searching*. 2nd. Addison-Wesley Professional, 1998.

Ministerio de Educación (MINEDU)
Sílabo 2021-I

1. CURSO

CS271. Gerenciamiento de Datos I (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	4
2.2 Horas de teoría	:	2 (Semanal)
2.3 Horas de práctica	:	4 (Semanal)
2.4 Duración del periodo	:	16 semanas
2.5 Condición	:	Obligatorio
2.6 Modalidad	:	Presencial
2.7 Prerrequisitos	:	<ul style="list-style-type: none">• CS112. Ciencia de la Computación I. (2^{do} Sem)• CS1D2. Estructuras Discretas II. (2^{do} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

La gestión de la información (IM) juega un rol principal en casi todas las áreas donde los computadores son usados. Esta área incluye la captura, digitalización, representación, organización, transformación y presentación de información; algoritmos para mejorar la eficiencia y efectividad del acceso y actualización de información almacenada, modelamiento de datos y abstracción, y técnicas de almacenamiento de archivos físicos.

Este también abarca la seguridad de la información, privacidad, integridad y protección en un ambiente compartido. Los estudiantes necesitan ser capaces de desarrollar modelos de datos conceptuales y físicos, determinar que métodos de (IM) y técnicas son apropiados para un problema dado, y ser capaces de seleccionar e implementar una apropiada solución de IM que refleje todas las restricciones aplicables, incluyendo escalabilidad y usabilidad.

5. OBJETIVOS

- Que el alumno aprenda a representar información en una base de datos priorizando la eficiencia en la recuperación de la misma
- Que el alumno aprenda los conceptos fundamentales de gestión de bases de datos. Esto incluye aspectos de diseño de bases de datos, lenguajes de bases de datos y realización de bases de datos
- Discutir el modelo de bases de datos con base en el álgebra relacional, cálculo relacional y en el estudio de sentencias SQL.

6. COMPETENCIAS

- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Usar**)
- d) Trabajar efectivamente en equipos para cumplir con un objetivo común. (**Usar**)
- i) Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (**Evaluar**)
- j) Aplicar la base matemática, principios de algoritmos y la teoría de la CS en el modelamiento y diseño de sistemas. (**Evaluar**)
- b5) Modelar base de datos a través de modelos ER, MR, optimización, transacciones y recuperación de la información. (5)
- d2) Desarrollar presentaciones grupales e informes sobre tópicos específicos. (2)
- d3) Desarrollar trabajo en grupo en cada tópico del curso. (3)
- i3) Utilizar de forma apropiada los módulos de optimización de consultas, desempeño, indexación y fragmentación de tablas para BD distribuídas utilizando un motor de bases de datos de código abierto como PostgreSQL, Cassandra o MongoDB. (3)

j2) Aplicar teoría de grafos y árboles para la optimización y resolución de problemas. (2)

j3) Utilizar de forma adecuada herramientas como Relax *Relational Algebra Calculator* (<https://dbis-uibk.github.io/relax/calc.htm>) para la verificación del álgebra relacional de una consulta. (3)

7. TEMAS

Unidad 1: Sistemas de Bases de Datos (14)	
Competencias esperadas: b,d,i,j	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Enfoque y Evolución de Sistemas de Bases de Datos. • Componentes del Sistema de Bases de Datos. • Diseño de las funciones principales de un DBMS. • Arquitectura de base de datos e independencia de datos. • Uso de un lenguaje de consulta declarativa. • Sistemas de apoyo a contenido estructurado y / o corriente. • Enfoques para la gestión de grandes volúmenes de datos (por ejemplo, sistemas de bases de datos NoSQL, uso de MapReduce). 	<ul style="list-style-type: none"> • Explica las características que distinguen un esquema de base de datos de aquellos basados en la programación de archivos de datos [Usar] • Describe los diseños más comunes para los componentes base de sistemas de bases de datos incluyendo el optimizador de consultas, ejecutor de consultas, administrador de almacenamiento, métodos de acceso y procesador de transacciones [Usar] • Cita las metas básicas, funciones y modelos de un sistema de bases de datos [Usar] • Describe los componentes de un sistema de bases de datos y da ejemplos de su uso [Usar] • Identifica las funciones principales de un SGBD y describe sus roles en un sistema de bases de datos [Usar] • Explica los conceptos de independencia de datos y su importancia en un sistema de bases de datos [Usar] • Usa un lenguaje de consulta declarativo para recoger información de una base de datos [Usar] • Describe las capacidades que las bases de datos brindan al apoyar estructuras y/o la secuencia de flujo de datos, ejm. texto [Usar] • Describe los enfoques principales para almacenar y procesar largos volúmenes de datos [Usar]
Lecturas : [RC04], [EN04], [RG03], [ER15], [CJ11], [KS02]	

Unidad 2: Modelado de datos (14)	
Competencias esperadas: b,d,i,j	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Modelado de datos • Modelos conceptuales (e.g., entidad-relación, diagramas UML) • Modelos de hoja de cálculo • Modelos Relacionales. • Modelos orientados a objetos. • Modelos de datos semi-estructurados (expresados usando DTD o XML Schema, por ejemplo) 	<ul style="list-style-type: none"> • Compare y contrasta modelos apropiados de datos, incluyendo estructuras sus estructuras internas, para diversos tipos de datos [Usar] • Describe los conceptos en notación de modelos (ejm. Diagramas Entidad-Relación o UML) y cómo deben de ser usados [Usar] • Define la terminología fundamental a ser usada en un modelo relacional de datos [Usar] • Describe los principios básicos del modelo relacional de datos [Usar] • Aplica los conceptos de modelado y la notación de un modelo relacional de datos [Usar] • Describe los conceptos principales del modelado OO como son identidad de objetos, constructores de tipos, encapsulación, herencia, polimorfismo, y versiones [Usar] • Describe las diferencias entre modelos de datos relacionales y semi-estructurados [Usar] • Da una semi estructura equivalente (ejm. en DTD o Esquema XML) para un esquema relacional dado [Usar]
Lecturas : [SW04], [EN04], [KS02]	

Unidad 3: Indexación (4)	
Competencias esperadas: b,d,i	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • El impacto de índices en el rendimiento de consultas. • La estructura básica de un índice. • Mantener un buffer de datos en memoria. • Creando índices con SQL. • Indexando texto. • Indexando la web (e.g., web crawling) 	<ul style="list-style-type: none"> • Generar un archivo índice para una colección de recursos [Usar] • Explicar la función de un índice invertido en la localización de un documento en una colección [Usar] • Explicar cómo rechazar y detener palabras que afectan a la indexación [Usar] • Identificar los índices adecuados para determinado el esquema relacional y el conjunto de consultas [Usar] • Estimar el tiempo para recuperar información, cuando son usados los índices comparado con cuando no son usados [Usar] • Describir los desafíos claves en el rastreo web, por ejemplo, la detección de documentos duplicados, la determinación de la frontera de rastreo [Usar]
Lecturas : [WM01], [RG03], [ER15], [CJ11], [KS02]	

Unidad 4: Bases de Datos Relacionales (14)**Competencias esperadas: b,d,i**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Mapeo de esquemas conceptuales a esquemas relacionales.• Entidad y integridad referencial.• Algebra relacional y calculo relacional.• Diseño de bases de datos relacionales.• Dependencia funcional.• Descomposición de un esquema.• Llaves candidatas, SuperLlaves y cierre de un conjunto de atributos.• Formas Normales (BCNF)• Dependencias multi-valoradas (4NF)• Uniendo dependencias (PJNF, 5NF)• Teoría de la representación.	<ul style="list-style-type: none">• Prepara un esquema relacional de un modelo conceptual desarrollado usando el modelo entidad-relación [Usar]• Explica y demuestra los conceptos de restricciones de integridad de la entidad e integridad referencial (incluyendo la definición del concepto de clave foránea) [Usar]• Demuestra el uso de las operaciones de álgebra relacional de la teoría matemática de conjuntos (unión, intersección, diferencia, y producto Cartesiano) y de las operaciones de álgebra relacional desarrolladas específicamente para las bases de datos relacionales (selección (restringida), proyección, unión y división) [Usar]• Escribe consultas en álgebra relacional [Usar]• Escribe consultas en cálculo relacional de tuplas [Usar]• Determina la dependencia funcional entre dos o más atributos que son subconjunto de una relación [Usar]• Conecta restricciones expresadas como clave primaria y foránea, con dependencias funcionales [Usar]• Calcula la cerradura de un conjunto de atributos dado dependencias funcionales [Usar]• Determina si un conjunto de atributos forma una superclave y/o una clave candidata de una relación dada dependencias funcionales [Usar]• Evalua una descomposición propuesta, a fin de determinar si tiene una unión sin pérdidas o preservación de dependencias [Usar]• Describe las propiedades de la FNBC, FNUP (forma normal unión de proyecto), 5FN [Usar]• Explica el impacto de la normalización en la eficacia de las operaciones de una base de datos especialmente en la optimización de consultas [Usar]• Describe que es una dependencia de multi valor y cual es el tipo de restricciones que especifica [Usar]
Lecturas : [WM01], [RG03], [ER15], [CJ11], [KS02]	

Unidad 5: Lenguajes de Consulta (12)	
Competencias esperadas: b,d,i,j	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Visión general de lenguajes de base de datos. • SQL (definición de datos, formulacion de consultas, sublenguaje update, restricciones, integridad) • Selecciones • Proyecciones • Select-project-join • Agregaciones y agrupaciones. • Subconsultas. • Entornos QBE de cuarta generación. • Diferentes maneras de invocar las consultas no procedimentales en lenguajes convencionales. • Introducción a otros lenguajes importantes de consulta (por ejemplo, XPATH, SPARQL) • Procedimientos almacenados. 	<ul style="list-style-type: none"> • Crear un esquema relacional de bases de datos en SQL que incorpora restricciones clave y restricciones de integridad de entidad e integridad referencial [Usar] • Usar SQL para crear tablas y devuelve (SELECT) la información de una base de datos [Usar] • Evaluar un conjunto de estrategias de procesamiento de consultas y selecciona la estrategia óptima [Usar] • Crear una consulta no-procedimental al llenar plantillas de relaciones para construir un ejemplo del resultado de una consulta requerida [Usar] • Adicionar consultas orientadas a objetos en un lenguaje stand-alone como C++ o Java (ejm. SELECT ColMethod() FROM Objeto) [Usar] • Escribe un procedimiento almacenado que trata con parámetros y con algo de flujo de control de tal forma que tenga funcionalidad [Usar]
Lecturas : [Die01], [EN04], [Cel05], [KS02]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

***** EVALUATION MISSING *****

10. BIBLIOGRAFÍA BÁSICA

- [Cel05] Joe Celko. *Joe Celko's SQL Programming Style*. Elsevier, 2005.
- [CJ11] Date C.J. *SQL and Relational Theory: How to Write Accurate SQL Code*. O'Reilly Media, 2011.
- [Die01] Suzanne W Dietrich. *Understanding Relational Database Query Languages, First Edition*. Prentice Hall, 2001.
- [EN04] Ramez Elmasri and Shamkant B. Navathe. *Fundamentals of Database Systems, Fourth Edition*. Addison Wesley, 2004.
- [ER15] Jim Webber Emil Eifrem and Ian Robinson. *Graph Databases*. 2nd. O'Reilly Media, 2015.
- [KS02] Henry F. Korth and Abraham Silberschatz. *Fundamentos de Base de Datos*. McGraw-Hill, 2002.

- [RC04] Peter Rob and Carlos Coronel. *Database Systems: Design, Implementation and Management, Sixth Edition*. Morgan Kaufmann, 2004.
- [RG03] Raghu Ramakrishnan and Johannes Gehrke. *Database Management Systems*. 3rd. McGraw-Hill, 2003.
- [SW04] Graeme Simsion and Graham Witt. *Data Modeling Essentials, Third Edition*. Morgan Kaufmann, 2004.
- [WM01] Mark Whitehorn and Bill Marklyn. *Inside Relational Databases, Second Edition*. Springer, 2001.

Ministerio de Educación (MINEDU)
Sílabo 2021-I

1. CURSO

CS2S1. Sistemas Operativos (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	4
2.2 Horas de teoría	:	2 (Semanal)
2.3 Horas de práctica	:	2 (Semanal)
2.4 Duración del periodo	:	16 semanas
2.5 Condición	:	Obligatorio
2.6 Modalidad	:	Presencial
2.7 Prerrequisitos	:	CS221. Arquitectura de Computadores. (3 ^{er} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Un sistema operativo (SO) gestiona los recursos computacionales para completar la ejecución de múltiples aplicaciones y sus procesos asociados. Este curso enseña el diseño de sistemas operativos modernos; e introduce sus conceptos fundamentales que cubren la ejecución multi-programa, *scheduling*, gerencia de memoria, sistemas de archivos y seguridad. Además, el curso incluye actividades de programación en un *sistema operativo mínimo* para resolver problemas y ampliar su funcionalidad. Tenga en cuenta que estas actividades requieren mucho tiempo para completarse. Sin embargo, trabajar en ellos proporciona un valioso aprendizaje sobre los sistemas operativos.

5. OBJETIVOS

- Estudiar el diseño de sistemas operativos modernos.
- Proveer una experiencia práctica al diseñar e implementar un sistema operativo mínimo.

6. COMPETENCIAS

- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Evaluar**)

■NoSpecificOutcomes■

7. TEMAS

Unidad 1: Visión general de Sistemas Operativos (3)	
Competencias esperadas: b	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Papel y el propósito del sistema operativo. • Funcionalidad de un sistema operativo típico. • Los mecanismos de apoyo modelos cliente-servidor. • Cuestiones de diseño (eficiencia, robustez, flexibilidad, portabilidad, seguridad, compatibilidad) • Influencias de seguridad, creación de redes, multimedia, sistemas de ventanas. 	<ul style="list-style-type: none"> • Explicar los objetivos y funciones de un sistema operativo moderno [Familiarizarse] • Analizar las ventajas y desventajas inherentes en el diseño de un sistema operativo [Evaluar] • Describir las funciones de un sistema operativo contemporáneo respecto a conveniencia, eficiencia, y su habilidad para evolucionar [Familiarizarse] • Discutir acerca de sistemas operativos cliente-servidor, en red, distribuidos y cómo se diferencian de los sistemas operativos de un solo usuario [Familiarizarse] • Identificar amenazas potenciales a sistemas operativos y las características del diseño de seguridad para protegerse de ellos [Familiarizarse]
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unidad 2: Principios de Sistemas Operativos (6)	
Competencias esperadas: b	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Estructuración de Sistemas Operativos (monolítico, capas, modular, los modelos micro-kernel) • Abstracciones, procesos y recursos. • Los conceptos de interfaces de programa de aplicación (API) • La evolución de las técnicas de hardware / software y las necesidades de aplicación • Organización de dispositivos. • Interrupciones: métodos e implementaciones. • Concepto de estado de usuario / sistema y la protección, la transición al modo kernel. 	<ul style="list-style-type: none"> • Explicar el concepto de una capa lógica [Familiarizarse] • Explicar los beneficios de construir capas abstractas en forma jerárquica [Familiarizarse] • Describir el valor de la API y <i>middleware</i> [Familiarizarse] • Describir como los recursos computacionales son usados por aplicaciones de software y administradas por el software del sistema [Familiarizarse] • Contrastar el modo <i>kernel</i> y modo usuario en un sistema operativo [Evaluar] • Discutir las ventajas y desventajas del uso de procesamiento con interrupciones [Familiarizarse] • Explicar el uso de una lista de dispositivos y el controlador de colas de entrada y salida [Familiarizarse]
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unidad 3: Concurrency (9)**Competencias esperadas: b**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Diagramas de estado.• Estructuras (lista preparada, bloques de control de procesos, y así sucesivamente)• <i>Dispatching</i> y cambio de contexto.• El papel de las interrupciones.• Gestionar el acceso a los objetos del sistema operativo de forma atómica.• La implementación de primitivas de sincronización.• Problemas de multiprocesador (spin-locks, reentrada)	<ul style="list-style-type: none">• Describir la necesidad de concurrencia en el marco de un sistema operativo [Familiarizarse]• Demostrar los potenciales problemas de tiempo de ejecución derivados de la operación simultánea de muchas tareas diferentes [Usar]• Resumir el rango de mecanismos que pueden ser usados a nivel del sistema operativo para realizar sistemas concurrentes y describir los beneficios de cada uno [Familiarizarse]• Explicar los diferentes estados por los que una tarea debe pasar y las estructuras de datos necesarias para el manejo de varias tareas [Familiarizarse]• Resumir las técnicas para lograr sincronización en un sistema operativo (por ejemplo, describir como implementar semáforos usando primitivas del sistema operativo.) [Familiarizarse]• Describir las razones para usar interrupciones, <i>dispatching</i>, y cambio de contexto para soportar concurrencia en un sistema operativo [Familiarizarse]• Crear diagramas de estado y transición para los problemas de dominios simples [Usar]
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unidad 4: Planificación y despacho (6)**Competencias esperadas: b**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• <i>Scheduling preemptive</i> y <i>non-preemptive</i>.• <i>Scheduling</i> y políticas.• Procesos y subprocesos.• Plazos y cuestiones en tiempo real.	<ul style="list-style-type: none">• Comparar y contrastar los algoritmos comunes que se utilizan tanto para <i>scheduling preemptive</i> y <i>pre-emptive</i> de tareas en los sistemas operativos, como la comparación de prioridad, el rendimiento, y los esquemas de distribución equitativa [Evaluar]• Describir las relaciones entre los algoritmos de <i>scheduling</i> y dominios de aplicación [Familiarizarse]• Discutir los tipos de <i>scheduling</i> en procesadores en de corto, mediano, largo plazo y I/O [Familiarizarse]• Describir las diferencias entre procesos y <i>threads</i> [Familiarizarse]• Comparar y contrastar enfoques estáticos y dinámicos para <i>scheduling</i> en tiempo real [Evaluar]• Discutir sobre la necesidad de <i>preemption</i> y <i>deadline scheduling</i> [Familiarizarse]• Identificar formas en que la lógica expresada en algoritmos de planificación son de aplicación a otros ámbitos, tales como I/O del disco, la programación de disco de red, programación de proyectos y problemas más allá de la computación [Familiarizarse]
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unidad 5: Manejo de memoria (6)	
Competencias esperadas: b	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Revisión de la memoria física y hardware de gestión de memoria. • Conjuntos de trabajo y thrashing. • El almacenamiento en caché 	<ul style="list-style-type: none"> • Explicar la jerarquía de la memoria y <i>tradeoffs</i> de costo-rendimiento [Familiarizarse] • Resumir los principios de memoria virtual tal como se aplica para el almacenamiento en cache y paginación [Familiarizarse] • Evaluar las ventajas y desventajas en términos del tamaño de memoria (memoria principal, memoria caché, memoria auxiliar) y la velocidad del procesador [Evaluar] • Describir las diferentes formas de asignar memoria a las tareas, citando las ventajas relativas de cada uno [Familiarizarse] • Describir el motivo y el uso de memoria caché (rendimiento y proximidad, dimensión diferente de como los caches complican el aislamiento y abstracción en VM) [Familiarizarse] • Estudiar los conceptos de <i>thrashing</i>, tanto en términos de las razones por las que se produce y las técnicas usadas para el reconocimiento y manejo del problema [Familiarizarse]
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unidad 6: Seguridad y protección (6)	
Competencias esperadas: b	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Visión general de la seguridad del sistema . • Política / mecanismo de separación. • Métodos de seguridad y dispositivos. • Protección, control de acceso y autenticación. • Las copias de seguridad. 	<ul style="list-style-type: none"> • Explicar la necesidad para la protección y seguridad en un sistema operativo [Familiarizarse] • Resumir las características y limitaciones de un sistema operativo usado para proporcionar protección y seguridad [Familiarizarse] • Explicar el mecanismo disponible en un OS para controlar los accesos a los recursos [Familiarizarse] • Realizar tareas de administración de sistemas sencillas de acuerdo a una política de seguridad, por ejemplo la creación de cuentas, el establecimiento de permisos, aplicación de parches y organización de backups regulares [Familiarizarse]
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unidad 7: Máquinas virtuales (6)	
Competencias esperadas: b	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Tipos de virtualización (incluyendo Hardware / Software, OS, Servidor, Servicio, Red) • Paginación y la memoria virtual. • Sistemas de archivos virtuales. • Los Hypervisores. • Virtualización portátil; emulación vs aislamiento. • Costo de la virtualización. 	<ul style="list-style-type: none"> • Explicar el concepto de memoria virtual y la forma cómo se realiza en hardware y software [Familiarizarse] • Diferenciar emulación y el aislamiento [Familiarizarse] • Evaluar virtualización de compensaciones [Evaluar] • Discutir sobre hipervisores y la necesidad para ellos en conjunto con diferentes tipos de hipervisores [Familiarizarse]
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unidad 8: Manejo de dispositivos (6)	
Competencias esperadas: b	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Características de los dispositivos serie y paralelo. • Haciendo de abstracción de dispositivos. • Estrategias de buffering. • Acceso directo a memoria. • La recuperación de fallos. 	<ul style="list-style-type: none"> • Explique la diferencia clave entre dispositivos seriales y paralelos e identificar las condiciones en las cuales cada uno es apropiado [Familiarizarse] • Identificar la relación entre el hardware físico y los dispositivos virtuales mantenidos por el sistema operativo [Familiarizarse] • Explicar <i>buffering</i> y describir las estrategias para su aplicación [Familiarizarse] • Diferenciar los mecanismos utilizados en la interconexión de un rango de dispositivos (incluyendo dispositivos portátiles, redes, multimedia) a un ordenador y explicar las implicaciones de éstas para el diseño de un sistema operativo [Familiarizarse] • Describir las ventajas y desventajas de acceso directo a memoria y discutir las circunstancias en cuales se justifica su uso [Familiarizarse] • Identificar los requerimientos para recuperación de errores [Familiarizarse] • Implementar un controlador de dispositivo simple para una gama de posibles equipos [Usar]
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unidad 9: Sistema de archivos (6)	
Competencias esperadas: b	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Archivos: los datos, metadatos, operaciones, organización, amortiguadores, secuenciales, no secuencial. • Directorios: contenido y estructura. • Los sistemas de archivos: partición, montar / desmontar sistemas de archivos virtuales. • Técnicas estándar de implementación . • Archivos asignados en memoria. • Sistemas de archivos de propósito especial. • Naming, búsqueda, acceso, copias de seguridad. • La bitacora y los sistemas de archivos estructurados (log) 	<ul style="list-style-type: none"> • Describir las decisiones que deben tomarse en el diseño de sistemas de archivos [Familiarizarse] • Comparar y contrastar los diferentes enfoques para la organización de archivos, el reconocimiento de las fortalezas y debilidades de cada uno. [Evaluar] • Resumir cómo el desarrollo de hardware ha dado lugar a cambios en las prioridades para el diseño y la gestión de sistemas de archivos [Familiarizarse] • Resumir el uso de diarios y como los sistemas de archivos de registro estructurado mejora la tolerancia a fallos [Familiarizarse]
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unidad 10: Sistemas <i>embedded</i> y de tiempo real (6)	
Competencias esperadas: b	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Proceso y programación de tareas. • Los requisitos de gestión de memoria / disco en un entorno en tiempo real. • Los fracasos, los riesgos y la recuperación. • Preocupaciones especiales en sistemas de tiempo real. 	<ul style="list-style-type: none"> • Describir que hace a un sistema un sistema en tiempo real [Familiarizarse] • Explicar la presencia y describir las características de latencia en sistemas de tiempo real [Familiarizarse] • Resumir los problemas especiales que los sistemas en tiempo real presentan, incluyendo el riesgo, y cómo se tratan estos problemas [Familiarizarse]
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unidad 11: Tolerancia a fallas (3)	
Competencias esperadas: b	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Conceptos fundamentales: sistemas fiables y disponibles. • Redundancia espacial y temporal. • Los métodos utilizados para implementar la tolerancia a fallos. • Los ejemplos de los mecanismos del sistema operativo para la detección, recuperación, reinicio para implementar la tolerancia a fallos, el uso de estas técnicas para los servicios propios del sistema operativo. 	<ul style="list-style-type: none"> • Explicar la importancia de los términos tolerancia a fallos, fiabilidad y disponibilidad [Familiarizarse] • Explicar en términos generales la gama de métodos para implementar la tolerancia a fallos en un sistema operativo [Familiarizarse] • Explicar cómo un sistema operativo puede continuar funcionando después de que ocurra una falla [Familiarizarse]
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unidad 12: Evaluación del desempeño de sistemas (3)	
Competencias esperadas: b	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • ¿Por qué el rendimiento del sistema debe ser evaluado? • ¿Qué se va a evaluar? • Sistemas de políticas de rendimiento, por ejemplo, el almacenamiento en caché, de paginación, la programación, la gestión de memoria, y la seguridad. • Modelos de evaluación: analítica, simulación, o de implementación específico determinista. • Cómo recoger los datos de evaluación (perfiles y mecanismos de localización) 	<ul style="list-style-type: none"> • Describir las medidas de rendimiento utilizados para determinar cómo el sistema funciona [Familiarizarse] • Explicar los principales modelos de evaluación utilizados para evaluar un sistema [Familiarizarse]
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

***** EVALUATION MISSING *****

10. BIBLIOGRAFÍA BÁSICA

- [AD14] Thomas Anderson and Michael Dahlin. *Operating Systems: Principles and Practice*. 2nd. Recursive Books, 2014. ISBN: 978-0985673529.
- [Avi12] Greg Gagne Avi Silberschatz Peter Baer Galvin. *Operating System Concepts, 9/E*. John Wiley & Sons, Inc., 2012. ISBN: 978-1-118-06333-0.
- [Sta05] William Stallings. *Operating Systems: Internals and Design Principles, 5/E*. Prentice Hall, 2005. ISBN: 0-13-147954-7.
- [Tan01] Andrew S. Tanenbaum. *Modern Operating Systems, 4/E*. Prentice Hall, 2001. ISBN: 0-13-031358-0.
- [Tan06] Andrew S. Tanenbaum. *Operating Systems Design and Implementation, 3/E*. Prentice Hall, 2006. ISBN: 0-13-142938-8.

Ministerio de Educación (MINEDU)
Sílabo 2021-I

1. CURSO

FG350. Liderazgo y Desempeño (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	: 2
2.2 Horas de teoría	: 2 (Semanal)
2.3 Horas de práctica	: -
2.4 Duración del periodo	: 16 semanas
2.5 Condición	: Obligatorio
2.6 Modalidad	: Presencial
2.7 Prerrequisitos	: FG203. Oratoria. (3 ^{er} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

En la actualidad las diferentes organizaciones en el mundo exigen a sus integrantes el ejercicio de liderazgo, esto significa asumir los retos asignados con eficacia y afán de servicio, siendo estas exigencias necesarias para la búsqueda de una sociedad más justa y reconciliada. Este desafío, pasa por la necesidad de formar a nuestros alumnos con un recto conocimiento de sí mismos, con capacidad de juzgar objetivamente la realidad y de proponer orientaciones que busquen modificar positivamente el entorno.

5. OBJETIVOS

- Desarrollar conocimientos, criterios, capacidades y actitudes para ejercer liderazgo, con el objeto de lograr la eficacia y servicio en los retos asignados, contribuyendo así en la construcción de una mejor sociedad.

6. COMPETENCIAS

- d) Trabajar efectivamente en equipos para cumplir con un objetivo común. (**Usar**)
- f) Comunicarse efectivamente con audiencias diversas. (**Usar**)
- ñ) Comprender que la formación de un buen profesional no se desliga ni se opone sino mas bien contribuye al auténtico crecimiento personal. Esto requiere de la asimilación de valores sólidos, horizontes espirituales amplios y una visión profunda del entorno cultural. (**Usar**)

■NoSpecificOutcomes■

7. TEMAS

Unidad 1: Primera Unidad: Fundamentos del liderazgo (15)	
Competencias esperadas: C18,C24	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Teorías de Liderazgo:• Definición de Liderazgo.• Fundamentos de Liderazgo.• Visión integral del Ser Humano y Motivos de la acción.• La práctica de la Virtud en el ejercicio de Liderazgo.	<ul style="list-style-type: none">• Analizar y comprender las bases teóricas del ejercicio de Liderazgo.[Familiarizarse]• En base a lo comprendido, asumir la actitud correcta para llevarlo a la práctica.[Familiarizarse]• Iniciar un proceso de autoconocimiento orientado a descubrir rasgos de liderazgo en sí mismo.[Familiarizarse]
Lecturas : [Pil02], [Man09], [Ale09], [D S], [Alf10]	

Unidad 2: (15)	
Competencias esperadas: C17,C18,C24	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Teoría de las Competencias • Reconocimiento de Competencias • Plan de Desarrollo • Modelos Mentales • Necesidades Emocionales • Perfiles Emocionales • Vicios Motivacionales 	<ul style="list-style-type: none"> • Conocer y Desarrollar competencias de Liderazgo, centradas en lograr la eficacia, sin dejar de lado el deber de servicio con los demás.[Familiarizarse] • Reconocer las tendencias personales y grupales necesarias para el ejercicio de Liderazgo.[Familiarizarse]
Lecturas : [Wil09], [Lui08], [Pil02], [Mar07]	

Unidad 3: (18)	
Competencias esperadas: C18,C24	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • La relación personal con el equipo • Liderazgo integral • Acompañamiento y discipulado • Fundamentos de unidad 	<ul style="list-style-type: none"> • Desarrollar habilidades para el trabajo en equipo[Familiarizarse]
Lecturas : [Gol12], [CardonaP], [Hersey], [Hun10], [Haw12], [Ginebra]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

***** EVALUATION MISSING *****

10. BIBLIOGRAFÍA BÁSICA

- [Ale09] Dianine-Havard Alexandre. *Perfil del Líder. Hacia un Liderazgo Virtuoso*. Ediciones Urano S.A, 2009.
- [Alf10] Sonnenfeld Alfred. *Liderazgo Ético. La Sabiduría de decidir bien*. Ediciones Encuentro S.A Madrid y Nueva Revista de Madrid, 2010.
- [D S] SJ Anthony. D' Souza. *Descubre tu Liderazgo*. Editorial Sal Terrae.

- [Gol12] D. Goleman. *Inteligencia emocional*. Editorial Kairós., 2012.
- [Haw12] Peter. Hawkins. *Coaching y liderazgo de equipos: coaching para un liderazgo con capacidad de transformación*. Ediciones Granica, 2012.
- [Hun10] Phil. Hunsaker. *El nuevo arte de gestionar equipos: Un enfoque actual para guiar y motivar con éxito*. 2010.
- [Lui08] Huete Luis. *Construye tu Sueño*. LID Editorial Empresarial, 2008.
- [Man09] Ferreiro Pablo/Alcázar Manuel. *Gobierno de Personas en la Empresa*. Ediciones Universidad de Navarra EUNSA, 2009.
- [Mar07] Chinchilla Nuria/Moragas Maruja. *Dueños de Nuestro Destino*. Editorial Ariel, 2007.
- [Pil02] Cardona Pablo/García Lombardi Pilar. *Cómo desarrollar las Competencias de Liderazgo*. PAD Lima- Perú, Tercera Edición., 2002.
- [Wil09] Cardona Pablo/ Helen Wilkinson. *Creciendo como Líder*. Ediciones Universidad de Navarra S.A (EUNSA), Primera Edición, 2009.

Ministerio de Educación (MINEDU)
Sílabo 2021-I

1. CURSO

ID202. Inglés IV (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	: 3
2.2 Horas de teoría	: -
2.3 Horas de práctica	: -
2.4 Duración del periodo	: 16 semanas
2.5 Condición	: Obligatorio
2.6 Modalidad	: Presencial
2.7 Prerrequisitos	: ID201. Inglés III. (3 ^{er} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Parte fundamental de la formación integral de un profesional es la habilidad de comunicarse en un idioma extranjero además del propio idioma nativo. No solamente amplía su horizonte cultural sino que permite una visión más humana y comprensiva de la vida. En el caso de los idiomas extranjeros, indudablemente el Inglés es el más práctico porque es hablado alrededor de todo el mundo. No hay país alguno donde este no sea hablado. En las carreras relacionadas con los servicios al turista el inglés es tal vez la herramienta práctica más importante que el alumno debe dominar desde el primer momento como parte de su formación integral.

5. OBJETIVOS

- Incrementar el nivel de conversación en diferentes temas, en los alumnos. Así como la capacidad de escribir y leer documentación de todo tipo.
- Llevar al alumno a una expresión más intensa en el dominio del idioma.

6. COMPETENCIAS

f) Comunicarse efectivamente con audiencias diversas. (**Usar**)

■NoSpecificOutcomes■

7. TEMAS

Unidad 1: Do and don't! (0)	
Competencias esperadas: C25	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Auxiliares de Modo should, must y have got to.• Oraciones afirmativas, negativas e interrogativas con modals.• Términos para cartas formales.• Partes de las respuestas cortas.• Expresiones para ocupaciones.	<ul style="list-style-type: none">• Al terminar la octava unidad, cada una de las alumnas, comprendiendo la gramática de los auxiliares should y must es capaz de expresar una mayor cantidad de acciones en forma obligación y sugerencia. Además es capaz de expresar ideas describiendo ocupaciones. Asume la necesidad de escribir cartas formales.
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 2: Going places! (0)	
Competencias esperadas: C25	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Tiempo Presente Simple y Futuro con Will • Primer Condicional • Colocaciones • Vocabulario de preposiciones de lugar y de tiempo • Expresiones de conexión de ideas 	<ul style="list-style-type: none"> • Al terminar la novena unidad, los alumnos habiendo identificado la forma de expresar presente reconocen la diferencia entre las formas de futuro y las aplican adecuadamente. Describen condiciones acuciosamente. Asumen expresiones para demostrar ubicación de lugar. Utilizan expresiones de tiempo y conectores para unir ideas varias.
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 3: Scared to death! (0)	
Competencias esperadas: C25	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Patrones de Verbos Infinitivos y gerundios • What + Infinitivo • Something + infinitive • Expresiones de sentimientos • Exclamaciones de sorpresa 	<ul style="list-style-type: none"> • Al terminar la décimo unidad los alumnos serán capaces de reconocer y utilizar patrones tiempos en pasado los utilizan adecuadamente. Utilizan expresiones de exclamación. Y describen sentimientos. Utilizarán conjunciones para unir ideas tipo.
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 4: Things that changed the world! (0)	
Competencias esperadas: C25	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Voz Pasiva • Oraciones Afirmativas, Negativas y Preguntas • Uso de Participios, verbos y sustantivos que van unidos • Señales. Signos y notas • Resúmenes • Expresiones para indicar prohibición 	<ul style="list-style-type: none"> • Al terminar la décimo primera primera unidad, los alumnos habiendo identificado la idea de acciones pasivas describen acciones adecuadamente en diversas situaciones que la involucran. Reconocen y aplican participios. Asumen la idea de respetar signos y señales públicas. Expresan ideas de hábitos . Hacen resúmenes.
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 5: Dreams and reality! (0)	
Competencias esperadas: C25	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Segundo Condicional • Auxiliar de Modo Might • Verbos de Frase • Vocabulario de expresiones sociales • Adverbios • Expresiones para dar consejo 	<ul style="list-style-type: none"> • Al finalizar la décimo segunda unidad, los alumnos, a partir de la comprensión de la idea de Condicionales y de expresar posibilidad elaborarán oraciones utilizando los elementos necesarios. Asimilarán además la necesidad de frases verbales (verbos de 2 palabras). Adquirirán vocabulario para describir expresiones sociales.
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 6: Making a living! (0)	
Competencias esperadas: C25	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Present Perfect Continuous • Present Continuous • Ocupaciones • Formación de palabras • Adverbios • Expresiones de uso en el teléfono. 	<ul style="list-style-type: none"> • Al finalizar la décimo tercera unidad estructuran oraciones con acciones que incluyen presente y pasado en contextos adecuados. Enfatizan la diferencia entre tipos de ocupaciones. Utilizan expresiones adecuadas para conversaciones telefónicas
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 7: All you need is love! (0)	
Competencias esperadas: C25	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Pasado Perfecto y Pasado Simple • Expresiones de Reporte • Expresiones de palabras en contextos diferentes • Despedidas cortas y formales • Historias de amor 	<ul style="list-style-type: none"> • Al finalizar la décimo cuarta unidad, los alumnos habiendo conocido los fundamentos de la estructuración del tiempo pasado perfecto, lo diferencian del pasado simple. Enfatizan la diferencia entre palabras en contextos diferentes. Describen ideas de despedidas. Utilizan expresiones para escribir historias de amor. Asumen la idea de dar y hacer entrevistas.
Lecturas : [SJ02], [Cam06], [Mac99]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

***** EVALUATION MISSING *****

10. BIBLIOGRAFÍA BÁSICA

- [Cam06] Cambridge. *Diccionario Inglés-Español Cambridge*. Editorial Oxford, 2006.
[Mac99] James MacGrew. *Focus on Grammar Basic*. Editorial Oxford, 1999.
[SJ02] Liz Soars and John. *American Headway N 2 Student Book*. Editorial Oxford, 2002.

Ministerio de Educación (MINEDU)
Sílabo 2021-I

1. CURSO

CS212. Análisis y Diseño de Algoritmos (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	4
2.2 Horas de teoría	:	2 (Semanal)
2.3 Horas de práctica	:	4 (Semanal)
2.4 Duración del periodo	:	16 semanas
2.5 Condición	:	Obligatorio
2.6 Modalidad	:	Presencial
2.7 Prerrequisitos	:	<ul style="list-style-type: none">• CS210. Algoritmos y Estructuras de Datos. (4^{to} Sem)• MA101. Matemática II. (2^{do} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Un algoritmo es, esencialmente, un conjunto bien definido de reglas o instrucciones que permitan resolver un problema computacional. El estudio teórico del desempeño de los algoritmos y los recursos utilizados por estos, generalmente tiempo y espacio, nos permite evaluar si un algoritmo es adecuado para un resolver un problema específico, compararlo con otros algoritmos para el mismo problema o incluso delimitar la frontera entre lo viable y lo imposible.

Esta materia es tan importante que incluso Donald E. Knuth definió a Ciencia de la Computación como el estudio de algoritmos.

En este curso serán presentadas las técnicas más comunes utilizadas en el análisis y diseño de algoritmos eficientes, con el propósito de aprender los principios fundamentales del diseño, implementación y análisis de algoritmos para la solución de problemas computacionales.

5. OBJETIVOS

- Desarrollar la capacidad para evaluar la complejidad y calidad de algoritmos propuestos para un determinado problema.
- Estudiar los algoritmos más representativos, introductorios de las clases más importantes de problemas tratados en computación.
- Desarrollar la capacidad de resolución de problemas algorítmicos utilizando los principios fundamentales de diseño de algoritmos aprendidos.
- Ser capaz de responder a las siguientes preguntas cuando le sea presentado un nuevo algoritmo: ¿Cuán buen desempeño tiene?, ¿Existe una mejor forma de resolver el problema?

6. COMPETENCIAS

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Evaluar**)
- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Evaluar**)
- a8) Hacer un análisis computacional que permita calcular el tiempo de ejecución de un determinado algoritmo. (**8**)
- a9) Utilizar técnicas matemáticas que permitan acotar sumatorias y resolver recurrencias que reflejan los costos computacionales de un algoritmo. (**9**)
- b1) Identificar y aplicar de forma eficiente diversas estrategias algorítmicas y estructuras de datos para la solución de un problema dadas ciertas restricciones de espacio y tiempo. (**1**)
- b3) Entender la diferencia entre un problema NP-difícil y uno que tiene solución polinomial. (**3**)

- b4) Dado un problema con solución polinomial, identificar si es posible resolverlo mediante una estrategia voraz, mediante una estrategia de programación dinámica o una de división y conquista tomando en cuenta el tamaño de la entrada..
(4)

7. TEMAS

Unidad 1: Análisis Básico (10)	
Competencias esperadas: C1	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Diferencias entre el mejor, el esperado y el peor caso de un algoritmo. • Análisis asintótico de complejidad de cotas superior y esperada. • Definición formal de la Notación Big O. • Clases de complejidad como constante, logarítmica, lineal, cuadrática y exponencial. • Uso de la notación Big O. • Relaciones recurrentes. • Análisis de algoritmos iterativos y recursivos. • Teorema Maestro y Árboles Recursivos. 	<ul style="list-style-type: none"> • Explique a que se refiere con “mejor”, “esperado” y “peor” caso de comportamiento de un algoritmo [Evaluar] • En el contexto de a algoritmos específicos, identifique las características de data y/o otras condiciones o suposiciones que lleven a diferentes comportamientos [Evaluar] • Determine informalmente el tiempo y el espacio de complejidad de diferentes algoritmos [Evaluar] • Indique la definición formal de Big O [Evaluar] • Lista y contraste de clases estándares de complejidad [Evaluar] • Use la notación formal de la Big O para dar límites superiores asintóticos en la complejidad de tiempo y espacio de los algoritmos [Evaluar] • Usar la notación formal Big O para dar límites de casos esperados en el tiempo de complejidad de los algoritmos [Evaluar] • Explicar el uso de la notación theta grande, omega grande y o pequeña para describir la cantidad de trabajo hecho por un algoritmo [Evaluar] • Usar relaciones recurrentes para determinar el tiempo de complejidad de algoritmos recursivamente definidos [Evaluar] • Resuelve relaciones de recurrencia básicas, por ejemplo. usando alguna forma del Teorema Maestro [Evaluar]
Lecturas : [KT05], [DPV06], [RS09], [SF13], [Knu97]	

Unidad 2: Estrategias Algorítmicas (30)**Competencias esperadas: C2**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Algoritmos de fuerza bruta.• Algoritmos voraces.• Divide y vencerás.• Programación Dinámica.	<ul style="list-style-type: none">• Para cada una de las estrategias (fuerza bruta, algoritmo goloso, divide y vencerás, recursividad en reversa y programación dinámica), identifica un ejemplo práctico en el cual se pueda aplicar [Evaluar]• Utiliza un enfoque voraz para resolver un problema específico y determina si la regla escogida lo guía a una solución óptima [Evaluar]• Usa un algoritmo de divide-y-vencerás para resolver un determinado problema [Evaluar]• Usa programación dinámica para resolver un problema determinado [Evaluar]• Determina el enfoque algorítmico adecuado para un problema [Evaluar]

Lecturas : [KT05], [DPV06], [RS09], [Als99]

Unidad 3: Algoritmos y Estructuras de Datos fundamentales (10)	
Competencias esperadas: C6	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Algoritmos numéricos simples, tales como el cálculo de la media de una lista de números, encontrar el mínimo y máximo. • Algoritmos de búsqueda secuencial y binaria. • Algoritmos de ordenamiento de peor caso cuadrático (selección, inserción) • Algoritmos de ordenamiento con peor caso o caso promedio en $O(N \lg N)$ (Quicksort, Heapsort, Mergesort) • Grafos y algoritmos en grafos: <ul style="list-style-type: none"> – Representación de grafos (ej., lista de adyacencia, matriz de adyacencia) – Recorrido en profundidad y amplitud • Montículos (Heaps) • Grafos y algoritmos en grafos: <ul style="list-style-type: none"> – Problema de corte máximo y mínimo – Búsqueda local 	<ul style="list-style-type: none"> • Implementar algoritmos numéricos básicos [Evaluar] • Implementar algoritmos de búsqueda simple y explicar las diferencias en sus tiempos de complejidad [Evaluar] • Ser capaz de implementar algoritmos de ordenamiento comunes cuadráticos y $O(N \log N)$ [Evaluar] • Discutir el tiempo de ejecución y eficiencia de memoria de los principales algoritmos de ordenamiento, búsqueda y hashing [Usar] • Discutir factores otros que no sean eficiencia computacional que influyan en la elección de algoritmos, tales como tiempo de programación, mantenibilidad, y el uso de patrones específicos de la aplicación en los datos de entrada [Familiarizarse] • Resolver problemas usando algoritmos básicos de grafos, incluyendo búsqueda por profundidad y búsqueda por amplitud [Evaluar] • Demostrar habilidad para evaluar algoritmos, para seleccionar de un rango de posibles opciones, para proveer una justificación por esa selección, y para implementar el algoritmo en un contexto en específico [Evaluar] • Describir la propiedad del heap y el uso de heaps como una implementación de colas de prioridad [Evaluar] • Resolver problemas usando algoritmos de grafos, incluyendo camino más corto de una sola fuente y camino más corto de todos los pares, y como mínimo un algoritmo de árbol de expansión mínima [Evaluar]
Lecturas : [KT05], [DPV06], [RS09], [SW11], [GT09]	

Unidad 4: Computabilidad y complejidad básica de autómatas (2)	
Competencias esperadas: C9	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Introducción a las clases P y NP y al problema P vs. NP. • Introducción y ejemplos de problemas NP- Completos y a clases NP-Completos. 	<ul style="list-style-type: none"> • Defina las clases P y NP [Familiarizarse] • Explique el significado de NP-Complejidad [Familiarizarse]
Lecturas : [KT05], [DPV06], [RS09]	

Unidad 5: Estructuras de Datos Avanzadas y Análisis de Algoritmos (8)	
Competencias esperadas: C16	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Grafos (ej. Ordenamiento Topológico, encontrando componentes fuertemente conectados) • Algoritmos Teórico-Numéricos (Aritmética Modular, Prueba del Número Primo, Factorización Entera) • Algoritmos aleatorios. • Análisis amortizado. • Análisis Probabilístico. 	<ul style="list-style-type: none"> • Entender el mapeamiento de problemas del mundo real a soluciones algorítmicas (ejemplo, problemas de grafos, programas lineales, etc) [Familiarizarse] • Seleccionar y aplicar técnicas de algoritmos avanzadas (ejemplo, randomización, aproximación) para resolver problemas reales [Usar] • Seleccionar y aplicar técnicas avanzadas de análisis (ejemplo, amortizado, probabilístico, etc) para algoritmos [Usar]
Lecturas : [KT05], [DPV06], [RS09], [Tar83], [Raw92]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

***** EVALUATION MISSING *****

10. BIBLIOGRAFÍA BÁSICA

- [Als99] H. Alsuwaiyel. *Algorithms: Design Techniques and Analysis*. World Scientific, 1999. ISBN: 9789810237400.
- [DPV06] S. Dasgupta, C. Papadimitriou, and U. Vazirani. *Algorithms*. McGraw-Hill Education, 2006. ISBN: 9780073523408.
- [GT09] Michael T. Goodrich and Roberto Tamassia. *Algorithm Design: Foundations, Analysis and Internet Examples*. 2nd. John Wiley & Sons, Inc., 2009. ISBN: 0470088540, 9780470088548.
- [Knu97] D.E. Knuth. *The Art of Computer Programming: Fundamental algorithms Vol 1*. Third Edition. Addison-Wesley, 1997. ISBN: 9780201896831. URL: <http://www-cs-faculty.stanford/~knuth/taocp.html>.
- [KT05] Jon Kleinberg and Eva Tardos. *Algorithm Design*. Addison-Wesley Longman Publishing Co., Inc., 2005. ISBN: 0321295358.
- [Raw92] G.J.E. Rawlins. *Compared to What?: An Introduction to the Analysis of Algorithms*. Computer Science Press, 1992. ISBN: 9780716782438.
- [RS09] Thomas H. Cormen; Charles E. Leiserson ; Ronald L. Rivest and Clifford Stein. *Introduction to Algorithms, Third Edition*. 3rd. The MIT Press, 2009. ISBN: 0262033844.
- [SF13] R. Sedgewick and P. Flajolet. *An Introduction to the Analysis of Algorithms*. Pearson Education, 2013. ISBN: 9780133373486.
- [SW11] R. Sedgewick and K. Wayne. *Algorithms*. Pearson Education, 2011. ISBN: 9780132762564.
- [Tar83] Robert Endre Tarjan. *Data Structures and Network Algorithms*. Society for Industrial and Applied Mathematics, 1983. ISBN: 0-89871-187-8.

Ministerio de Educación (MINEDU)
Sílabo 2021-I

1. CURSO

CS231. Redes y Comunicaciones (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	3
2.2 Horas de teoría	:	1 (Semanal)
2.3 Horas de práctica	:	4 (Semanal)
2.4 Duración del periodo	:	16 semanas
2.5 Condición	:	Obligatorio
2.6 Modalidad	:	Presencial
2.7 Prerrequisitos	:	CS251. Sistemas Operativos. (4 ^{to} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

El siempre creciente desarrollo de las tecnologías de comunicación y la información hace que exista una marcada tendencia a establecer más redes de computadores que permitan una mejor gestión de la información.

En este segundo curso se brindará a los participantes una introducción a los problemas que conlleva la comunicación entre computadores, a través del estudio e implementación de protocolos de comunicación como TCP/IP y la implementación de software sobre estos protocolos.

5. OBJETIVOS

- Que el alumno implemente y/o modifique un protocolo de comunicación de datos.
- Que el alumno domine las técnicas de transmisión de datos utilizadas por los protocolos de red existentes.
- Que el alumno conozca las últimas tendencias en redes que se están aplicando en el Internet.

6. COMPETENCIAS

- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Familiarizarse**)
- c) Diseñar, implementar y evaluar un sistema, proceso, componente o programa computacional para alcanzar las necesidades deseadas. (**Usar**)
- i) Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (**Usar**)

■NoSpecificOutcomes■

7. TEMAS

Unidad 1: Introducción (5)	
Competencias esperadas: b,c	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Organización de la Internet (proveedores de servicios de Internet, proveedores de contenido, etc) • Técnicas de Switching (por ejemplo, de circuitos, de paquetes) • Piezas físicas de una red, incluidos hosts, routers, switches, ISPs, inalámbrico, LAN, punto de acceso y firewalls. • Principios de capas (encapsulación, multiplexación) • Roles de las diferentes capas (aplicación, transporte, red, enlace de datos, física) 	<ul style="list-style-type: none"> • Articular la organización de la Internet [Familiarizarse] • Listar y definir la terminología de red apropiada [Familiarizarse] • Describir la estructura en capas de una arquitectura típica en red [Familiarizarse] • Identificar los diferentes tipos de complejidad en una red (bordes, núcleo, etc.) [Familiarizarse]
Lecturas : [KR13]	

Unidad 2: Aplicaciones en red (5)	
Competencias esperadas: b,c,i	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Esquemas de denominación y dirección (DNS, direcciones IP, identificadores de recursos uniformes, etc) • Las aplicaciones distribuidas (cliente / servidor, peer-to-peer, nube, etc) • HTTP como protocolo de capa de aplicación . • Multiplexación con TCP y UDP • API de Socket 	<ul style="list-style-type: none"> • Listar las diferencias y las relaciones entre los nombres y direcciones en una red [Familiarizarse] • Definir los principios detrás de esquemas de denominación y ubicación del recurso [Familiarizarse] • Implementar una aplicación simple cliente-servidor basada en <i>sockets</i> [Usar]
Lecturas : [KR13]	

Unidad 3: Entrega confiable de datos (10)	
Competencias esperadas: C6,b,c,i	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Control de errores (técnicas de retransmisión, temporizadores) • El control de flujo (agradecimientos, ventana deslizante) • Problemas de rendimiento (pipelining) • TCP 	<ul style="list-style-type: none"> • Describir el funcionamiento de los protocolos de entrega fiables [Familiarizarse] • Listar los factores que afectan al rendimiento de los protocolos de entrega fiables [Familiarizarse] • Diseñar e implementar un protocolo confiable simple [Usar]
Lecturas : [KR13]	

Unidad 4: Ruteo y reenvío (12)	
Competencias esperadas: b,c,i	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Enrutamiento vs reenvío . • Enrutamiento estático . • Protocolo de Internet (IP) • Problemas de escalabilidad (direccionamiento jerárquico) 	<ul style="list-style-type: none"> • Describir la organización de la capa de red [Familiarizarse] • Describir cómo los paquetes se envían en una red IP [Familiarizarse] • Listar las ventajas de escalabilidad de direccionamiento jerárquico [Familiarizarse]
Lecturas : [KR13]	

Unidad 5: Redes de área local (10)	
Competencias esperadas: b,c	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Problemas de Acceso Múltiple. • Enfoques comunes a Acceso múltiple (exponencial backoff, multiplexación por división de tiempo, etc) • Redes de área local . • Ethernet . • Switching . 	<ul style="list-style-type: none"> • Describir como los paquetes son enviados en una red Ethernet [Familiarizarse] • Describir las relaciones entre IP y Ethernet [Familiarizarse] • Describir las etapas usadas en un enfoque común para el problema de múltiples accesos [Familiarizarse]
Lecturas : [KR13]	

Unidad 6: Asignación de recursos (12)	
Competencias esperadas: b,c,i	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Necesidad de asignación de recursos . • Asignación fija (TDM, FDM, WDM) versus la asignación dinámica . • De extremo a extremo frente a las red de enfoque asistida . • Justicia. • Principios del control de congestión. • Enfoques para la congestión (por ejemplo, redes de distribución de contenidos) 	<ul style="list-style-type: none"> • Describir como los recursos pueden ser almacenados en la red [Familiarizarse] • Describir los problemas de congestión en una red grande [Familiarizarse] • Comparar y contrastar las técnicas de almacenamiento estático y dinámico [Familiarizarse] • Comparar y contrastar los enfoques actuales de la congestión [Familiarizarse]
Lecturas : [KR13]	

Unidad 7: Celulares (5)	
Competencias esperadas: b,c	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Principios de redes celulares. • Redes 802.11 • Problemas en el apoyo a los nodos móviles (agente local) 	<ul style="list-style-type: none"> • Describir la organización de una red inalámbrica [Familiarizarse] • Describir como las redes inalámbricas soportan usuarios móviles [Familiarizarse]
Lecturas : [KR13], [Cha16]	

Unidad 8: Redes sociales (5)	
Competencias esperadas: b,c,i	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Panorama de las redes sociales. • Ejemplo plataformas de redes sociales. • Estructura de los grafos de redes sociales. • Análisis de redes sociales. 	<ul style="list-style-type: none"> • Discutir los principios fundamentales (como pertenencia, confianza) de una red social [Familiarizarse] • Describir como redes sociales existentes operan [Familiarizarse] • Construir un grafo de una red social a partir de datos de la red [Usar] • Analizar una red social para determinar quienes son las personas importantes [Usar] • Evaluar una determinada interpretación de una pregunta de red social con los datos asociados [Familiarizarse]
Lecturas : [KR13], [Kad11]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

***** EVALUATION MISSING *****

10. BIBLIOGRAFÍA BÁSICA

[Cha16] Paresh Chayapathi Rajendra; Syed F. Hassan; Shah. *Network Functions Virtualization (NFV) with a Touch of SDN*. Addison-Wesley Professional; 1 edition, 2016. ISBN: 978-0134463056.

- [Kad11] Charles Kadushin. *Understanding Social Networks: Theories, Concepts, And Findings*. Oxford University Press, Usa; 1 edition, 2011. ISBN: 978-0195379471.
- [KR13] J.F. Kurose and K.W. Ross. *Computer Networking: A Top-down Approach*. 7th. Always learning. Pearson, 2013. ISBN: 978-0133594140.

Ministerio de Educación (MINEDU)
Sílabo 2021-I

1. CURSO

CS261. Sistemas Inteligentes (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	4
2.2 Horas de teoría	:	2 (Semanal)
2.3 Horas de práctica	:	2 (Semanal)
2.4 Duración del periodo	:	16 semanas
2.5 Condición	:	Obligatorio
2.6 Modalidad	:	Presencial
2.7 Prerrequisitos	:	MA203. Estadística y Probabilidades. (3 ^{er} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

La investigación en Inteligencia Artificial ha conducido al desarrollo de numerosas técnicas relevantes, dirigidas a la automatización de la inteligencia humana, dando una visión panorámica de diferentes algoritmos que simulan los diferentes aspectos del comportamiento y la inteligencia del ser humano.

5. OBJETIVOS

- Evaluar las posibilidades de simulación de la inteligencia, para lo cual se estudiarán las técnicas de modelización del conocimiento.
- Construir una noción de inteligencia que soporte después las tareas de su simulación.

6. COMPETENCIAS

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Usar**)
- j) Aplicar la base matemática, principios de algoritmos y la teoría de la CS en el modelamiento y diseño de sistemas. (**Familiarizarse**)

■NoSpecificOutcomes■

7. TEMAS

Unidad 1: Cuestiones fundamentales (2)**Competencias esperadas: a**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Descripción general de los problemas de Inteligencia Artificial, ejemplos recientes de aplicaciones de Inteligencia artificial.• ¿Qué es comportamiento inteligente?<ul style="list-style-type: none">– El Test de Turing– Razonamiento Racional versus No Racional• Características del Problema:<ul style="list-style-type: none">– Observable completamente versus observable parcialmente– Individual versus multi-agente– Determinístico versus estocástico– Estático versus dinámico– Discreto versus continuo• Naturaleza de agentes:<ul style="list-style-type: none">– Autónomo versus semi-autónomo– Reflexivo, basado en objetivos, y basado en utilidad– La importancia en percepción e interacciones con el entorno• Cuestiones filosóficas y éticas.	<ul style="list-style-type: none">• Describir el test de Turing y el experimento pensado "cuarto chino" (<i>Chinese Room</i>) [Usar]• Determinando las características de un problema dado que sistemas inteligentes deberían resolver [Usar]

Lecturas : [De 06], [Pon+14]

Unidad 2: Agentes (2)	
Competencias esperadas: a	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Definición de Agentes • Arquitectura de agentes (Ej. reactivo, en capa, cognitivo) • Teoría de agentes • Racionalidad, teoría de juegos: <ul style="list-style-type: none"> – Agentes de decisión teórica – Procesos de decisión de Markov (MDP) • Agentes de Software, asistentes personales, y acceso a información: <ul style="list-style-type: none"> – Agentes colaborativos – Agentes de recolección de información – Agentes creíbles (carácter sintético, modelamiento de emociones en agentes) • Agentes de aprendizaje • Sistemas Multi-agente <ul style="list-style-type: none"> – Agentes Colaborativos – Equipos de Agentes – Agentes Competitivos (ej., subastas, votaciones) – Sistemas de enjambre y modelos biológicamente inspirados 	<ul style="list-style-type: none"> • Lista las características que definen un agente inteligente [Usar] • Describe y contrasta las arquitecturas de agente estándares [Usar] • Describe las aplicaciones de teoría de agentes para dominios como agentes de software, asistentes personales, y agentes creíbles [Usar] • Describe los paradigmas primarios usados por agentes de aprendizaje [Usar] • Demuestra mediante ejemplos adecuados como los sistemas multi-agente soportan interacción entre agentes [Usar]
Lecturas : [Nil01], [RN03], [Pon+14]	

Unidad 3: Estrategias de búsquedas básicas (2)	
Competencias esperadas: a,j	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Espacios de Problemas (estados, metas y operadores), solución de problemas mediante búsqueda. • Factored representation (factoring state hacia variables) • Uninformed search (breadth-first, depth-first, depth-first with iterative deepening) • Heurísticas y búsqueda informada (hill-climbing, generic best-first, A*) • El espacio y el tiempo de la eficiencia de búsqueda. • Dos jugadores juegos (introducción a la búsqueda minimax). • Satisfacción de restricciones (backtracking y métodos de búsqueda local). 	<ul style="list-style-type: none"> • Formula el espacio eficiente de un problema para un caso expresado en lenguaje natural (ejm. Inglés) en términos de estados de inicio y final, así como sus operadores [Usar] • Describe el rol de las heurísticas y describe los intercambios entre completitud, óptimo, complejidad de tiempo, y complejidad de espacio [Usar] • Describe el problema de la explosión combinatoria del espacio de búsqueda y sus consecuencias [Usar] • Compara y contrasta tópicos de búsqueda básica con temas jugabilidad de juegos [Usar]
Lecturas : [Nil01], [Pon+14]	

Unidad 4: Búsqueda Avanzada (18)	
Competencias esperadas: a,j	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Búsqueda estocástica: <ul style="list-style-type: none"> – Simulated annealing – Algoritmos genéticos – Búsqueda de árbol Monte-Carlo • Construcción de árboles de búsqueda, espacio de búsqueda dinámico, explosión combinatoria del espacio de búsqueda. • Implementación de búsqueda A*, búsqueda en haz. • Búsqueda Minimax, poda alfa-beta. • Búsqueda Expectimax (MDP-Solving) y los nodos de azar. 	<ul style="list-style-type: none"> • Diseñar e implementar una solución a un problema con algoritmo genético [Usar] • Diseñar e implementar un esquema de recocido simulado (<i>simulated annealing</i>) para evitar mínimos locales en un problema [Usar] • Diseñar e implementar una búsqueda A* y búsqueda en haz (<i>beam search</i>) para solucionar un problema [Usar] • Aplicar búsqueda minimax con poda alfa-beta para simplificar el espacio de búsqueda en un juego con dos jugadores [Usar] • Comparar y contrastar los algoritmos genéticos con técnicas clásicas de búsqueda [Usar] • Comparar y contrastar la aplicabilidad de varias heurísticas de búsqueda, para un determinado problema [Usar]
Lecturas : [Gol89], [Nil01], [RN03], [Pon+14]	

Unidad 5: Razonamiento Bajo Incertidumbre (18)	
Competencias esperadas: a,j	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Revisión de Probabilidad Básica • Variables aleatorias y distribuciones de probabilidad: <ul style="list-style-type: none"> – Axiomas de probabilidad – Inferencia probabilística – Regla de Bayes • Independencia Condicional • Representaciones del conocimiento: <ul style="list-style-type: none"> – Redes bayesianas <ul style="list-style-type: none"> * Inferencia exacta y su complejidad * Métodos de Muestreo aleatorio (Monte Carlo) (p.e. Muestreo de Gibbs) – Redes Markov – Modelos de probabilidad relacional – Modelos ocultos de Markov 	<ul style="list-style-type: none"> • Aplicar la regla de Bayes para determinar el cumplimiento de una hipótesis [Usar] • Explicar cómo al tener independencia condicional permite una gran eficiencia en sistemas probabilísticos [Usar] • Identificar ejemplos de representación de conocimiento para razonamiento bajo incertidumbre [Usar] • Indicar la complejidad de la inferencia exacta. Identificar métodos para inferencia aproximada [Usar]
Lecturas : [KF09], [RN03]	

Unidad 6: Aprendizaje Automático Básico (4)	
Competencias esperadas: a,j	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Definición y ejemplos de la extensa variedad de tareas de aprendizaje de máquina, incluida la clasificación. • Aprendizaje inductivo • Aprendizaje simple basado en estadísticas, como el clasificador ingenuo de Bayes, árboles de decisión. • El problema exceso de ajuste. • Medición clasificada con exactitud. 	<ul style="list-style-type: none"> • Listar las diferencias entre los tres principales tipos de aprendizaje: supervisado, no supervisado y por refuerzo [Usar] • Identificar ejemplos de tareas de clasificación, considerando las características de entrada disponibles y las salidas a ser predecidas [Usar] • Explicar la diferencia entre aprendizaje inductivo y deductivo [Usar] • Describir el sobre ajuste (<i>overfitting</i>) en el contexto de un problema [Usar] • Aplicar un algoritmo de aprendizaje estadístico simple como el Clasificador Naive Bayesiano e un problema de clasificación y medirla precisión del clasificador [Usar]
Lecturas : [Mit98], [RN03], [Pon+14]	

Unidad 7: Aprendizaje de máquina avanzado (20)	
Competencias esperadas: a,j	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Definición y ejemplos de una amplia variedad de tareas de aprendizaje de máquina • Aprendizaje general basado en estadística, estimación de parámetros (máxima probabilidad) • Programación lógica inductiva (<i>Inductive logic programming ILP</i>) • Aprendizaje supervisado <ul style="list-style-type: none"> – Aprendizaje basado en árboles de decisión – Aprendizaje basado en redes neuronales – Aprendizaje basado en máquinas de soporte vectorial (<i>Support vector machines SVMs</i>) • Aprendizaje y <i>clustering</i> no supervisado <ul style="list-style-type: none"> – EM – K-means – Mapas auto-organizados • Aprendizaje semi-supervisado. • Aprendizaje de modelos gráficos • Evaluación del desempeño (tal como cross-validation, area bajo la curva ROC) • Aplicación de algoritmos Machine Learning para Minería de datos. 	<ul style="list-style-type: none"> • Explica las diferencias entre los tres estilos de aprendizaje: supervisado, por refuerzo y no supervisado [Usar] • Implementa algoritmos simples para el aprendizaje supervisado, aprendizaje por refuerzo, y aprendizaje no supervisado [Usar] • Determina cuál de los tres estilos de aprendizaje es el apropiado para el dominio de un problema en particular [Usar] • Compara y contrasta cada una de las siguientes técnicas, dando ejemplo de cuando una estrategia es la mejor: árboles de decisión, redes neuronales, y redes bayesianas [Usar] • Evalúa el rendimiento de un sistema de aprendizaje simple en un conjunto de datos reales [Usar] • Describe el estado del arte en la teoría del aprendizaje, incluyendo sus logros y limitantes [Usar] • Explica el problema del sobreajuste, conjuntamente con técnicas para determinar y manejar el problema [Usar]
Lecturas : [RN03], [KF09], [Mur12]	

Unidad 8: Procesamiento del Lenguaje Natural (12)**Competencias esperadas: a,j**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Gramaticas determinísticas y estocásticas• Algoritmos de parseo<ul style="list-style-type: none">– Gramáticas libres de contexto (CFGs) y cuadros de parseo (e.g. Cocke-Younger-Kasami CYK)– CFGs probabilísticos y ponderados CYK• Representación del significado / Semántica<ul style="list-style-type: none">– Representación de conocimiento basado en lógica– Roles semánticos– Representaciones temporales– Creencias, deseos e intenciones• Metodos basados en el corpus• N-gramas y Modelos ocultos de Markov (HMMs)• Suavizado y back-off• Ejemplos de uso: POS etiquetado y morfología• Recuperación de la información:<ul style="list-style-type: none">– Modelo de espacio vectorial<ul style="list-style-type: none">* TF & IDF– Precision y cobertura• Extracción de información• Traducción de lenguaje• Clasificación y categorización de texto:<ul style="list-style-type: none">– Modelo de bolsa de palabras	<ul style="list-style-type: none">• Define y contrasta gramáticas de tipo estocásticas y determinísticas, dando ejemplos y demostrando como adecuar cada una de ellas [Usar]• Simula, aplica, o implementa algoritmos clásicos y estocásticos para el parseo de un lenguaje natural [Usar]• Identifica los retos de la representación del significado [Usar]• Lista las ventajas de usar corpus estándares. Identifica ejemplos de corpus actuales para una variedad de tareas de PLN [Usar]• Identifica técnicas para la recuperación de la información, traducción de lenguajes, y clasificación de textos [Usar]

Lecturas : [Nil01], [RN03], [Pon+14]

Unidad 9: Visión y percepción por computador (12)	
Competencias esperadas: a,j	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Visión Computacional <ul style="list-style-type: none"> – Adquisición de imágenes, representación, procesamiento y propiedades – Representación de formas, reconocimiento y segmentación de objetos – Análisis de movimiento • Modularidad en reconocimiento. • Enfoques de reconocimiento de patrones <ul style="list-style-type: none"> – Algoritmos de clasificación y medidas de calidad de la clasificación. – Técnicas estadísticas. 	<ul style="list-style-type: none"> • Resumir la importancia del reconocimiento de imágenes y objetos en Inteligencia Artificial (AI) e indicar varias aplicaciones significativas de esta tecnología [Usar] • Listar al menos tres aproximaciones de segmentación de imágenes, tales como algoritmos de límites (thresholding), basado en el borde y basado en regiones, junto con sus características definitorias, fortalezas y debilidades [Usar] • Implementar reconocimiento de objetos en 2d basados en la representación del contorno y/o regiones basadas en formas [Usar] • Proporcionar al menos dos ejemplos de transformación de una fuente de datos de un dominio sensorial a otro, ejemplo, datos táctiles interpretados como imágenes en 2d de una sola banda [Usar] • Implementar un algoritmo para la extracción de características en información real, ejemplo, un detector de bordes o esquinas para imágenes o vectores de coeficientes de Fourier describiendo una pequeña porción de señal de audio [Usar] • Implementar un algoritmo de clasificación que segmenta percepciones de entrada en categorías de salida y evalúa cuantitativamente la clasificación resultante [Usar] • Evaluar el desempeño de la función de extracción subyacente, en relación con al menos una aproximación alternativa posible (ya sea implementado o no) en su contribución a la tarea de clasificación (8) anterior [Usar]
Lecturas : [Nil01], [RN03], [Pon+14]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

10. BIBLIOGRAFÍA BÁSICA

- [De 06] L.N. De Castro. *Fundamentals of natural computing: basic concepts, algorithms, and applications*. CRC Press, 2006.
- [Gol89] David Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, 1989.
- [KF09] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009. ISBN: 0262013193.
- [Mit98] M. Mitchell. *An introduction to genetic algorithms*. The MIT press, 1998.
- [Mur12] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012. ISBN: 0262018020.
- [Nil01] Nils Nilsson. *Inteligencia Artificial: Una nueva visión*. McGraw-Hill, 2001.
- [Pon+14] Julio Ponce-Gallegos et al. *Inteligencia Artificial*. Iniciativa Latinoamericana de Libros de Texto Abiertos (LATIn), 2014.
- [RN03] Stuart Russell and Peter Norvig. *Inteligencia Artificial: Un enfoque moderno*. Prentice Hall, 2003.

Ministerio de Educación (MINEDU)
Sílabo 2021-I

1. CURSO

CS291. Ingeniería de Software I (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	4
2.2 Horas de teoría	:	2 (Semanal)
2.3 Horas de práctica	:	2 (Semanal)
2.4 Duración del periodo	:	16 semanas
2.5 Condición	:	Obligatorio
2.6 Modalidad	:	Presencial
2.7 Prerrequisitos	:	<ul style="list-style-type: none">• CS113. Ciencia de la Computación II. (3^{er} Sem)• CS271. Gerenciamiento de Datos I. (4^{to} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

La tarea de desarrollar software, excepto para aplicaciones sumamente simples, exige la ejecución de un proceso de desarrollo bien definido. Los profesionales de esta área requieren un alto grado de conocimiento de los diferentes modelos e proceso de desarrollo, para que sean capaces de elegir el más idóneo para cada proyecto de desarrollo. Por otro lado, el desarrollo de sistemas de mediana y gran escala requiere del uso de bibliotecas de patrones y componentes y del dominio de técnicas relacionadas al diseño basado en componentes.

5. OBJETIVOS

- Brindar al alumno un marco teórico y práctico para el desarrollo de software bajo estándares de calidad.
- Familiarizar al alumno con los procesos de modelamiento y construcción de software a través del uso de herramientas CASE.
- Los alumnos debe ser capaces de seleccionar Arquitecturas y Plataformas tecnológicas ad-hoc a los escenarios de implementación.
- Aplicar el modelamiento basado en componentes y fin de asegurar variables como calidad, costo y *time-to-market* en los procesos de desarrollo.
- Brindar a los alumnos mejores prácticas para la verificación y validación del software.

6. COMPETENCIAS

- d) Trabajar efectivamente en equipos para cumplir con un objetivo común. (**Usar**)
- i) Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (**Evaluar**)
- k) Aplicar los principios de desarrollo y diseño en la construcción de sistemas de software de complejidad variable. (**Usar**)
- d1) Desarrollo colaborativo de software utilizando repositorios de código y gestión de versiones (ej. Git, Bitbucket, SVN). (1)
- d2) Desarrollar presentaciones grupales e informes sobre tópicos específicos. (2)
- d2) Desarrollar presentaciones grupales e informes sobre tópicos específicos. (2)
- i1) Desarrollar componentes que haciendo uso de técnicas modernas de computación implementen funcionalidad que sean de utilidad para diversos sistemas de informacion. (1)
- i2) Utilizacion lenguajes y entornos de programación que permitan la implementacion y depuración de las soluciones. (2)

i4) Utilizar técnicas de verificación y validación de software. (4)

i5) Utilizar técnicas y herramientas de integración continua. (5)

k2) Desempeñarse adecuadamente como parte de un proyecto de implementación de software. (2)

k3) Aplicar metodologías de desarrollo de software. (3)

k4) Utilizar los paradigmas de programación para construir software. (4)

k5) Utilizar técnicas de algoritmos y estructuras de datos para construir software escalable. (5)

k6) Utilizar los principios de arquitectura de software para construir productos de software confiables. (6)

7. TEMAS

Unidad 1: Ingeniería de Requisitos (18)**Competencias esperadas: i,k**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Al describir los requisitos funcionales utilizando, por ejemplo, los casos de uso o historias de los usuarios.• Propiedades de requisitos, incluyendo la consistencia, validez, integridad y viabilidad.• Requisitos de software elicitación.• Descripción de datos del sistema utilizando, por ejemplo, los diagramas de clases o diagramas entidad-relación.• Requisitos no funcionales y su relación con la calidad del software.• Evaluación y uso de especificaciones de requisitos.• Requisitos de las técnicas de modelado de análisis.• La aceptabilidad de las consideraciones de certeza/incertidumbre sobre el comportamiento del software/sistema.• Prototipos.• Conceptos básicos de la especificación formal de requisitos.• Especificación de requisitos.• Validación de requisitos.• Rastreo de requisitos.	<ul style="list-style-type: none">• Enumerar los componentes clave de un caso de uso o una descripción similar de algún comportamiento que es requerido para un sistema [Evaluar]• Describir cómo el proceso de ingeniería de requisitos apoya la obtención y validación de los requisitos de comportamiento [Evaluar]• Interpretar un modelo de requisitos dada por un sistema de software simple [Evaluar]• Describir los retos fundamentales y técnicas comunes que se utilizan para la obtención de requisitos [Evaluar]• Enumerar los componentes clave de un modelo de datos (por ejemplo, diagramas de clases o diagramas ER) [Evaluar]• Identificar los requisitos funcionales y no funcionales en una especificación de requisitos dada por un sistema de software [Evaluar]• Realizar una revisión de un conjunto de requisitos de software para determinar la calidad de los requisitos con respecto a las características de los buenos requisitos [Evaluar]• Aplicar elementos clave y métodos comunes para la obtención y el análisis para producir un conjunto de requisitos de software para un sistema de software de tamaño medio [Evaluar]• Comparar los métodos ágiles y el dirigido por planes para la especificación y validación de requisitos y describir los beneficios y riesgos asociados con cada uno [Evaluar]• Usar un método común, no formal para modelar y especificar los requisitos para un sistema de software de tamaño medio [Evaluar]• Traducir al lenguaje natural una especificación de requisitos de software (por ejemplo, un contrato de componentes de software) escrito en un lenguaje de especificación formal [Evaluar]• Crear un prototipo de un sistema de software para reducir el riesgo en los requisitos [Evaluar]• Diferenciar entre el rastreo (<i>tracing</i>) hacia adelante y hacia atrás y explicar su papel en el proceso de validación de requisitos [Evaluar]
Lecturas : [ES14], [HF03]	

Unidad 2: Diseño de Software (18)**Competencias esperadas: i,k****Temas****Objetivos de Aprendizaje**

- Principios de diseño del sistema: niveles de abstracción (diseño arquitectónico y el diseño detallado), separación de intereses, ocultamiento de información, de acoplamiento y de cohesión, de reutilización de estructuras estándar.
- Diseño de paradigmas tales como diseño estructurado (descomposición funcional de arriba hacia abajo), el análisis orientado a objetos y diseño, orientado a eventos de diseño, diseño de nivel de componente, centrado datos estructurada, orientada a aspectos, orientado a la función, orientado al servicio.
- Modelos estructurales y de comportamiento de los diseños de software.
- Diseño de patrones.
- Relaciones entre los requisitos y diseños: La transformación de modelos, el diseño de los contratos, invariantes.
- Conceptos de arquitectura de software y arquitecturas estándar (por ejemplo, cliente-servidor, n-capas, transforman centrados, tubos y filtros).
- El uso de componentes de diseño: selección de componentes, diseño, adaptación y componentes de ensamblaje, componentes y patrones, componentes y objetos (por ejemplo, construir una GUI usando un standard widget set)
- Diseños de refactorización utilizando patrones de diseño
- Calidad del diseño interno, y modelos para: eficiencia y desempeño, redundancia y tolerancia a fallos, trazabilidad de los requerimientos.
- Medición y análisis de la calidad de un diseño.
- Compensaciones entre diferentes aspectos de la calidad.
- Aplicaciones en frameworks.
- Middleware: El paradigma de la orientación a objetos con middleware, requerimientos para correr y clasificar objetos, monitores de procesamiento de transacciones y el sistema de flujo de trabajo.
- Principales diseños de seguridad y codificación (cross-reference IAS/Principles of secure design).
 - Principio de privilegios mínimos
 - Principio de falla segura por defecto
 - Principio de aceptabilidad psicológica

- Formular los principios de diseño, incluyendo la separación de problemas, ocultación de información, acoplamiento y cohesión, y la encapsulación [Familiarizarse]
- Usar un paradigma de diseño para diseñar un sistema de software básico y explicar cómo los principios de diseño del sistema se han aplicado en este diseño [Usar]
- Construir modelos del diseño de un sistema de software simple los cuales son apropiado para el paradigma utilizado para diseñarlo [Usar]
- En el contexto de un paradigma de diseño simple, describir uno o más patrones de diseño que podrían ser aplicables al diseño de un sistema de software simple [Familiarizarse]
- Para un sistema simple adecuado para una situación dada, discutir y seleccionar un paradigma de diseño apropiado [Usar]
- Crear modelos apropiados para la estructura y el comportamiento de los productos de software desde la especificaciones de requisitos [Usar]
- Explicar las relaciones entre los requisitos para un producto de software y su diseño, utilizando los modelos apropiados [Evaluar]
- Para el diseño de un sistema de software simple dentro del contexto de un único paradigma de diseño, describir la arquitectura de software de ese sistema [Familiarizarse]
- Dado un diseño de alto nivel, identificar la arquitectura de software mediante la diferenciación entre las arquitecturas comunes de software, tales como 3 capas (*3-tier*), *pipe-and-filter*, y cliente-servidor [Familiarizarse]
- Investigar el impacto de la selección arquitecturas de software en el diseño de un sistema simple [Evaluar]
- Aplicar ejemplos simples de patrones en un diseño de software [Usar]
- Describir una manera de refactorar y discutir cuando esto debe ser aplicado [Familiarizarse]
- Seleccionar componentes adecuados para el uso en un diseño de un producto de software [Usar]
- Explicar cómo los componentes deben ser adaptados para ser usados en el diseño de un producto de software [Familiarizarse]
- Diseñar un contrato para un típico componente de software pequeño para el uso de un dado sistema [Usar]
- Discutir y seleccionar la arquitectura de software

Unidad 3: Construcción de Software (24)	
Competencias esperadas: i,k	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> ● Prácticas de codificación: técnicas, idiomas/patrones, mecanismos para construcción de programas de calidad: <ul style="list-style-type: none"> – Prácticas de codificación defensiva – Prácticas de codificación segura – Utilizando mecanismos de manejo de excepciones para hacer el programa más robusto, tolerante a fallas ● Normas de codificación. ● Estrategias de integración. ● Desarrollando contexto: "campo verde" frente a la base de código existente : <ul style="list-style-type: none"> – Análisis de cambio impacto – Cambio de actualización ● Los problemas de seguridad potenciales en los programas : <ul style="list-style-type: none"> – Buffer y otros tipos de desbordamientos – Condiciones elemento Race – Inicialización incorrecta, incluyendo la elección de los privilegios – Entrada Comprobación – Suponiendo éxito y corrección – La validación de las hipótesis 	<ul style="list-style-type: none"> ● Describir técnicas, lenguajes de codificación y mecanismos de implementación para conseguir las propiedades deseadas, tales como la confiabilidad, la eficiencia y la robustez [Evaluar] ● Construir código robusto utilizando los mecanismos de manejo de excepciones [Evaluar] ● Describir la codificación segura y prácticas de codificación de defensa [Evaluar] ● Seleccionar y utilizar un estándar de codificación definido en un pequeño proyecto de software [Evaluar] ● Comparar y contrastar las estrategias de integración incluyendo: de arriba hacia abajo (<i>top-down</i>), de abajo hacia arriba (<i>bottom-up</i>), y la integración Sándwich [Evaluar] ● Describir el proceso de analizar e implementar los cambios a la base de código desarrollado para un proyecto específico [Evaluar] ● Describir el proceso de analizar e implementar los cambios a una gran base de código existente [Evaluar] ● Reescribir un programa sencillo para eliminar vulnerabilidades comunes, tales como desbordamientos de búffer, desbordamientos de enteros y condiciones de carrera [Evaluar] ● Escribir un componente de software que realiza alguna tarea no trivial y es resistente a errores en la entrada y en tiempo de ejecución [Evaluar]
Lecturas : [ES14], [HF03]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

***** EVALUATION MISSING *****

10. BIBLIOGRAFÍA BÁSICA

- [ES14] Bert Bates Eric Freeman Elisabeth Robson and Kathy Sierra. *Head First Design Patterns*. 2nd. O'Reilly Media, Inc, July 2014.
- [HF03] Brian Lyons Hans-Erik Eriksson Magnus Penker and Davis Fado. *UML 2 Toolkit*. 2nd. Wiley, Oct. 2003.

Ministerio de Educación (MINEDU)
Sílabo 2021-I

1. CURSO

CS2H1. Experiencia de Usuario (UX) (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	3
2.2 Horas de teoría	:	1 (Semanal)
2.3 Horas de práctica	:	4 (Semanal)
2.4 Duración del periodo	:	16 semanas
2.5 Condición	:	Obligatorio
2.6 Modalidad	:	Presencial
2.7 Prerrequisitos	:	CS113. Ciencia de la Computación II. (3 ^{er} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

El lenguaje ha sido una de las creaciones más significativas de la humanidad. Desde el lenguaje corporal y gestual, pasando por la comunicación verbal y escrita, hasta códigos simbólicos icónicos y otros, ha posibilitado interacciones complejas entre los seres humanos y facilitado considerablemente la comunicación de información. Con la invención de dispositivos automáticos y semiautomáticos, entre los que se cuentan las computadoras, la necesidad de lenguajes o interfaces para poder interactuar con ellos, ha cobrado gran importancia.

La usabilidad del software, aunada a la satisfacción del usuario y su incremento de productividad, depende de la eficacia de la Interfaz Usuario-Computador. Tanto es así, que a menudo la interfaz es el factor más importante en el éxito o el fracaso de cualquier sistema computacional. El diseño e implementación de adecuadas Interfaces Humano-Computador, que además de cumplir los requisitos técnicos y la lógica transaccional de la aplicación, considere las sutiles implicaciones psicológicas, culturales y estéticas de los usuarios, consume buena parte del ciclo de vida de un proyecto software, y requiere habilidades especializadas, tanto para la construcción de las mismas, como para la realización de pruebas de usabilidad.

5. OBJETIVOS

- Conocer y aplicar criterios de usabilidad y accesibilidad al diseño y construcción de interfaces humano-computador, buscando siempre que la tecnología se adapte a las personas y no las personas a la tecnología.
- Que el alumno tenga una visión centrada en la experiencia de usuario al aplicar apropiados enfoques conceptuales y tecnológicos.
- Entender como la tecnologica emergente hace posible nuevos estilos de interacción.
- Determinar los requerimientos básicos a nivel de interfaces, hardware y software para la construcción de ambientes inmersivos.

6. COMPETENCIAS

- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Familiarizarse**)
- c) Diseñar, implementar y evaluar un sistema, proceso, componente o programa computacional para alcanzar las necesidades deseadas. (**Evaluar**)
- d) Trabajar efectivamente en equipos para cumplir con un objetivo común. (**Usar**)
- o) Mejorar las condiciones de la sociedad poniendo la tecnología al servicio del ser humano. (**Familiarizarse**)

■NoSpecificOutcomes■

7. TEMAS

Unidad 1: Fundamentos (8)	
Competencias esperadas: b	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Contextos para IHC (cualquiera relacionado con una interfaz de usuario, p.e., página web, aplicaciones de negocios, aplicaciones móviles y juegos) • Heurística de usabilidad y los principios de pruebas de usabilidad. • Procesos para desarrollo centrado en usuarios, p.e., enfoque inicial en usuarios, pruebas empíricas, diseño iterativo. • Principios del buen diseño y buenos diseñadores; ventajas y desventajas de ingeniería. • Diferentes medidas para evaluación, p.e., utilidad, eficiencia, facilidad de aprendizaje, satisfacción de usuario. 	<ul style="list-style-type: none"> • Discutir por qué el desarrollo de software centrado en el hombre es importante [Familiarizarse] • Define un proceso de diseño centrado en el usuario que de forma explícita considere el hecho que un usuario no es como un desarrollador o como sus conocimientos [Familiarizarse] • Resumir los preceptos básicos de la interacción psicológica y social [Familiarizarse] • Desarrollar y usar un vocabulario conceptual para analizar la interacción humana con el software: disponibilidad, modelo conceptual, retroalimentación, y demás [Familiarizarse]
Lecturas : [Dix+04], [Sto+05], [RS11]	

Unidad 2: Factores Humanos (8)	
Competencias esperadas: b	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Modelos cognoscitivos que informan diseño de interacciones, p.e., atención, percepción y reconocimiento, movimiento, memoria, golfos de expectativa y ejecución. • Capacidades físicas que informan diseño de interacción, p.e. percepción del color, ergonomía. • Accesibilidad, p.e., interfaces para poblaciones con diferentes habilidades (p.e., invidentes, discapacitados) • Interfaces para grupos de población de diferentes edades (p.e., niños, mayores de 80) 	<ul style="list-style-type: none"> • Crear y dirigir una simple prueba de usabilidad para una aplicación existente de software [Familiarizarse]
Lecturas : [Dix+04], [Sto+05], [RS11], [Mat11], [Nor04]	

Unidad 3: Diseño y Testing centrados en el usuario (16)**Competencias esperadas: b,c**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Enfoque y características del proceso de diseño.• Requerimientos de funcionalidad y usabilidad.• Técnicas de recolección de requerimientos, ej. entrevistas, encuestas, etnografía e investigación contextual.• Técnicas y herramientas para el análisis y presentación de requerimientos ej. reportes, personas.• Análisis de tareas, incluidos los aspectos cualitativos de la generación de modelos de análisis de tareas.• Consideración de IHC como una disciplina de diseño:<ul style="list-style-type: none">– Sketching– Diseño participativo– Sketching– Diseño participativo• Técnicas de creación de prototipos y herramientas, ej. bosquejos, <i>storyboards</i>, prototipos de baja fidelidad, esquemas de página.• Prototipos de baja fidelidad (papel)• Técnicas de evaluación cuantitativa ej. evaluación Keystroke-level.• Evaluación sin usuarios, usando ambas técnicas cualitativas y cuantitativas. Ej. Revisión estructurada, GOMS, análisis basado en expertos, heurísticas, lineamientos y estándar.• Evaluación con usuarios. Ej. Observación, Método de pensamiento en voz alta, entrevistas, encuestas, experimentación.• Desafíos para la evaluación efectiva, por ejemplo, toma de muestras, la generalización.• Reportar los resultados de las evaluaciones.• Internacionalización, diseño para usuarios de otras culturas, intercultural.	<ul style="list-style-type: none">• Llevar a cabo una evaluación cuantitativa y discutir / informar sobre los resultados [Familiarizarse]• Para un grupo de usuarios determinado, realizar y documentar un análisis de sus necesidades [Familiarizarse]• Discutir al menos un standard nacional o internacional de diseño de interfaz de usuario [Familiarizarse]• Explicar cómo el diseño centrado en el usuario complementa a otros modelos de proceso software [Familiarizarse]• Utilizar <i>lo-fi</i> (baja fidelidad) técnicas de prototipado para recopilar y reportar, las respuestas del usuario [Usar]• Elegir los métodos adecuados para apoyar el desarrollo de una específica interfaz de usuario [Evaluar]• Utilizar una variedad de técnicas para evaluar una interfaz de usuario dada [Evaluar]• Comparar las limitaciones y beneficios de los diferentes métodos de evaluación [Evaluar]
Lecturas : [Dix+04], [Sto+05], [RS11], [Mat11], [Bux07]	

Unidad 4: Diseño de Interacción (8)**Competencias esperadas: b,c,d,o**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Principios de interfaces gráficas de usuario (GUIs)• Elementos de diseño visual (disposición, color, fuentes, etiquetado)• Manejo de fallas humanas/sistema.• Estándares de interfaz de usuario.• Presentación de información: navegación, representación, manipulación.• Técnicas de animación de interfaz (ej. grafo de escena)• Clases Widget y bibliotecas.• Internacionalización, diseño para usuarios de otras culturas, intercultural.• Elección de estilos de interacción y técnicas de interacción.	<ul style="list-style-type: none">• Crear una aplicación simple, junto con la ayuda y la documentación, que soporta una interfaz gráfica de usuario [Usar]
Lecturas : [Dix+04], [Sto+05], [RS11], [Joh10], [Mat11], [LS06]	

Unidad 5: Nuevas Tecnologías Interactivas (8)**Competencias esperadas: o**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Elección de estilos de interacción y técnicas de interacción.• Enfoques para el diseño, implementación y evaluación de la interacción sin mouse<ul style="list-style-type: none">– Interfaces táctiles y multitáctiles.– Interfaces compartidas, incorporadas y grandes– Nuevas modalidades de entrada (tales como datos de sensores y localización)– Nuevas ventanas, por ejemplo, iPhone, Android– Reconocimiento de voz y procesamiento del lenguaje natural– Interfaces utilizables y tangibles– Interacción persuasiva y emoción– Tecnologías de interacción ubicuas y contextuales (Ubicomp)– Inferencia bayesiana (por ejemplo, texto predictivo, orientación guiada)– Visualización e interacción de ambiente / periféricos• Salida:<ul style="list-style-type: none">– Sonido– Visualización estereoscópica– Forzar la simulación de retroalimentación, dispositivos hápticos• Arquitectura de Sistemas:<ul style="list-style-type: none">– Motores de Juego– Realidad Aumentada móvil– Simuladores de vuelo– CAVEs– Imágenes médicas	<ul style="list-style-type: none">• Describe cuando son adecuadas las interfaces sin uso de ratón [Familiarizarse]• Comprende las posibilidades de interacción que van más allá de las interfaces de ratón y puntero [Familiarizarse]• Discute las ventajas (y desventajas) de las interfaces no basadas en ratón [Usar]• Describir el modelo óptico realizado por un sistema de gráficos por computadora para sintetizar una visión estereoscópica [Familiarizarse]• Describir los principios de las diferentes tecnologías de seguimiento de espectador [Familiarizarse]• Determinar los requerimientos básicos en interfaz, software, hardware, y configuraciones de software de un sistema VR para una aplicación específica [Evaluar]
Lecturas : [Dix+04], [Sto+05], [RS11], [WW11], [Mat11]	

Unidad 6: Colaboración y Comunicación (8)	
Competencias esperadas: d,o	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • La comunicación asíncrona en grupo, por ejemplo, el correo electrónico, foros, redes sociales. • Medios de comunicación social, informática social, y el análisis de redes sociales. • Colaboración en línea, espacios "inteligentes" y aspectos de coordinación social de tecnologías de flujo de trabajo. • Comunidades en línea. • Personajes de Software y agentes inteligentes, mundos virtuales y avatares. • Psicología Social 	<ul style="list-style-type: none"> • Describir la diferencia entre la comunicación sincrónica y asíncrona [Familiarizarse] • Comparar los problemas de IHC en la interacción individual con la interacción del grupo [Familiarizarse] • Discuta varias problemas de interés social planteados por el software colaborativo [Usar] • Discutir los problemas de IHC en software que personifica la intención humana [Evaluar]
Lecturas : [Dix+04], [Sto+05], [RS11]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

***** EVALUATION MISSING *****

10. BIBLIOGRAFÍA BÁSICA

- [Bux07] Bill Buxton. *Sketching User Experiences: Getting the Design Right and the Right Design*. Morgan Kaufmann Publishers Inc., 2007.
- [Dix+04] Alan Dix et al. *Human-computer Interaction*. 3 ed. Prentice-Hall, Inc, 2004.
- [Joh10] Jeff Johnson. *Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Rules*. 3 ed. Morgan Kaufmann Publishers Inc., 2010.
- [LS06] M. Leavitt and B. Shneiderman. *Research-Based Web Design & Usability Guidelines*. Health and Human Services Dept, 2006.
- [Mat11] Lukas Mathis. *Designed for Use: Create Usable Interfaces for Applications and the Web*. Pragmatic Bookshelf, 2011.
- [Nor04] Donald A. Norman. *Emotional Design: Why We Love (or Hate) Everyday Things*. Basic Book, 2004.
- [RS11] Y. Rogers and J Sharp H. & Preece. *Interaction Design: Beyond Human-Computer Interaction*. 3 ed. John Wiley and Sons Ltd, 2011.
- [Sto+05] D. Stone et al. *User Interface Design and Evaluation*. Morgan Kaufmann Series in Interactive Technologies, 2005.

[WW11] D. Wigdor and D. Wixon. *Brave NUI World: Designing Natural User Interfaces for Touch and Gesture*. Morgan Kaufmann Publishers Inc, 2011.

Ministerio de Educación (MINEDU)
Sílabo 2021-I

1. CURSO

CB111. Física Computacional (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	: 4
2.2 Horas de teoría	: 2 (Semanal)
2.3 Horas de práctica	: 2 (Semanal)
2.4 Duración del periodo	: 16 semanas
2.5 Condición	: Obligatorio
2.6 Modalidad	: Presencial
2.7 Prerrequisitos	: MA100. Matemática I. (1 ^{er} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Física I es un curso que le permitirá al estudiante entender las leyes de física de macropartículas y micropartículas considerado desde un punto material hasta un sistemas de partículas; debiéndose tener en cuenta que los fenómenos aquí estudiados se relacionan a la física clásica: Cinemática, Dinámica, Trabajo y Energía; además se debe asociar que éstos problemas deben ser resueltos con algoritmos computacionales.

Poseer capacidad y habilidad en la interpretación de problemas clásicos con condiciones de frontera reales que contribuyen en la elaboración de soluciones eficientes y factibles en diferentes áreas de la Ciencia de la Computación.

5. OBJETIVOS

- Conocer los principios básicos de los fenómenos que gobiernan la física clásica.
- Aplicar los principios básicos a situaciones específicas y poder asociarlos con situaciones reales.
- Analizar algunos de los fenómenos físicos así como su aplicación a situaciones reales.

6. COMPETENCIAS

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Familiarizarse**)
- i) Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (**Familiarizarse**)
- j) Aplicar la base matemática, principios de algoritmos y la teoría de la CS en el modelamiento y diseño de sistemas. (**Usar**)

■NoSpecificOutcomes■

7. TEMAS

Unidad 1: Vectores (6)	
Competencias esperadas: C1	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Análisis dimensional.• Vectores. Propiedades. Operaciones.• Caso práctico: Estimación de fuerzas.	<ul style="list-style-type: none">• Entender y trabajar con las magnitudes físicas del SI.[Usar]• Abstracter de la naturaleza los conceptos físicos rigurosos y representarlos en modelos vectoriales.[Usar]• Entender y aplicar los conceptos vectoriales a problemas físicos reales.[Usar]
Lecturas : [Bur06], [Res07], [Ser09], [Tip09]	

Unidad 2: (6)	
Competencias esperadas: C20	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Primera y tercera Ley de Newton. • Diagrama de cuerpo libre. • Primera condición de equilibrio. • Caso práctico: Estimación de la fuerza humana. • Segunda condición de equilibrio. • Torque. • Casos prácticos: Aplicaciones en dispositivos mecánicos. • Fricción. 	<ul style="list-style-type: none"> • Conocer los conceptos que rigen la primera Ley y tercera Ley de Newton. • Conocer y aplicar los conceptos de la primera y segunda condición de equilibrio. • Capacidad para resolver problemas de casos prácticos. • Entender el concepto de fricción y resolver problemas.
Lecturas : [Bur06], [Res07], [Ser09], [Tip09]	

Unidad 3: (6)	
Competencias esperadas: C24	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Posición, Velocidad, Aceleración. • Gráficas de movimiento. • Casos prácticos: Representación gráfica de movimiento utilizando Excel. • Movimiento circular. • Velocidad angular y velocidad tangencial. • Mecanismos rotativos. • Caso práctico: Operación de la caja de cambios de un automóvil. 	<ul style="list-style-type: none"> • Poder determinar la posición, velocidad y aceleración de un cuerpo. • Conocer el concepto de composición de movimientos y saberlo aplicar, en la descripción de un movimiento circular. • Conocer el significado de las componentes tangencial y normal de la aceleración y saberlas calcular en un instante determinado. • Utilizar excel para el procesamiento de datos experimentales.
Lecturas : [Bur06], [Res07], [Ser09], [Tip09]	

Unidad 4: (6)	
Competencias esperadas: C1	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Segunda Ley de Newton. • Fuerza y movimiento. • Momento de inercia. 	<ul style="list-style-type: none"> • Aplicar las leyes de Newton en la solución de problemas. • Describir las diversas interacciones por sus correspondientes fuerzas. • Determinar el momento de inercia de un cuerpo usando un método dinámico
Lecturas : [Bur06], [Res07], [Ser09], [Tip09]	

Unidad 5: (6)	
Competencias esperadas: C20	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Trabajo. • Fuerzas constantes. • Fuerzas variables. • Potencia. • Caso práctico: Estimación de la potencia de una planta hidroeléctrica. 	<ul style="list-style-type: none"> • Comprender el concepto de Trabajo. • Comprender y aplicar el concepto de Potencia a la resolución de problemas. • Resolver problemas.
Lecturas : [Bur06], [Res07], [Ser09], [Tip09]	

Unidad 6: (6)	
Competencias esperadas: C24	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Tipos de energía. • Conservación de la energía. • Dinámica de un sistema de partículas. • Colisiones. 	<ul style="list-style-type: none"> • Conocer los tipos de energía que existen. • Aplicar el principio de conservación de la energía mecánica a distintas situaciones, diferenciando aquellas en las que la energía total no se mantiene constante. • Aplicar los principios de conservación del momento lineal y de la energía a un sistema aislado de dos o más partículas interactuantes.
Lecturas : [Bur06], [Res07], [Ser09], [Tip09]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

***** EVALUATION MISSING *****

10. BIBLIOGRAFÍA BÁSICA

[Bur06] S. Burbano. *Física General*. Alfaomega, 2006.

[Res07] D. Resnik R. y Halliday. *Física*. 5th. Vol. 1. Patria, 2007.

[Ser09] J.W. Serway R. A. y Jewett. *Física para Ciencias e Ingeniería con Física Moderna*. 7th. Vol. 1. Cengage Learning, 2009.

[Tip09] G. Tipler P. y Mosca. *Física para la ciencia y la tecnología*. 7th. Vol. 1. Reverte, 2009.

Ministerio de Educación (MINEDU)
Sílabo 2021-I

1. CURSO

ID203. Inglés V (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	3
2.2 Horas de teoría	:	2 (Semanal)
2.3 Horas de práctica	:	-
2.4 Duración del periodo	:	16 semanas
2.5 Condición	:	Obligatorio
2.6 Modalidad	:	Presencial
2.7 Prerrequisitos	:	ID202. Inglés IV. (4 ^{to} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Parte fundamental de la formación integral de un profesional es la habilidad de comunicarse en un idioma extranjero además del propio idioma nativo. No solamente amplía su horizonte cultural sino que permite una visión más humana y comprensiva de la vida. En el caso de los idiomas extranjeros, indudablemente el Inglés es el más práctico porque es hablado alrededor de todo el mundo. No hay país alguno donde éste no sea hablado. En las carreras relacionadas con los servicios al turista el inglés es tal vez la herramienta práctica más importante que el alumno debe dominar desde el primer momento como parte de su formación integral.

5. OBJETIVOS

- Incrementar la capacidad y la fluidez de hablar y entender el idioma Inglés.
- Hacer que los alumnos interactuen con mayor énfasis en la creación de diálogos.

6. COMPETENCIAS

- f) Comunicarse efectivamente con audiencias diversas. (Usar)
- i) Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (Usar)

■NoSpecificOutcomes■

7. TEMAS

Unidad 1: It's a wonderful world (0)	
Competencias esperadas: 5	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Verbos Auxiliares• Tiempos Verbales• Preguntas y oraciones negativas• Respuestas cortas• Formación de palabras• Expresiones coloquiales• rrección de errores	<ul style="list-style-type: none">• Al terminar la primera unidad, cada uno de los alumnos, comprendiendo la gramática de los auxiliares y de los diferentes tipos de oración es capaz de expresar una mayor cantidad de expresiones de tiempo y además usar preposiciones para describir lugares y tiempos variados. Además es capaz de analizar y expresar ideas acerca de formación de palabras.
Lecturas : [SJ02a], [SJ02c], [SJ02b], [Cam06], [Mac99]	

Unidad 2: Happiness! (0)	
Competencias esperadas: 5	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Presente Simple • Presente Continuo • Voz Pasiva en Presente • Verbos para deportes y tiempo libre • Tipos de números y fechas • Inventos/Mundo Moderno • Corrección de errores 	<ul style="list-style-type: none"> • Al terminar la segunda unidad, los alumnos habiendo identificado la forma de expresar deportes y actividades de tiempo libre. Utiliza todo tipo de expresiones numéricas. Expresar situaciones y estados relacionados con formas de presente. Explica y aplica vocabulario de actividades al aire libre.
Lecturas : [SJ02a], [SJ02c], [SJ02b], [Cam06], [Mac99]	

Unidad 3: Telling tales! (0)	
Competencias esperadas: 5	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Tiempo Pasado Simple • Pasado Continuo • Voz Pasiva en Pasado • Vocabulario de Arte y Literatura • Expresiones para dar y pedir opiniones • Cuentos e historias 	<ul style="list-style-type: none"> • Al terminar la tercera unidad, los alumnos habiendo reconocido las características de las formas de pasado pasivo, utilizan éstos para hacer descripciones de diversos tipos. Describen arte y literatura y dan indicaciones de opinión. Utilizarán conjunciones para unir ideas tipo.
Lecturas : [SJ02a], [SJ02c], [SJ02b], [Cam06], [Mac99]	

Unidad 4: Doing the right thing! (0)	
Competencias esperadas: 5	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Verbos Auxiliares de Modo I • Oraciones Afirmativas, Negativas y Preguntas con Modals • Uso de nacionalidades i otros adjetivos • Expresiones de pedidos y Ofrecimientos • Guía de los Buenos Modales • Llenado de Formatos • Símbolos fonéticos 	<ul style="list-style-type: none"> • Al terminar la cuarta unidad, los alumnos habiendo identificado la idea de expresar ideas de modos de acciones que suceden en el momento o que se relacionan a cualquier tiempo estructuran oraciones en Presente. Expresan ideas de nacionalidades y hacen pedidos y ofrecimientos variados.
Lecturas : [SJ02a], [SJ02c], [SJ02b], [Cam06], [Mac99]	

Unidad 5: On the move! (0)	
Competencias esperadas: 5	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Futuro con Hill • Oraciones en Tiempo Futuro con going to • Uso de might para futuro • Expresiones del clima • Vocabulario del clima • Expresiones para hoteles y transporte • E-mails 	<ul style="list-style-type: none"> • Al finalizar la quinta unidad, los alumnos, a partir de la comprensión del tiempo futuro, elaborarán oraciones utilizando los elementos necesarios. Asimilarán además la necesidad de expresar ideas del clima. Adquirirán vocabulario para describir uso de transporte público. Se presentará expresiones para hacer pedidos en hoteles.
Lecturas : [SJ02a], [SJ02c], [SJ02b], [Cam06], [Mac99]	

Unidad 6: I just love it! (0)	
Competencias esperadas: 5	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Preguntas con formas Like • Patrones Verbales II • Vocabulario de Comida, Lugares y ocupaciones • Palabras que van unidas en contexto • Expresiones para vistas y sonidos • Composición de Impresiones personales 	<ul style="list-style-type: none"> • Al finalizar la sexta unidad, los alumnos habiendo conocido los fundamentos de la estructuración de preguntas con like y con patrones verbales trabajos aplicados a contextos adecuados. Enfatizan la diferencia entre comidas, lugares y personas. Describen vistas y sonidos. Utilizan expresiones comparar la vida diaria en diferentes lugares. Asumen la idea de estilos de vida diferentes.
Lecturas : [SJ02a], [SJ02c], [SJ02b], [Cam06], [Mac99]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

***** EVALUATION MISSING *****

10. BIBLIOGRAFÍA BÁSICA

[Cam06] Cambridge. *Diccionario Inglés-Español Cambridge*. Editorial Oxford, 2006.

[Mac99] James MacGrew. *Focus on Grammar Basic*. Editorial Oxford, 1999.

- [SJ02a] Liz Soars and John. *American Headway N 3 Student Book*. Editorial Oxford, 2002.
- [SJ02b] Liz Soars and John. *American Headway N 3 Teachers Book*. Editorial Oxford, 2002.
- [SJ02c] Liz Soars and John. *American Headway N 3 Work Book*. Editorial Oxford, 2002.

Ministerio de Educación (MINEDU)
Sílabo 2021-I

1. CURSO

CS292. Ingeniería de Software II (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	4
2.2 Horas de teoría	:	2 (Semanal)
2.3 Horas de práctica	:	2 (Semanal)
2.4 Duración del periodo	:	16 semanas
2.5 Condición	:	Obligatorio
2.6 Modalidad	:	Presencial
2.7 Prerrequisitos	:	CS291. Ingeniería de Software I. (5 ^{to} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Los tópicos de este curso extienden las ideas del diseño y desarrollo de software desde la secuencia de introducción a la programación para abarcar los problemas encontrados en proyectos de gran escala. Es una visión más amplia y completa de la Ingeniería de Software apreciada desde un punto de vista de Proyectos.

5. OBJETIVOS

- Capacitar a los alumnos para formar parte y definir equipos de desarrollo de software que afronten problemas de envergadura real.
- Familiarizar a los alumnos con el proceso de administración de un proyecto de software de tal manera que sea capaz de crear, mejorar y utilizar herramientas y métricas que le permitan realizar la estimación y seguimiento de un proyecto de software.
- Crear, evaluar e implementar un plan de prueba para segmentos de código de tamaño medio , Distinguir entre los diferentes tipos de pruebas , sentar las bases para crear, mejorar los procedimientos de prueba y las herramientas utilizadas con ese propósito.
- Seleccionar con justificación un apropiado conjunto de herramientas para soportar el desarrollo de un rango de productos de software.
- Crear, mejorar y utilizar los patrones existentes para el mantenimiento de software . Dar a conocer las características y patrones de diseño para la reutilización de software.
- Identificar y discutir diferentes sistemas especializados , crear , mejorar y utilizar los patrones especializados para el diseño , implementación , mantenimiento y prueba de sistemas especializados

6. COMPETENCIAS

- c) Diseñar, implementar y evaluar un sistema, proceso, componente o programa computacional para alcanzar las necesidades deseadas. (**Usar**)
- d) Trabajar efectivamente en equipos para cumplir con un objetivo común. (**Usar**)
- i) Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (**Evaluar**)
- k) Aplicar los principios de desarrollo y diseño en la construcción de sistemas de software de complejidad variable. (**Usar**)
- c1) Identificar e implementar estructuras de datos para la solución de un problema computacional. (**1**)
- c3) Utilizar distintas herramientas y lenguajes de programación en los componentes de software (*Full stack*). (**3**)
- c4) Diseñar e implementar arquitecturas de software escalables en distintas plataformas. (**4**)

- d1) Desarrollo colaborativo de software utilizando repositorios de código y gestión de versiones (ej. Git, Bitbucket, SVN). (1)
- d2) Desarrollar presentaciones grupales e informes sobre tópicos específicos. (2)
- d2) Desarrollar presentaciones grupales e informes sobre tópicos específicos. (2)
- i1) Desarrollar componentes que haciendo uso de técnicas modernas de computación implementen funcionalidad que sean de utilidad para diversos sistemas de informacion. (1)
- i2) Utilización lenguajes y entornos de programación que permitan la implementación y depuración de las soluciones. (2)
- i4) Utilizar técnicas de verificación y validación de software. (4)
- i5) Utilizar técnicas y herramientas de integración continua. (5)
- k2) Desempeñarse adecuadamente como parte de un proyecto de implementación de software. (2)
- k3) Aplicar metodologías de desarrollo de software. (3)
- k4) Utilizar los paradigmas de programación para construir software. (4)
- k5) Utilizar técnicas de algoritmos y estructuras de datos para construir software escalable. (5)
- k6) Utilizar los principios de arquitectura de software para construir productos de software confiables. (6)

7. TEMAS

Unidad 1: Herramientas y Entornos (12)	
Competencias esperadas: c,f,i	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Administración de configuración de software y control de versiones. • Administración de despliegues. • Análisis de requerimientos y herramientas para modelado del diseño. • Herramientas de <i>testing</i> incluyendo herramientas de análisis estático y dinámico. • Entornos de programación que automatizan el proceso de construcción de partes de programa (ejem., construcciones automatizadas) <ul style="list-style-type: none"> – Integración continua. • Mecanismos y conceptos de herramientas de integración. 	<ul style="list-style-type: none"> • Administración de configuración de software y control de versiones. [Usar] • Administración de despliegues. [Usar] • Análisis de requerimientos y herramientas para modelado del diseño. [Usar] • Herramientas de <i>testing</i> incluyendo herramientas de análisis estático y dinámico. [Usar] • Entornos de programación que automatizan el proceso de construcción de partes de programa (ejem., construcciones automatizadas) <ul style="list-style-type: none"> – Integración continua. [Usar] • Mecanismos y conceptos de herramientas de integración. [Usar]
Lecturas : [Pre04], [Blu92], [Sch04], [WK00], [Key04], [WA02], [PS01], [Sch04], [Mon96], [Amb01], [Con00], [Oqu03]	

Unidad 2: Verificación y Validación de Software (12)

Competencias esperadas: c,f,i

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Verificación y validación de conceptos. • Inspecciones, revisiones, auditorias. • Tipos de pruebas, incluyendo la interfaz humano computador, usabilidad, confiabilidad, seguridad, desempeño para la especificación. • Fundamentos de testeo: <ul style="list-style-type: none"> – Pruebas de Unit, integración, validación y de Sistema – Creación de plan de pruebas y generación de casos de test – Técnicas de test de caja negra y caja blanca – Test de regresión y automatización de pruebas • Seguimiento de defectos. • Limitaciones de testeo en dominios particulares, tales como sistemas paralelos o críticos en cuanto a seguridad. • Enfoques estáticos y enfoques dinámicos para la verificación. • Desarrollo basado en pruebas. • Plan de Validación, documentación para validación. • Pruebas Orientadas a Objetos, Sistema de Pruebas. • Verificación y validación de artefactos no codificados (documentación, archivos de ayuda, materiales de entrenamiento) • Logeo fallido, error crítico y apoyo técnico para dichas actividades. • Estimación fallida y terminación de las pruebas que incluye la envios por defecto. 	<ul style="list-style-type: none"> • Distinguir entre la validación y verificación del programa [Usar] • Describir el papel que las herramientas pueden desempeñar en la validación de software [Usar] • Realizar, como parte de una actividad de equipo, una inspección de un segmento de código de tamaño medio [Usar] • Describir y distinguir entre diferentes tipos y niveles de pruebas (unitaria, integración, sistemas y aceptación) [Usar] • Describir técnicas para identificar casos de prueba representativos para integración, regresión y pruebas del sistema [Usar] • Crear y documentar un conjunto de pruebas para un segmento de código de mediano tamaño [Usar] • Describir cómo seleccionar buenas pruebas de regresión y automatizarlas [Usar] • Utilizar una herramienta de seguimiento de defectos para manejar defectos de software en un pequeño proyecto de software [Usar] • Discutir las limitaciones de las pruebas en un dominio particular [Usar] • Evaluar un banco de pruebas (<i>a test suite</i>) para un segmento de código de tamaño medio [Usar] • Comparar los enfoques estáticos y dinámicos para la verificación [Usar] • Identificar los principios fundamentales de los métodos de desarrollo basado en pruebas y explicar el papel de las pruebas automatizadas en estos métodos [Usar] • Discutir los temas relacionados con las pruebas de software orientado a objetos [Usar] • Describir las técnicas para la verificación y validación de los artefactos de no código [Usar] • Describir los enfoques para la estimación de fallos [Usar] • Estimar el número de fallos en una pequeña aplicación de software basada en la densidad de defectos y siembra de errores [Usar] • Realizar una inspección o revisión del de código fuente de un software para un proyecto de software de tamaño pequeño o mediano [Usar]
Lecturas : [Pre04], [Blu92], [Sch04], [WK00], [Key04], [WA02], [PS01], [Sch04], [Mon96], [Amb01], [Con00], [Oqu03]	

Unidad 3: Evolución de Software (12)

Competencias esperadas: c,f,i

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Desarrollo de Software en el contexto de código grande pre existente<ul style="list-style-type: none">– Cambios de software– Preocupaciones y ubicación de preocupaciones– <i>Refactoring</i>• Evolución de Software.• Características de Software mantenible.• Sistemas de Reingeniería.• Reuso de Software.<ul style="list-style-type: none">– Segmentos de código– Bibliotecas y <i>frameworks</i>– Componentes– Líneas de Producto	<ul style="list-style-type: none">• Identificar los problemas principales asociados con la evolución del software y explicar su impacto en el ciclo de vida del software [Usar]• Estimar el impacto del cambio de requerimientos en productos existentes de tamaño medio [Usar]• Usar refactorización en el proceso de modificación de un componente de software [Usar]• Estudiar los desafíos de mejorar sistemas en un entorno cambiante [Usar]• Perfilar los procesos de pruebas de regresión y su rol en el manejo de versiones [Usar]• Estudiar las ventajas y desventajas de diferentes tipos de niveles de confiabilidad [Usar]
Lecturas : [Pre04], [Blu92], [Sch04], [WK00], [Key04], [WA02], [PS01], [Sch04], [Mon96], [Amb01], [Con00], [Oqu03]	

Unidad 4: Gestión de Proyectos de Software (12)	
Competencias esperadas: c,f,i	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • La participación del equipo: <ul style="list-style-type: none"> – Procesos elemento del equipo, incluyendo responsabilidades de tarea, la estructura de reuniones y horario de trabajo – Roles y responsabilidades en un equipo de software – Equipo de resolución de conflictos – Los riesgos asociados con los equipos virtuales (comunicación, la percepción, la estructura) • Estimación de esfuerzo (a nivel personal) • Riesgo. <ul style="list-style-type: none"> – El papel del riesgo en el ciclo de vida – Categorías elemento de riesgo, incluyendo la seguridad, la seguridad, mercado, finanzas, tecnología, las personas, la calidad, la estructura y el proceso de • Gestión de equipos: <ul style="list-style-type: none"> – Organización de equipo y la toma de decisiones – Roles de identificación y asignación – Individual y el desempeño del equipo de evaluación • Gestión de proyectos: <ul style="list-style-type: none"> – Programación y seguimiento de elementos – Herramientas de gestión de proyectos – Análisis de Costo/Beneficio • Software de medición y técnicas de estimación. • Aseguramiento de la calidad del software y el rol de las mediciones. • Riesgo. <ul style="list-style-type: none"> – Identificación de riesgos y gestión. – Análisis riesgo y evaluación. – La tolerancia al riesgo (por ejemplo, riesgo adverso, riesgo neutral, la búsqueda de riesgo) – Planificación de Riesgo • En todo el sistema de aproximación al riesgo, incluyendo riesgos asociados con herramientas. 	<ul style="list-style-type: none"> • Discutir los comportamientos comunes que contribuyen al buen funcionamiento de un equipo [Usar] • Crear y seguir un programa para una reunión del equipo [Usar] • Identificar y justificar las funciones necesarias en un equipo de desarrollo de software [Usar] • Entender las fuentes, obstáculos y beneficios potenciales de un conflicto de equipo [Usar] • Aplicar una estrategia de resolución de conflictos en un ambiente de equipo [Usar] • Utilizar un método ad hoc para estimar el esfuerzo de desarrollo del software (ejemplo, tiempo) y comparar con el esfuerzo actual requerido [Usar] • Listar varios ejemplos de los riesgos del software [Usar] • Describir el impacto del riesgo en el ciclo de vida de desarrollo de software [Usar] • Describir las diferentes categorías de riesgo en los sistemas de software [Usar] • Demostrar a través de la colaboración de proyectos de equipo los elementos centrales de la construcción de equipos y gestión de equipos [Usar]
Lecturas : [Pre04], [Blu92], [Sch04], [WK00], [Key04], [WA02], [PS01], [Sch04], [Mon96], [Amb01], [Con00], [Oqu03]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

***** EVALUATION MISSING *****

10. BIBLIOGRAFÍA BÁSICA

- [Amb01] Vincenzo Ambriola. *Software Process Technology*. Springer, July 2001.
- [Blu92] Bruce I. Blum. *Software Engineering: A Holistic View*. 7th. Oxford University Press US, May 1992.
- [Con00] R Conradi. *Software Process Technology*. Springer, Mar. 2000.
- [Key04] Jessica Keyes. *Software Configuration Management*. CRC Press, Feb. 2004.
- [Mon96] Carlo Montangero. *Software Process Technology*. Springer, Sept. 1996.
- [Oqu03] Flavio Oquendo. *Software Process Technology*. Springer, Sept. 2003.
- [Pre04] Roger S. Pressman. *Software Engineering: A Practitioner's Approach*. 6th. McGraw-Hill, Mar. 2004.
- [PS01] John W. Priest and Jose M. Sanchez. *Product Development and Design for Manufacturing*. Marcel Dekker, Jan. 2001.
- [Sch04] Stephen R Schach. *Object-Oriented and Classical Software Engineering*. McGraw-Hill, Jan. 2004.
- [WA02] Daniel R. Windle and L. Rene Abreo. *Software Requirements Using the Unified Process*. Prentice Hall, Aug. 2002.
- [WK00] Yingxu Wang and Graham King. *Software Engineering Processes: Principles and Applications*. CRC Press, Apr. 2000.

Ministerio de Educación (MINEDU)
Sílabo 2021-I

1. CURSO

CS311. Programación Competitiva (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	4
2.2 Horas de teoría	:	2 (Semanal)
2.3 Horas de práctica	:	2 (Semanal)
2.4 Duración del periodo	:	16 semanas
2.5 Condición	:	Obligatorio
2.6 Modalidad	:	Presencial
2.7 Prerrequisitos	:	CS212. Análisis y Diseño de Algoritmos. (5 ^{to} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

La Programación Competitiva combina retos de solucionar problemas con el añadido de poder competir con otras personas. Enseña a los participantes a pensar más rápido y desarrollar habilidades para resolver problemas, que son de gran demanda en la industria. Este curso enseñará la resolución de problemas algorítmicos de manera rápida combinando la teoría de algoritmos y estructuras de datos con la práctica la solución de los problemas.

5. OBJETIVOS

- Que el alumno utilice técnicas de estructuras de datos y algoritmos complejos.
- Que el alumno aplique los conceptos aprendidos para la aplicación sobre un problema real.
- Que el alumno investigue la posibilidad de crear un nuevo algoritmo y/o técnica nueva para resolver un problema real.

6. COMPETENCIAS

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Usar**)
- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Usar**)
- a4) Aplicar técnicas eficientes de resolución de problemas computacionales. (4)
- b1) Identificar y aplicar de forma eficiente diversas estrategias algorítmicas y estructuras de datos para la solución de un problema dadas ciertas restricciones de espacio y tiempo. (1)

7. TEMAS

Unidad 1: Introducción (20)	
Competencias esperadas: a,b,h	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Introducción a la Programación competitiva • Modelo computacional • Complejidad algorítmica • Problemas sobre búsqueda y ordenamiento • Recursión y recurrencia • Estrategia divide y conquista 	<ul style="list-style-type: none"> • Reconocer y saber como usar los recursos del modelo de computación RAM (Random Access Machine). [Usar] • Determinar el tiempo y espacio de complejidad de algoritmos. [Usar] • Determinar relaciones de recurrencia para algoritmos recursivos.[Usar] • Resolver problemas de búsqueda y ordenamiento.[Usar] • Aprender a seleccionar los algoritmos adecuados para problemas de tipo divide y conquista.[Usar] • Diseñar nuevos algoritmos para la resolución de problemas.[Usar]
Lecturas : [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

Unidad 2: Estructuras de datos (20)	
Competencias esperadas: a,b,h	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Problemas sobre arrays y strings • Problemas sobre listas enlazadas • Problemas sobre pilas, colas • Problemas sobre arboles • Problemas sobre Hash tables • Problemas sobre Heaps 	<ul style="list-style-type: none"> • Reconocer las distintas estructuras de datos sus complejidades usos y restricciones. [Usar] • Identificar el tipo de estructura de datos adecuado a la resolución del problema. [Usar] • Reconocer tipos de problemas asociados a operaciones sobre estructuras de datos como búsqueda, inserción, eliminación y actualización.[Usar]
Lecturas : [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

Unidad 3: Paradigmas de diseño (20)	
Competencias esperadas: a,b,h	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Fuerza bruta • Divide y conquista • Backtracking • Greedy • Programación Dinamica 	<ul style="list-style-type: none"> • Aprender los distintos paradigmas de resolución de problemas.[Usar] • Aprender a seleccionar los algoritmos adecuados para distintos problemas según el tipo de paradigma.[Usar]
Lecturas : [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

Unidad 4: Gráfos (20)	
Competencias esperadas: a,b,h	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Recorrido de gráfos • Aplicaciones y problemas sobre gráfos • Camino mas corto • Redes y flujos 	<ul style="list-style-type: none"> • Identificar problemas clasificados como problemas de grafos. [Usar] • Aprender a seleccionar los algoritmos adecuados para problemas de grafos (recorrido, MST, camino mas costo, redes y flujos) y conocer sus soluciones eficientes. [Usar]
Lecturas : [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

Unidad 5: Tópicos avanzados (20)	
Competencias esperadas: a,b,h	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Teoria de números • Probabilidad y combinaciones • Algoritmos para manejos de strings (tries, string hashing, z-algorithm) • Geometria y sweep line algorithms, segment trees 	<ul style="list-style-type: none"> • Aprender a elegir los algoritmos adecuados para problemas sobre teoria de números y matemáticas ya que son importantes en programación competitiva. [Usar] • Aprender a seleccionar los algoritmos adecuados para problemas sobre probabilidades y combinaciones, manejos de strings y geometría computacional. [Usar]
Lecturas : [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

Unidad 6: Problemas de dominio especifico (20)	
Competencias esperadas: a,b,h	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Latencia y rendimiento • Paralelismo • Redes • Almacenamiento • Alta disponibilidad • Caching • Proxies • Equilibradores de carga • Almacenamiento clave-valor • Replicar y compartir • Elección del líder • Limitación de la tasa • Registro y monitoreo 	<ul style="list-style-type: none"> • Aprender a diseñar sistemas para diferentes problemas de dominio especifico aplicando conocimiento sobre redes, computación distribuida, alta disponibilidad, almacenamiento y arquitectura de sistemas. [Usar]
Lecturas : [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

***** EVALUATION MISSING *****

10. BIBLIOGRAFÍA BÁSICA

- [ALP12] A. Aziz, T.H. Lee, and A. Prakash. *Elements of Programming Interviews: The Insiders' Guide*. ElementsOfProgrammingInterviews.com, 2012. ISBN: 9781479274833. URL: <https://books.google.com.pe/books?id=y6FLBQAAQBAJ>.
- [Cor+09] T. H. Cormen et al. *Introduction to Algorithms*. MIT Press, 2009.
- [Hal13] Steven Halim. *Competitive Programming*. 3 rd. Lulu, 2013.
- [Kul19] Alexander S. Kulikov. *Learning Algorithms Through Programming and Puzzle Solving*. Active Learning Technologies, 2019.
- [Laa17] Antti Laaksonen. *Guide to Competitive Programming: Learning and Improving Algorithms Through Contests*. Springer, 2017.
- [Mig03] Steve Skiena Miguel A. Revilla. *Programming Challenges: The Programming Contest Training Manual*. Springer, May 2003. ISBN: 978-0387001630.

Ministerio de Educación (MINEDU)
Sílabo 2021-I

1. CURSO

CS312. Estructuras de Datos Avanzadas (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	4
2.2 Horas de teoría	:	2 (Semanal)
2.3 Horas de práctica	:	2 (Semanal)
2.4 Duración del periodo	:	16 semanas
2.5 Condición	:	Obligatorio
2.6 Modalidad	:	Presencial
2.7 Prerrequisitos	:	CS212. Análisis y Diseño de Algoritmos. (5 ^{to} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Los algoritmos y estructuras de datos son una parte fundamental de la ciencia de la computación que nos permiten organizar la información de una manera más eficiente, por lo que es importante para todo profesional del área tener una sólida formación en este aspecto.

En el curso de estructuras de datos avanzadas nuestro objetivo es que el alumno conozca y analice estructuras complejas, como los Métodos de Acceso Multidimensional, Métodos de Acceso Espacio-Temporal y Métodos de Acceso Métrico, etc.

5. OBJETIVOS

- Que el alumno entienda, diseñe, implemente, aplique y proponga estructuras de datos innovadoras para solucionar problemas relacionados al tratamiento de datos multidimensionales, recuperación de información por similitud, motores de búsqueda y otros problemas computacionales.

6. COMPETENCIAS

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Familiarizarse**)
- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Usar**)
- c) Diseñar, implementar y evaluar un sistema, proceso, componente o programa computacional para alcanzar las necesidades deseadas. (**Familiarizarse**)

■NoSpecificOutcomes■

7. TEMAS

Unidad 1: Técnicas Básicas de Implementación de Estructuras de Datos (16)	
Competencias esperadas: a,b,c	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Programación estructurada • Programación Orientada a Objetos • Tipos Abstractos de Datos • Independencia del lenguaje de programación del usuario de la estructura • Independencia de Plataforma • Control de concurrencia • Protección de Datos • Niveles de encapsulamiento (struct, class, namespace, etc) 	<ul style="list-style-type: none"> • Que el alumno entienda las diferencias básicas que involucran las distintas técnicas de implementación de estructuras de datos[Usar] • Que el alumno analice las ventajas y desventajas de cada una de las técnicas existentes[Usar]
Lecturas : [Cua+04], [Knu07a], [Knu07b], [Gam+94], [Bjö18], [Dav18]	

Unidad 2: Métodos de Acceso Multidimensionales (16)	
Competencias esperadas: a,b,c	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Métodos de Acceso para datos puntuales • Métodos de Acceso para datos no puntuales • Problemas relacionados con el aumento de dimensión 	<ul style="list-style-type: none"> • Que el alumno entienda conozca e implemente algunos Métodos de Acceso para datos multidimensionales y espacio temporales[Usar] • Que el alumno entienda el potencial de estos Métodos de Acceso en el futuro de las bases de datos comerciales[Usar]
Lecturas : [Sam06], [Gü98]	

Unidad 3: Métodos de Acceso Métrico (20)	
Competencias esperadas: a,b,c	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Métodos de Acceso Métrico para distancias discretas • Métodos de Acceso Métrico para distancias continuas 	<ul style="list-style-type: none"> • Que el alumno entienda conozca e implemente algunos métodos de acceso métrico[Usar] • Que el alumno entienda la importancia de estos Métodos de Acceso para la Recuperación de Información por Similitud[Usar]
Lecturas : [Sam06], [Chá+01], [Tra+00], [Zez+07]	

Unidad 4: Métodos de Acceso Aproximados (20)	
Competencias esperadas: a,b,c	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Space Filling Curves • Locality Sensitive Hashing 	<ul style="list-style-type: none"> • Que el alumno entienda conozca e implemente algunos métodos de acceso aproximados[Usar] • Que el alumno entienda la importancia de estos Métodos de Acceso para la Recuperación de Información por Similitud en entornos donde la Escalabilidad sea una factor muy importante[Usar]
Lecturas : [Sam06], [PI06], [Zez+07]	

Unidad 5: Seminarios (8)	
Competencias esperadas: a,b,c	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Métodos de Acceso Espacio Temporal • Estructuras de Datos con programación genérica 	<ul style="list-style-type: none"> • Que el alumno pueda discutir sobre los últimos avances en métodos de acceso para distintos dominios de conocimiento[Usar]
Lecturas : [Sam06], [Chá+01]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

***** EVALUATION MISSING *****

10. BIBLIOGRAFÍA BÁSICA

- [Bjö18] Stefan Björnander. *C++17 By Example: Practical projects to get you up and running with C++17*. Packt Publishing, Feb. 2018.
- [Chá+01] E. Chávez et al. "Proximity Searching in Metric Spaces". In: *ACM Computing Surveys* 33.3 (Sept. 2001), pp. 273–321.
- [Cua+04] Ernesto Cuadros-Vargas et al. "Implementing data structures: An incremental approach". <http://socios.spc.org.pe/ecuadros/cursos/pdfs/>. 2004.
- [Dav18] Doug Gregor David Vandevoorde Nicolai M. Josuttis. *C++ Templates: The Complete Guide*. Addison-Wesley Professional, Sept. 2018. URL: <http://informit.com/aw>.
- [Gam+94] Erich Gamma et al. *Design Patterns: Elements of Reusable Object-Oriented Software*. Computing Series. ISBN-10: 0201633612. Addison-Wesley Professional, Nov. 1994.

- [Gü98] Volker Gaede and Oliver ünther. “Multidimensional Access Methods”. In: *ACM Computing Surveys* 30.2 (1998), pp. 170–231.
- [Knu07a] Donald Ervin Knuth. *The Art of Computer Programming, Fundamental Algorithms*. 3rd. Vol. I. 0-201-89683-4. Addison-Wesley, Feb. 2007.
- [Knu07b] Donald Ervin Knuth. *The Art of Computer Programming, Sorting and Searching*. 2nd. Vol. II. 0-201-89685-0. Addison-Wesley, Feb. 2007.
- [PI06] Trevor Darrell PGregory Shakhnarovich and Piotr Indyk. *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice*. 1st. ISBN 0-262-19547-X. MIT Press, Mar. 2006.
- [Sam06] Hanan Samet. *Foundations of Multidimensional and Metric Data Structures*. Illustrated. Elsevier/Morgan Kaufmann, Aug. 2006. ISBN: 9780123694461. URL: <http://books.google.com.pe/books?id=v0-NRRKHG84C>.
- [Tra+00] C. Traina Jr et al. “Slim-Trees: High Performance Metric Trees Minimizing Overlap between Nodes”. In: *Advances in Database Technology - EDBT 2000, 6th International Conference on Extending Database Technology*. Vol. 1777. Lecture Notes in Computer Science. Konstanz, Germany: Springer, Mar. 2000, pp. 51–65.
- [Zez+07] Pavel Zezula et al. *Similarity Search: The Metric Space Approach*. 1st. ISBN-10: 0387291466. Springer, Nov. 2007.

Ministerio de Educación (MINEDU)
Sílabo 2021-I

1. CURSO

CS393. Sistemas de Información (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	: 4
2.2 Horas de teoría	: 2 (Semanal)
2.3 Horas de práctica	: 2 (Semanal)
2.4 Duración del periodo	: 16 semanas
2.5 Condición	: Obligatorio
2.6 Modalidad	: Presencial
2.7 Prerrequisitos	: CS291. Ingeniería de Software I. (5 ^{to} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Analizar técnicas para la correcta implementación de Sistemas de Información escalables, robustos, confiables y eficientes en las organizaciones.

5. OBJETIVOS

- Implementar de forma correcta (escalables, robustos, confiables y eficientes) Sistemas de Información en las organizaciones.

6. COMPETENCIAS

- c) Diseñar, implementar y evaluar un sistema, proceso, componente o programa computacional para alcanzar las necesidades deseadas. (**Usar**)
- i) Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (**Usar**)
- k) Aplicar los principios de desarrollo y diseño en la construcción de sistemas de software de complejidad variable. (**Evaluar**)
- c2) Diseñar y desarrollar sistemas de información que implementen las reglas de negocio. (**2**)
- i1) Desarrollar componentes que haciendo uso de técnicas modernas de computación implementen funcionalidad que sean de utilidad para diversos sistemas de información. (**1**)
- k1) Desempeñarse adecuadamente como parte de un proyecto de implementación de sistemas de información. (**1**)

7. TEMAS

Unidad 1: Introducción (15)	
Competencias esperadas: c,i	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Introducción a la gestión de la información• Software para gestión de información.• Tecnología para gestión de información.	<ul style="list-style-type: none">• Aplicar correctamente la tecnología para la gestión de la información [Evaluar]
Lecturas : [Som17], [PM15], [LL17]	

Unidad 2: Estrategia (15)	
Competencias esperadas: i,k	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Estrategia para gestión de información • Estrategia para gestión conocimiento • Estrategia para sistema de información. 	<ul style="list-style-type: none"> • Aplicar y evaluar correctamente estrategias de gestión [Evaluar]
Lecturas : [Som17], [PM15]	

Unidad 3: Implementación (15)	
Competencias esperadas: c,i,k	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Gestión de desarrollo de sistemas de información. • Gestión del cambio • Arquitectura de Información 	<ul style="list-style-type: none"> • Aplicar y evaluar correctamente estrategias de implementación [Evaluar]
Lecturas : [Som17], [PM15]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

***** EVALUATION MISSING *****

10. BIBLIOGRAFÍA BÁSICA

- [LL17] Kenneth C. Laudon and Jane P. Laudon. *Management Information Systems: Managing the Digital Firm*. 15th. Pearson, Mar. 2017.
- [PM15] Roger S. Pressman and Bruce Maxim. *Software Engineering: A Practitioner's Approach*. 8th. McGraw-Hill, Jan. 2015.
- [Som17] Ian Sommerville. *Software Engineering*. 10th. Pearson, Mar. 2017.

Ministerio de Educación (MINEDU)
Sílabo 2021-I

1. CURSO

CS3II. Seguridad en Computación (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	3
2.2 Horas de teoría	:	1 (Semanal)
2.3 Horas de práctica	:	4 (Semanal)
2.4 Duración del periodo	:	16 semanas
2.5 Condición	:	Obligatorio
2.6 Modalidad	:	Presencial
2.7 Prerrequisitos	:	CS231. Redes y Comunicaciones. (5 ^{to} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Hoy en día la información es uno de los activos más preciados en cualquier organización. Este curso está orientado a poder brindar al alumno los elementos de seguridad orientados a proteger la información de la organización y principalmente poder prever los posibles problemas relacionados con este rubro. Esta materia involucra el desarrollo de una actitud preventiva por parte del alumno en todas las áreas relacionadas al desarrollo de software.

5. OBJETIVOS

- Discutir a un nivel intermedio avanzado los fundamentos de la Seguridad Informática.
- Brindar los diferentes aspectos que presenta el código malicioso.
- Que el alumno conozca los conceptos de criptografía y seguridad en redes de computadoras.
- Discutir y analizar junto con el alumno los aspectos de la Seguridad en Internet.

6. COMPETENCIAS

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Usar**)
- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Evaluar**)
- e) Entender correctamente las implicancias profesionales, éticas, legales, de seguridad y sociales de la profesión. (**Usar**)
- g) Analizar el impacto local y global de la computación sobre los individuos, organizaciones y sociedad. (**Evaluar**)
- h) Incorporarse a un proceso de aprendizaje profesional continuo. (**Usar**)
- i) Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (**Usar**)

■NoSpecificOutcomes■

7. TEMAS

Unidad 1: Fundamentos y Conceptos en Seguridad (25)**Competencias esperadas: a,g**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• CIA (Confidencialidad, Integridad, Disponibilidad)• Conceptos de riesgo, amenazas, vulnerabilidades, y los tipos de ataque .• Autenticación y autorización, control de acceso (vs. obligatoria discrecional)• Concepto de la confianza y la honradez .• Ética (revelación responsable)	<ul style="list-style-type: none">• Analizar las ventajas y desventajas de equilibrar las propiedades clave de seguridad(Confidenciabilidad, Integridad, Disponibilidad) [Familiarizarse]• Describir los conceptos de riesgo, amenazas, vulnerabilidades y vectores de ataque(incluyendo el hecho de que no existe tal cosa como la seguridad perfecta) [Familiarizarse]• Explicar los conceptos de autenticación, autorización, control de acceso [Familiarizarse]• Explicar el concepto de confianza y confiabilidad [Familiarizarse]• Reconocer de que hay problemas éticos más importantes que considerar en seguridad computacional, incluyendo problemas éticos asociados a arreglar o no arreglar vulnerabilidades y revelar o no revelar vulnerabilidades [Familiarizarse]
Lecturas : [WL14]	

Unidad 2: Principios de Diseño Seguro (25)**Competencias esperadas: g,a,e,h**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Menor privilegio y aislamiento.• Valores predeterminados a prueba de fallos.• Diseño abierto.• La seguridad de extremo a extremo.• La defensa en profundidad (por ejemplo, la programación defensiva, defensa en capas)• Diseño de seguridad.• Las tensiones entre la seguridad y otros objetivos de diseño.• Mediación completa.• El uso de componentes de seguridad vetados.• Economía del mecanismo (la reducción de la base informática de confianza, minimizar la superficie de ataque)• Seguridad utilizable.• Componibilidad de seguridad.• Prevención, detección y disuasión.	<ul style="list-style-type: none">• Describir el principio de privilegios mínimos y el aislamiento que se aplican al diseño del sistema [Familiarizarse]• Resumir el principio de prueba de fallos y negar por defecto [Familiarizarse]• Discutir las implicaciones de depender de diseño abierto o secreto de diseño para la seguridad [Familiarizarse]• Explicar los objetivos de seguridad de datos de extremo a extremo [Familiarizarse]• Discutir los beneficios de tener múltiples capas de defensas [Familiarizarse]• Por cada etapa en el ciclo de vida de un producto, describir que consideraciones de seguridad deberían ser evaluadas [Familiarizarse]• Describir el costo y ventajas y desventajas asociadas con el diseño de seguridad de un producto. [Familiarizarse]• Describir el concepto de mediación y el principio de mediación completa [Familiarizarse]• Conocer los componentes estándar para las operaciones de seguridad, en lugar de reinventar las operaciones fundamentales [Familiarizarse]• Explicar el concepto de computación confiable incluyendo base informática confiable y de la superficie de ataque y el principio de minimización de base informática confiable [Familiarizarse]• Discutir la importancia de la usabilidad en el diseño de mecanismos de seguridad [Familiarizarse]• Describir problemas de seguridad que surgen en los límites entre varios componentes [Familiarizarse]• Identificar los diferentes roles de mecanismos de prevención y mecanismos de eliminación/disuasión [Familiarizarse]
Lecturas : [WL14]	

Unidad 3: Programación Defensiva (25)**Competencias esperadas: b,e,i**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Validación de datos de entrada y sanitización• Elección del lenguaje de programación y lenguajes con tipos de datos seguro.• Ejemplos de validación de entrada de datos y sanitización de errores.<ul style="list-style-type: none">– Desbordamiento de búfer– Errores enteros– Inyección SQL– Vulnerabilidad XSS• Las condiciones de carrera.• Manejo correcto de las excepciones y comportamientos inesperados.• Uso correcto de los componentes de terceros.• Desplegar eficazmente las actualizaciones de seguridad.• Información de control de flujo.• Generando correctamente el azar con fines de seguridad.• Mecanismos para la detección y mitigación de datos de entrada y errores de sanitización.• Fuzzing• El análisis estático y análisis dinámico.• Programa de verificación.• Soporte del sistema operativo (por ejemplo, la asignación al azar del espacio de direcciones, canarios)• El soporte de hardware (por ejemplo, el DEP, TPM)	<ul style="list-style-type: none">• Explicar por que la validación de entrada y desinfección de datos es necesario en el frente del control contencioso del canal de entrada [Usar]• Explicar por que uno debería escoger para desallorar un programa en un lenguaje tipo seguro como Java, en contraste con un lenguaje de programación no seguro como C/C++ [Usar]• Clasificar los errores de validación de entrada común, y escribir correctamente el código de validación de entrada [Usar]• Demostrar el uso de un lenguaje de programación de alto nivel cómo prevenir una condición de competencia que ocurran y cómo manejar una excepción [Usar]• Demostrar la identificación y el manejo elegante de las condiciones de error [Familiarizarse]• Explique los riesgos de mal uso de las interfaces con código de terceros y cómo utilizar correctamente el código de terceros [Familiarizarse]• Discutir la necesidad de actualizar el software para corregir las vulnerabilidades de seguridad y la gestión del ciclo de vida de la corrección [Familiarizarse]
Lecturas : [WL14]	

Unidad 4: Ataques y Amenazas (25)	
Competencias esperadas: b,e,i	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Atacante metas, capacidades y motivaciones (como economía sumergida, el espionaje digital, la guerra cibernética, las amenazas internas, hacktivismo, las amenazas persistentes avanzadas) • Los ejemplos de malware (por ejemplo, virus, gusanos, spyware, botnets, troyanos o rootkits) • Denegación de Servicio (DoS) y Denegación de Servicio Distribuida (DDoS) • Ingeniería social (por ejemplo, phishing) • Los ataques a la privacidad y el anonimato . • El malware / comunicaciones no deseadas, tales como canales encubiertos y esteganografía. 	<ul style="list-style-type: none"> • Describir tipos de ataques similares en contra de un sistema en particular [Familiarizarse] • Discutir los limitantes de las medidas en contra del malware (ejm. detección basada en firmas, detección de comportamiento) [Familiarizarse] • Identificar las instancias de los ataques de ingeniería social y de los ataques de negación de servicios [Familiarizarse] • Discutir como los ataques de negación de servicios puede ser identificados y reducido [Familiarizarse] • Describir los riesgos de la privacidad y del anonimato en aplicaciones comunmente usadas [Familiarizarse] • Discutir los conceptos de conversión de canales y otros procedimientos de filtrado de datos [Familiarizarse]
Lecturas : [WL14]	

Unidad 5: Seguridad de Red (25)	
Competencias esperadas: b,e,i	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Red de amenazas y tipos de ataques específicos (por ejemplo, la denegación de servicio, spoofing, olfateando y la redirección del tráfico, el hombre en el medio, ataques integridad de los mensajes, los ataques de enrutamiento, y el análisis de tráfico) • El uso de cifrado de datos y seguridad de la red . • Arquitecturas para redes seguras (por ejemplo, los canales seguros, los protocolos de enrutamiento seguro, DNS seguro, VPN, protocolos de comunicación anónimos, aislamiento) • Los mecanismos de defensa y contramedidas (por ejemplo, monitoreo de red, detección de intrusos, firewalls, suplantación de identidad y protección DoS, honeypots, seguimientos) • Seguridad para redes inalámbricas, celulares . • Otras redes no cableadas (por ejemplo, ad hoc, sensor, y redes vehiculares) • Resistencia a la censura. • Gestión de la seguridad operativa de la red (por ejemplo, control de acceso a la red configure) 	<ul style="list-style-type: none"> • Describir las diferentes categorías de amenazas y ataques en redes [Familiarizarse] • Describir las arquitecturas de criptografía de clave pública y privada y cómo las ICP brindan apoyo a la seguridad en redes [Familiarizarse] • Describir ventajas y limitaciones de las tecnologías de seguridad en cada capa de una torre de red [Familiarizarse] • Identificar los adecuados mecanismos de defensa y sus limitaciones dada una amenaza de red [Usar]
Lecturas : [WL14]	

Unidad 6: Criptografía (25)	
Competencias esperadas: b,e,i	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Terminología básica de criptografía cubriendo las nociones relacionadas con los diferentes socios (comunicación), canal seguro / inseguro, los atacantes y sus capacidades, cifrado, descifrado, llaves y sus características, firmas. • Tipos de cifrado (por ejemplo, cifrado César, cifrado affine), junto con los métodos de ataque típicas como el análisis de frecuencia. • Apoyo a la infraestructura de clave pública para la firma digital y el cifrado y sus desafíos. • Criptografía de clave simétrica: <ul style="list-style-type: none"> – El secreto perfecto y el cojín de una sola vez – Modos de funcionamiento para la seguridad semántica y encriptación autenticada (por ejemplo, cifrar-entonces-MAC, OCB, GCM) – Integridad de los mensajes (por ejemplo, CMAC, HMAC) • La criptografía de clave pública: <ul style="list-style-type: none"> – Permutación de trampa, por ejemplo, RSA – Cifrado de clave pública, por ejemplo, el cifrado RSA, cifrado El Gamal – Las firmas digitales – Infraestructura de clave pública (PKI) y certificados – Supuestos de dureza, por ejemplo, Diffie-Hellman, factoring entero • Protocolos de intercambio de claves autenticadas, por ejemplo, TLS . • Primitivas criptográficas: <ul style="list-style-type: none"> – generadores pseudo-aleatorios y cifrados de flujo – cifrados de bloque (permutaciones pseudo-aleatorios), por ejemplo, AES – funciones de pseudo-aleatorios – funciones de hash, por ejemplo, SHA2, resistencia colisión – códigos de autenticación de mensaje – funciones derivaciones clave 	<ul style="list-style-type: none"> • Describir el propósito de la Criptografía y listar formas en las cuales es usada en comunicación de datos [Familiarizarse] • Definir los siguientes términos: Cifrado, Criptoanálisis, Algoritmo Criptográfico, y Criptología y describe dos métodos básicos (cifrados) para transformar texto plano en un texto cifrado [Familiarizarse] • Discutir la importancia de los números primos en criptografía y explicar su uso en algoritmos criptográficos [Familiarizarse] • Ilustrar como medir la entropía y como generar aleatoriedad criptográfica [Usar] • Usa primitivas de clave pública y sus aplicaciones [Usar] • Explicar como los protocolos de intercambio de claves trabajan y como es que pueden fallar [Familiarizarse] • Discutir protocolos criptográficos y sus propiedades [Familiarizarse]
Lecturas : [WL14]	

Unidad 7: Seguridad en la Web (25)**Competencias esperadas: a,g**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">● Modelo de seguridad Web<ul style="list-style-type: none">– Modelo de seguridad del navegador incluida la política de mismo origen– Los límites de confianza de cliente-servidor, por ejemplo, no pueden depender de la ejecución segura en el cliente● Gestión de sesiones, la autenticación:<ul style="list-style-type: none">– Single Sign-On– HTTPS y certificados● Vulnerabilidades de las aplicaciones y defensas :<ul style="list-style-type: none">– Inyección SQL– XSS– CSRF● Seguridad del lado del cliente :<ul style="list-style-type: none">– Política de seguridad Cookies– Extensiones de seguridad HTTP, por ejemplo HSTS– Plugins, extensiones y aplicaciones web– Seguimiento de los usuarios Web● Herramientas de seguridad del lado del servidor, por ejemplo, los cortafuegos de aplicación Web (WAFS) y fuzzers	<ul style="list-style-type: none">● Describe el modelo de seguridad de los navegadores incluyendo las políticas del mismo origen y modelos de amenazas en seguridad web [Familiarizarse]● Discutir los conceptos de sesiones web, canales de comunicación seguros tales como Seguridad en la Capa de Transporte(<i>TLS</i>) y la importancia de certificados de seguridad, autenticación incluyendo inicio de sesión único, como OAuth y Lenguaje de Marcado para Confirmaciones de Seguridad(<i>SAML</i>) [Familiarizarse]● Investigar los tipos comunes de vulnerabilidades y ataques en las aplicaciones web, y defensas contra ellos [Familiarizarse]● Utilice las funciones de seguridad del lado del cliente [Usar]
Lecturas : [WL14]	

Unidad 8: Seguridad de plataformas (25)**Competencias esperadas: b,e,i**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Integridad de código y firma de código.• Arranque seguro, arranque medido, y la raíz de confianza.• Testimonio.• TPM y coprocesadores seguros.• Las amenazas de seguridad de los periféricos, por ejemplo, DMA, IOMMU.• Ataques físicos: troyanos de hardware, sondas de memoria, ataques de arranque en frío.• Seguridad de dispositivos integrados, por ejemplo, dispositivos médicos, automóviles.• Ruta confiable.	<ul style="list-style-type: none">• Explica el concepto de integridad de código y firma de códigos, así como el alcance al cual se aplica [Familiarizarse]• Discute los conceptos del origen de la confidencialidad y el de los procesos de arranque y carga segura [Familiarizarse]• Describe los mecanismos de arresto remoto de la integridad de un sistema [Familiarizarse]• Resume las metas y las primitivas claves de los modelos de plataforma confiable (TPM) [Familiarizarse]• Identifica las amenazas de conectar periféricos en un dispositivo [Familiarizarse]• Identifica ataques físicos y sus medidas de control [Familiarizarse]• Identifica ataques en plataformas con hardware que no son del tipo PC [Familiarizarse]• Discute los conceptos y la importancia de ruta confiable [Familiarizarse]
Lecturas : [WL14]	

Unidad 9: Investigación digital (Digital Forensics) (25)**Competencias esperadas: a,g**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">● Principios básicos y metodologías de análisis digital forense.● Diseñar sistemas con necesidades forenses en mente.● Reglas de Evidencia - conceptos generales y las diferencias entre las jurisdicciones y la Cadena de Custodia.● Búsqueda y captura de comprobación: requisitos legales y de procedimiento.● Métodos y normas de evidencia digital.● Las técnicas y los estándares para la conservación de los datos.● Cuestiones legales y reportes incluyendo el trabajo como perito.● Investigación digital de los sistema de archivos.● Los forenses de aplicación.● Investigación digital en la web.● Investigación digital en redes.● Investigación digital en dispositivos móviles.● Ataques al computador/red/sistema.● Detección e investigación de ataque.● Contra investigación digital.	<ul style="list-style-type: none">● Describe qué es una investigación digital, las fuentes de evidencia digital, y los límites de técnicas forenses [Familiarizarse]● Explica como diseñar software de apoyo a técnicas forenses [Familiarizarse]● Describe los requisitos legales para usar datos recuperados [Familiarizarse]● Describe el proceso de recolección de evidencia desde el tiempo en que se identifico el requisito hasta la colocación de los datos [Familiarizarse]● Describe como se realiza la recolección de datos y el adecuado almacenamiento de los datos originales y de la copia forense [Familiarizarse]● Realiza recolección de datos en un disco duro [Usar]● Describe la responsabilidad y obligación de una persona mientras testifica como un examinador forense [Familiarizarse]● Recupera datos basados en un determinado término de búsqueda en una imagen del sistema [Usar]● Reconstruye el historial de una aplicación a partir de los artefactos de la aplicación [Familiarizarse]● Reconstruye el historial de navegación web de los artefactos web [Familiarizarse]● Captura e interpreta el tráfico de red [Familiarizarse]● Discute los retos asociados con técnicas forenses de dispositivos móviles [Familiarizarse]
Lecturas : [WL14]	

Unidad 10: Seguridad en Ingeniería de Software (25)	
Competencias esperadas: a,g,i,c	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • La construcción de la seguridad en el ciclo de vida de desarrollo de software. • Principios y patrones de diseño seguros. • Especificaciones de software seguros y requisitos. • Prácticas de desarrollo de software de seguros. • Asegure probar el proceso de las pruebas de que se cumplan los requisitos de seguridad (incluyendo análisis estático y dinámico) 	<ul style="list-style-type: none"> • Describir los requisitos para la integración de la seguridad en el SDL [Familiarizarse] • Aplicar los conceptos de los principios de diseño para mecanismos de protección, los principios para seguridad de software (Viega and McGraw) y los principios de diseño de seguridad (Morrie Gasser) en un proyecto de desarrollo de software [Familiarizarse] • Desarrollar especificaciones para un esfuerzo de desarrollo de software que especifica completamente los requisitos funcionales y se identifican las rutas de ejecución esperadas [Familiarizarse]
Lecturas : [WL14]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

***** EVALUATION MISSING *****

10. BIBLIOGRAFÍA BÁSICA

[WL14] Stallings. W and Brown. L. *Computer Security: Principles and Practice*. Pearson Education, Limited, 2014. ISBN: 9780133773927.

Ministerio de Educación (MINEDU)
Sílabo 2021-I

1. CURSO

CS3P1. Computación Paralela y Distribuída (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	4
2.2 Horas de teoría	:	2 (Semanal)
2.3 Horas de práctica	:	4 (Semanal)
2.4 Duración del periodo	:	16 semanas
2.5 Condición	:	Obligatorio
2.6 Modalidad	:	Presencial
2.7 Prerrequisitos	:	<ul style="list-style-type: none">• CS212. Análisis y Diseño de Algoritmos. (5^{to} Sem)• CS231. Redes y Comunicaciones. (5^{to} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

La última década ha traído un crecimiento explosivo en computación con multiprocesadores, incluyendo los procesadores de varios núcleos y centros de datos distribuidos. Como resultado, la computación paralela y distribuida se ha convertido de ser un tema ampliamente electivo para ser uno de los principales componentes en la malla estudios en ciencia de la computación de pregrado. Tanto la computación paralela como la distribuida implica la ejecución simultánea de múltiples procesos, cuyas operaciones tienen el potencial para intercalarse de manera compleja. La computación paralela y distribuida construye sobre cimientos en muchas áreas, incluyendo la comprensión de los conceptos fundamentales de los sistemas, tales como: concurrencia y ejecución en paralelo, consistencia en el estado/manipulación de la memoria, y latencia. La comunicación y la coordinación entre los procesos tiene sus cimientos en el paso de mensajes y modelos de memoria compartida de la computación y conceptos algorítmicos como atomicidad, el consenso y espera condicional. El logro de aceleración en la práctica requiere una comprensión de algoritmos paralelos, estrategias para la descomposición problema, arquitectura de sistemas, estrategias de implementación y análisis de rendimiento. Los sistemas distribuidos destacan los problemas de la seguridad y tolerancia a fallos, hacen hincapié en el mantenimiento del estado replicado e introducen problemas adicionales en el campo de las redes de computadoras.

5. OBJETIVOS

- Que el alumno sea capaz de crear aplicaciones paralelas de mediana complejidad aprovechando eficientemente máquinas con múltiples núcleos.
- Que el alumno sea capaz de comparar aplicaciones secuenciales y paralelas.
- Que el alumno sea capaz de convertir, cuando la situación lo amerite, aplicaciones secuenciales a paralelas de forma eficiente.

6. COMPETENCIAS

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Usar**)
- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Usar**)
- a5) Aplicar técnicas eficientes de resolución de problemas computacionales en ambientes paralelos y distribuidos. (**5**)
- b2) Identificar y aplicar de forma eficiente diversas estrategias algorítmicas y estructuras de datos para la solución de un problema en ambientes paralelos y distribuidos. (**2**)

7. TEMAS

Unidad 1: Fundamentos de paralelismo (18)**Competencias esperadas: a**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Procesamiento Simultáneo Múltiple.• Metas del Paralelismo (ej. rendimiento) frente a Concurrencia (ej. control de acceso a recursos compartidos)• Paralelismo, comunicación, y coordinación:<ul style="list-style-type: none">– Paralelismo, comunicación, y coordinación– Necesidad de Sincronización• Errores de Programación ausentes en programación secuencial:<ul style="list-style-type: none">– Tipos de Datos (lectura/escritura simultánea o escritura/escritura compartida)– Tipos de Nivel más alto (interleavings violating program intention, no determinismo no deseado)– Falta de vida/progreso (deadlock, starvation)	<ul style="list-style-type: none">• Distinguir el uso de recursos computacionales para una respuesta más rápida para administrar el acceso eficiente a un recurso compartido [Familiarizarse]• Distinguir múltiples estructuras de programación suficientes para la sincronización que pueden ser interimplementables pero tienen ventajas complementarias [Familiarizarse]• Distinguir datos de carrera (<i>data races</i>) a partir de carreras de más alto nivel [Familiarizarse]
Lecturas : [Pac11], [Mat14], [quinnz], [Geo10]	

Unidad 2: Arquitecturas paralelas (12)	
Competencias esperadas: a,b	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Procesadores mutlinúcleo. • Memoria compartida vs memoria distribuida. • Multiprocesamiento simétrico. • SIMD, procesamiento de vectores. • GPU, coprocesamiento. • Taxonomía de Flynn. • Soporte a nivel de instrucciones para programación paralela. <ul style="list-style-type: none"> – Instrucciones atómicas como Compare/Set (Comparar / Establecer) • Problemas de Memoria: <ul style="list-style-type: none"> – Caches multiprocesador y coherencia de cache – Acceso a Memoria no uniforme (NUMA) • Topologías. <ul style="list-style-type: none"> – Interconexiones – Clusters – Compartir recursos (p.e., buses e interconexiones) 	<ul style="list-style-type: none"> • Explicar las diferencias entre memoria distribuida y memoria compartida [Evaluar] • Describir la arquitectura SMP y observar sus principales características [Evaluar] • Distinguir los tipos de tareas que son adecuadas para máquinas SIMD [Usar] • Describir las ventajas y limitaciones de GPUs vs CPUs [Usar] • Explicar las características de cada clasificación en la taxonomía de Flynn [Usar] • Describir los desafíos para mantener la coherencia de la caché [Familiarizarse] • Describir los desafíos clave del desempeño en diferentes memorias y topologías de sistemas distribuidos [Familiarizarse]
Lecturas : [Pac11], [KH13], [SK10], [Geo10]	

Unidad 3: Descomposición en paralelo (18)	
Competencias esperadas: a,b	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Necesidad de Comunicación y coordinación/sincronización. • Independencia y Particionamiento. • Conocimiento Básico del Concepto de Descomposición Paralela. • Descomposición basada en tareas: <ul style="list-style-type: none"> – Implementación de estrategias como hebras • Descomposición de Información Paralela <ul style="list-style-type: none"> – Estrategias como SIMD y MapReduce • Actores y Procesos Reactivos (solicitud de gestores) 	<ul style="list-style-type: none"> • Explicar por qué la sincronización es necesaria en un programa paralelo específico [Usar] • Identificar oportunidades para particionar un programa serial en módulos paralelos independientes [Familiarizarse] • Escribir un algoritmo paralelo correcto y escalable [Usar] • Paralelizar un algoritmo mediante la aplicación de descomposición basada en tareas [Usar] • Paralelizar un algoritmo mediante la aplicación de descomposición datos en paralelo [Usar] • Escribir un programa usando actores y/o procesos reactivos [Usar]
Lecturas : [Pac11], [Mat14], [Qui03], [Geo10]	

Unidad 4: Comunicación y coordinación (18)**Competencias esperadas: a,b**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Memoria Compartida.• La consistencia, y su papel en los lenguaje de programación garantías para los programas de carrera libre.• Pasos de Mensaje:<ul style="list-style-type: none">– Mensajes Punto a Punto versus multicast (o basados en eventos)– Estilos para enviar y recibir mensajes Blocking vs non-blocking– Buffering de mensajes• Atomicidad:<ul style="list-style-type: none">– Especificar y probar atomicidad y requerimientos de seguridad– Granularidad de accesos atómicos y actualizaciones, y uso de estructuras como secciones críticas o transacciones para describirlas– Exclusión mutua usando bloques, semáforos, monitores o estructuras relacionadas<ul style="list-style-type: none">* Potencial para fallas y bloqueos (<i>deadlock</i>) (causas, condiciones, prevención)– Composición<ul style="list-style-type: none">* Componiendo acciones atómicas granulares más grandes usando sincronización* Transacciones, incluyendo enfoques optimistas y conservadores• Consensos:<ul style="list-style-type: none">– (Cíclicos) barreras, contadores y estructuras relacionadas• Acciones condicionales:<ul style="list-style-type: none">– Espera condicional (p.e., empleando variables de condición)	<ul style="list-style-type: none">• Usar exclusión mutua para evitar una condición de carrera [Usar]• Dar un ejemplo de una ordenación de accesos entre actividades concurrentes (por ejemplo, un programa con condición de carrera) que no son secuencialmente consistentes [Familiarizarse]• Dar un ejemplo de un escenario en el que el bloqueo de mensajes enviados pueden dar <i>deadlock</i> [Usar]• Explicar cuándo y por qué mensajes de multidifusión (<i>multicast</i>) o basado en eventos puede ser preferible a otras alternativas [Familiarizarse]• Escribir un programa que termine correctamente cuando todo el conjunto de procesos concurrentes hayan sido completados [Usar]• Dar un ejemplo de un escenario en el que un intento optimista de actualización puede nunca completarse [Familiarizarse]• Usar semaforos o variables de condición para bloquear hebras hasta una necesaria precondition de mantenga [Usar]
Lecturas : [Pac11], [Mat14], [Qui03], [Geo10]	

Unidad 5: Análisis y programación de algoritmos paralelos (18)**Competencias esperadas: a,b**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Caminos críticos, el trabajo y la duración y la relación con la ley de Amdahl.• Aceleración y escalabilidad.• Naturalmente (vergonzosamente) algoritmos paralelos.• Patrones Algoritmicos paralelos (divide-y-conquista, map/reduce, amos-trabajadores, otros)<ul style="list-style-type: none">– Algoritmos específicos (p.e., MergeSort paralelo)• Algoritmos de grafos paralelo (por ejemplo, la ruta más corta en paralelo, árbol de expansión paralela)• Cálculos de matriz paralelas.• Productor-consumidor y algoritmos paralelos segmentados.• Ejemplos de algoritmos paralelos no-escalables.	<ul style="list-style-type: none">• Definir: camino crítico, trabajo y <i>span</i> [Familiarizarse]• Calcular el trabajo y el <i>span</i> y determinar el camino crítico con respecto a un diagrama de ejecución paralela. [Usar]• Definir <i>speed-up</i> y explicar la noción de escalabilidad de un algoritmo en este sentido [Familiarizarse]• Identificar tareas independientes en un programa que debe ser paralelizado [Usar]• Representar características de una carga de trabajo que permita o evite que sea naturalmente paralelizable [Familiarizarse]• Implementar un algoritmo dividir y conquistar paralelo (y/o algoritmo de un grafo) y medir empíricamente su desempeño relativo a su analogo secuencial [Usar]• Descomponer un problema (por ejemplo, contar el número de ocurrencias de una palabra en un documento) via operaciones <i>map</i> y <i>reduce</i> [Usar]• Proporcionar un ejemplo de un problema que se corresponda con el paradigma productor-consumidor [Usar]• Dar ejemplos de problemas donde el uso de <i>pipelining</i> sería un medio eficaz para la paralelización [Usar]• Implementar un algoritmo de matriz paralela [Usar]• Identificar los problemas que surgen en los algoritmos del tipo productor-consumidor y los mecanismos que pueden utilizarse para superar dichos problemas [Usar]
Lecturas : [Mat14], [Qui03], [Geo10]	

Unidad 6: Desempeño en paralelo (18)	
Competencias esperadas: a,b,c	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Equilibrio de carga. • La medición del desempeño. • Programación y contención. • Evaluación de la comunicación de arriba. • Gestión de datos: <ul style="list-style-type: none"> – Costos de comunicación no uniforme debidos a proximidad – Efectos de Cache (p.e., false sharing) – Manteniendo localidad espacial • Consumo de energía y gestión. 	<ul style="list-style-type: none"> • Detectar y corregir un desbalanceo de carga [Usar] • Calcular las implicaciones de la ley de Amdahl para un algoritmo paralelo particular [Usar] • Describir como la distribución/disposición de datos puede afectar a los costos de comunicación de un algoritmo [Familiarizarse] • Detectar y corregir una instancia de uso compartido falso (<i>false sharing</i>) [Usar] • Explicar el impacto de la planificación en el desempeño paralelo [Familiarizarse] • Explicar el impacto en el desempeño de la localidad de datos [Familiarizarse] • Explicar el impacto y los puntos de equilibrio relacionados al uso de energía en el desempeño paralelo [Familiarizarse]
Lecturas : [Pac11], [Mat14], [KH13], [SK10], [Geo10]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

***** EVALUATION MISSING *****

10. BIBLIOGRAFÍA BÁSICA

- [Geo10] Gerhard Wellein Georg Hager. *Introduction to High Performance Computing for Scientists and Engineers (Chapman & Hall/CRC Computational Science)*. Ed. by CRC Press. 1st. 2010. ISBN: 978-1439811924.
- [KH13] David B. Kirk and Wen-mei W. Hwu. *Programming Massively Parallel Processors: A Hands-on Approach*. 2nd. Morgan Kaufmann, 2013. ISBN: 978-0-12-415992-1.
- [Mat14] Norm Matloff. *Programming on Parallel Machines*. University of California, Davis, 2014. URL: <http://heather.cs.ucdavis.edu/~matloff/158/PLN/ParProcBook.pdf>.
- [Pac11] Peter S. Pacheco. *An Introduction to Parallel Programming*. 1st. Morgan Kaufmann, 2011. ISBN: 978-0-12-374260-5.
- [Qui03] Michael J. Quinn. *Parallel Programming in C with MPI and OpenMP*. 1st. McGraw-Hill Education Group, 2003. ISBN: 0071232656.

[SK10] Jason Sanders and Edward Kandrot. *CUDA by Example: An Introduction to General-Purpose GPU Programming*. 1st. Addison-Wesley Professional, 2010. ISBN: 0131387685, 9780131387683.