



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
ESCUELA NACIONAL COLEGIO DE CIENCIAS Y HUMANIDADES

PROGRAMAS DE ESTUDIO 2024

ÁREA DE MATEMÁTICAS

CIBERNÉTICA Y COMPUTACIÓN I-II

Primera edición: julio de 2024.

D.R. © UNAM 2024, Universidad Nacional Autónoma de México,
Ciudad Universitaria. Alcaldía Coyoacán, C.P. 04510, CDMX.

Esta edición y sus características son propiedad de la UNAM.
Prohibida la reproducción total o parcial por cualquier medio, sin
la autorización escrita del titular de los derechos patrimoniales.
Impreso y hecho en México - *Printed in Mexico*.

ÍNDICE

PRESENTACIÓN DE LA MATERIA	7
Relaciones con el Área y otras asignaturas	8
Enfoque disciplinario y didáctico	9
Contribución al Perfil del Egresado	15
Evaluación	16
Concreción en la materia de los principios del Modelo Educativo del Colegio: <i>aprender a aprender, aprender a hacer, aprender a ser</i>	17
Propósitos generales de la materia	18
Panorama general de las unidades	20

CIBERNÉTICA Y COMPUTACIÓN I

Presentación de la asignatura de Cibernética y Computación I	23
Unidad 1. La cibernética	24
Presentación de la unidad	24
Carta descriptiva	24
Evaluación	28
Referencias	30

Unidad 2. Representación de información en sistema binario y álgebra de Boole	32
Presentación de la unidad	32
Carta descriptiva	32
Evaluación	40
Referencias	41
Unidad 3. Metodología de solución de problemas e Introducción a la programación en Java	43
Presentación de la unidad	43
Carta descriptiva	43
Evaluación	51
Referencias	52

CIBERNÉTICA Y COMPUTACIÓN II

Presentación de la asignatura de Cibernética y Computación II	57
Unidad 1. Fundamentos de programación orientada a objetos con Java	58
Presentación de la unidad	58
Carta descriptiva	58
Evaluación	61
Referencias.....	62
Unidad 2. Estructuras de control de secuencia en Java	64
Presentación de la unidad	64
Carta descriptiva	64
Evaluación	70
Referencias	72

Unidad 3. Herencia y polimorfismo	74
Presentación de la unidad	74
Carta descriptiva	74
Evaluación	78
Referencias	80
Unidad 4. Interfaz Gráfica de Usuario	83
Presentación de la unidad	83
Carta descriptiva	83
Evaluación	88
Referencias	90

PRESENTACIÓN DE LA MATERIA

Una de las características distintivas de la Escuela Nacional Colegio de Ciencias y Humanidades (ENCCH) de otros bachilleratos, que la hacen innovadora y con una pedagogía vigente en México y América Latina, es su Modelo Educativo, el cual es de cultura básica, propedéutico (esto es, preparará al alumnado para ingresar a la licenciatura con los conocimientos necesarios para su vida profesional), y está orientado a la formación intelectual ética y social, para que el alumnado sea considerado sujeto de la cultura y responsable de su propia educación. Esto significa que en la institución la enseñanza fomentará en el alumnado conocimientos, habilidades y actitudes necesarias para que ellos generen pensamientos crítico y computacional para asumir decisiones de forma consiente y responsable.

Desde su origen, el CCH adoptó los principios de una educación moderna donde consideraba al alumnado capaz de apropiarse del conocimiento y sus aplicaciones. En este sentido, el trabajo del profesorado del Colegio consiste en dotarlos de los instrumentos teóricos y metodológicos necesarios para poseer los principios de una cultura científica-humanística.

El proceso enseñanza-aprendizaje es inseparable, sin embargo, la enseñanza gira en torno del alumnado y el profesorado se convierte en un guía que acompaña en la construcción de su conocimiento. La didáctica aplicada persigue que el alumnado *aprenda a aprender*, que la actividad receptiva y creadora sea continua y que adquiera capacidad auto formativa para lograr el conocimiento auténtico y la formación de actitudes, intercambiando experiencias con sus compañeros de forma colaborativa en diferentes espacios académicos en su beneficio.

El profesorado, basado en sus conocimientos, experiencia, habilidad, y didáctica, es el mediador, transmisor de los conocimientos y un acompañante responsable del alumnado, al que propone retos de aprendizaje para que ellos adquieran nuevos conocimientos, sean conscientes de su hacer y proceder de forma reflexiva al aplicarlo en su cotidianidad.

Al ser un aprendizaje dinámico, el alumnado desarrollará un aprendizaje activo, respetuoso, pensando en la comunidad, en la sustentabilidad, equidad e igualdad dentro y fuera del espacio académico, para la realización de trabajos de investigación y prácticas de laboratorios.

Las asignaturas de Cibernética y Computación I y II pertenecen al Área de Matemáticas, se cursan en dos semestres (quinto y sexto), son obligatorias de elección (optativas) de acuerdo con el área de la profesión y preferencia que el alumnado elige en cuarto semestre.

El programa contempla tres unidades para Cibernética y Computación I y cuatro para la asignatura de Cibernética y Computación II, con 64 horas por semestre, el tiempo asignado por semana corresponde a dos sesiones de dos horas.

Los aprendizajes de cada unidad están graduados para que el alumnado los adquiera, iniciando con los temas sencillos hasta llegar a los de mayor complejidad, con la finalidad de que lo aprendido en una unidad sirva de base para su utilización en las subsecuentes, mediante la siguiente estructura:

Relaciones con el Área y otras asignaturas

La computadora surge del ingenio de grandes matemáticos, Blaise Pascal y la pascalina, Gottfried Wilhelm von Leibnitz, quienes realizaron el primer dispositivo para hacer operaciones aritméticas, de aquí la relación con las matemáticas.

En la actualidad, las computadoras hacen miles de operaciones por segundo y, por tal razón, se ha convertido en una herramienta utilizada en la mayoría de profesiones, lo que vuelve a la computación en una ciencia transdisciplinaria, aunado al método didáctico aprendizaje basado en problemas o proyectos (ABP), presentando al alumnado distintas situaciones de la vida real a los que deben buscar soluciones y así obtener conocimientos, habilidades y actitudes.

A su vez, crean vínculos con distintas áreas del conocimiento y, en específico, con materias como las Matemáticas. Dentro de la asignatura se requiere del conocimiento de los sistemas de numeración y operaciones aritméticas con números binarios y teoría de conjuntos para relacionarla con el álgebra de Boole; por otro lado, el argumento deductivo de la lógica, en filosofía del *modus ponens*, de la física las conexiones eléctricas para el armado de los circuitos lógicos. En cualquier área del conocimiento el alumnado debe responder a la solución de problemas, para tal logro, aprenderá el uso de los algoritmos para encontrar las soluciones.

Continuando con otras asignaturas, en Física, Biología y Química el alumnado estudia los temas de sistemas naturales y artificiales, planteando soluciones sustentables y trabajando colaborativamente. La asignatura tiene estrecha relación con las TIC y Taller de Cómputo. El alumnado podrá usar un procesador de texto y una hoja de cálculo.

Con respecto a la asignatura de Taller de Lectura, Redacción e Iniciación a la Investigación Documental, el alumnado podrá comprender artículos de divulgación científica y realizar ensayos académicos o proyectos de investigación.

Es importante resaltar el uso de lenguajes naturales y formales, la aplicación de la semántica, el léxico y la sintaxis en el contexto de los lenguajes de programación; son elementos relacionados con el Área de Talleres de Lenguaje y Comunicación y con Inglés, que se abordan de manera más específica para la construcción de oraciones o sentencias en un lenguaje formal (reglas más estrictas que uno natural).

Por tanto, esta asignatura permite al alumnado reflexionar sobre lo que está aprendiendo en otros campos del saber y que pueda ir retomando aplicaciones de otras asignaturas para resolverlas mediante el empleo de la computadora, que sistematizará la solución con rapidez y eficiencia dentro de las asignaturas de Cibernética y Computación I y II.

Enfoque disciplinario y didáctico

En este apartado se dará una aproximación específica para abordar la materia dentro del proceso educativo. El Programa de Estudios de Cibernética y Computación I y II busca dar al alumnado una visión general de la cibernética y la computación desde un enfoque STEAM; busca de manera integral darle al alumnado el conocimiento y preparación relacionado con los avances tecnológicos, poniendo énfasis en los pilares de la ciencia, la tecnología, la ingeniería, el arte y las matemáticas, aunado al ABP que le permita desarrollar propuestas de solución a problemas; que el alumnado construya el conocimiento acompañado del profesorado de manera activa donde él tome el rol que le permita investigar para conocer el tema, discuta y argumente para obtener el análisis que derive en los pasos de solución, trabajando de forma comprometida con el cambio de pensamientos, formas de hacer y ser, todo orientado al bienestar común, equilibrado y en armonía con sus congéneres y medio ambiente.

Enfoque disciplinario

El enfoque transdisciplinario permite a la disciplina incidir tanto en las ciencias experimentales y sociales, en el dominio de las matemáticas y la tecnología buscando sentar las bases para que el alumnado pueda dar solución a la gran alteridad de situaciones que le rodean, unificado a lo que menciona la *Gaceta Amarilla*: “la materia debe proporcionar métodos, técnicas y el hábito de aplicarlos en problemas concretos y de adquirir nuevos conocimientos” (CCH, 1971, p.4).

En esta construcción de conocimientos de conceptos, metodologías y valores desde las ciencias de la Cibernética y la Computación, el alumnado confirmará los cimientos de la asignatura de Taller de Cómputo, equiparados a los conocimientos, habilidades y actitudes, señalados dentro de los primeros siete niveles de la matriz de habilidades digitales (DGTIC, 2014), como: el acceso a la información, seguridad informática, uso y manejo de ofimática y medios digitales, así como ambientes virtuales de aprendizaje.

Mientras, los contenidos temáticos de Cibernética y Computación I y II corresponden al octavo nivel de la matriz de habilidades en TIC, en el momento en que el alumnado aprenderá a analizar, diseñar e implementar una programación informática, a usar simuladores e interpretar sus resultados, incluyendo al álgebra de Boole que en Cibernética y Computación I le permitirá a diseñar circuitos lógicos.

Estas habilidades digitales, de nivel ocho, proporcionan al alumnado la capacidad de desarrollar programas informáticos, para lo cual necesitan comprender el algoritmo y diagrama de flujo como herramientas que permiten describir paso a paso la solución a problemas; incluso, en la obtención de programas informáticos con soluciones optimizadas sustentablemente, por ejemplo, en el impacto de energía en el medio ambiente.

Fundamentos teóricos sólidos: el alumnado, mediante el estudio introductorio de la Cibernética y los Sistemas, comprende la importancia de los conocimientos teóricos que sustentan el desarrollo tecnológico para centrar su atención en la solución de problemas computacionales. Por ello, se revisan los antecedentes de la cibernética y su concepto, los principales personajes que han aportado a su desarrollo hasta la actualidad, los elementos que conforman un sistema, y la conveniencia de su estudio mediante los modelos.

Metodología: desarrollar una metodología para resolver problemas, lo que permite al alumnado ampliar las habilidades analíticas, sintéticas, de abstracción, creativas, reflexivas, y fomenta un pensamiento lógico, crítico que les permita afrontar desafíos dentro del aula y fuera, ampliando su capacidad para resolver problemas cotidianos.

Además de la algoritmia, conocerá los principios básicos de la programación orientada a objetos que le permita integrar elementos de programación utilizando la abstracción, el acceso controlado a la información y la reutilización de código, como parte de las habilidades esenciales del siglo XXI.

Con el desarrollo de habilidades para la convivencia profesional, aunadas a las habilidades cognitiva y técnica, se busca fomentar el desarrollo de habilidades de comunicación, trabajo en equipo, colaboración y oralidad, preparando al alumnado para que continúe con éxito su preparación académica o profesional; además, que cuente con la habilidad para comunicarse de forma asertiva con equipos de desarrollo de software, con usuarios o administradores de proyecto y así, construir conocimiento desde la indagación, transformación y divulgación del conocimiento construido.

La materia se fundamenta en los siguientes conceptos básicos, los cuales se agrupan en unidades de estudio:

a. Cibernética y Computación I.

- **La cibernética.** En esta unidad se estudia la definición, antecedentes y personajes que contribuyeron a la formalización de la cibernética, el concepto de sistema, sus tipos y elementos, enfatizando la importancia del modelado de los sistemas para su estudio, ya que son conceptos asociados a cualquier rama del conocimiento humano se vuelve transdisciplinario. Además, se busca formar conciencia en la responsabilidad de mejorar a

través de la sustentabilidad el ambiente físico, psicológico y social, poniendo énfasis en la equidad e igualdad.

- **Representación de información en sistema binario y álgebra de Boole.** Se usan los sistemas de numeración binario, octal y hexadecimal, ya que se reconocen a éstos como la base para comprender el manejo y la representación de información en la memoria de la computadora. El alumnado explora los conceptos de tipos de datos, rango, precisión, uso de números enteros con y sin signo, variables booleanas, tipos alfanuméricos; en general tipos de datos primitivos y su codificación en bytes. Posteriormente, se estudia el álgebra de Boole, tablas de verdad, diagramas de compuertas lógicas para el diseño y armado de circuitos lógicos, sentando así, un precedente en el manejo de operadores lógicos que se utilizan para la toma de decisiones en estructuras de control en los lenguajes de programación de alto nivel, que se aprenderá en la siguiente unidad.
- **Metodología de solución de problemas e introducción a la programación en Java.** El alumnado adquiere una metodología que le permita aplicar el análisis, el razonamiento en el desarrollo de algoritmos, el uso de diagramas de flujo y pseudocódigos, para apoyar la codificación en el lenguaje de programación Java sistematizando necesidades planteadas, donde el alumnado propone y construye la solución de problemas de temas encontrados en algunas de las asignaturas que cursa (transdisciplina), verificando que los resultados sean correctos. Además, se busca que la forma de trabajo promueva actitudes de convivencia que contemplen la equidad y la igualdad.

b. Cibernética y Computación II.

- **Fundamentos de programación orientada a objetos con Java.** Reconocimiento de las características del paradigma de la Programación Orientada a Objetos (POO) y las del lenguaje de programación Java, implementando la abstracción y encapsulamiento de clases, con sus atributos, métodos e interfaces para poder crear objetos y así obtener programas sencillos con los principios básicos de la POO.

Se retoma lo aprendido en la **Unidad 3 Cibernética** y Computación I y se profundiza en los principios de la programación: abstracción y encapsulamiento, elementos de una clase, instancia de objetos, constructores, uso de la consola para pruebas y depuración; en otras palabras, la formalización de los elementos del lenguaje de programación Java.

También, inclusión del Lenguaje de Modelado Unificado (UML) de las herramientas de la ingeniería de software que combinan los aspectos del proceso de desarrollo (como fases definidas, técnicas, y prácticas) con otros componentes

de desarrollo (como documentos, modelos, manuales, código fuente, etcétera), llamado Proceso Racional Unificado (RUP). Se identifican las fases de desarrollo de programas desde el punto de vista de modelado utilizando algunos diagramas de UML que permitan la separación de elementos de lógica y de presentación en el diseño de programas. Se modelan diversas clases bajo los principios de la POO, que se irán perfeccionando conforme avancen los contenidos de las siguientes unidades.

- **Estructuras de control de secuencia en Java.** Aplicación de la programación orientada a objetos en la solución de problemas en los que se tomen decisiones simples, combinadas y múltiples; el alumnado utiliza estas estructuras para incorporar en sus clases la protección de los atributos mediante la validación previa a su asignación. Implementa programas que den solución a problemas que involucren estructuras de control iterativas, donde aplica las estructuras de datos: arreglos unidimensionales, bidimensionales, ArrayList para el manejo de arreglos dinámicos, la estructura foreach (para el recorrido de colecciones), operador ternario, operadores de incremento (++, --) y acumulación (+, -=, *=, /=, ...).
- **Herencia y polimorfismo.** Se profundiza en los conceptos de POO como herencia y polimorfismo, interacción y comunicación entre clases como temas que refuerzan las características de la programación orientada a objetos con Java. Permite entender la reutilización y especialización de código y las jerarquías de clases que se encuentran en un lenguaje de programación orientado a objetos, identificando elementos comunes que necesitan modificarse o adicionarse a una clase. Además, el alumnado aprende a integrar elementos de software desarrollados por otros programadores en proyectos propios y a generar elementos que pueden ser utilizados al administrar proyectos en equipos trabajando de forma colaborativa.
- **Interfaz Gráfica de Usuario.** Desarrollo de objetos del paquete Javax-Swing para crear la interfaz gráfica que permitan mostrar la solución a problemas que engloben los aprendizajes adquiridos. Deben considerarse los enfoques basados en problemas o en proyectos.

Enfoque didáctico

Este Programa de Estudios de las asignaturas de Cibernética y Computación I y II tiene el propósito didáctico centrado en la enseñanza-aprendizaje constructivista, para que el profesorado planee el proceso de enseñanza y guíe el aprendizaje del alumnado.

La propuesta educativa busca promover acciones didácticas para formar en el alumnado una autogestión del conocimiento, la cual genera el desarrollo de

hábitos que le permiten apropiarse de estrategias cognitivas y metacognitivas, para favorecer el dominio de las habilidades de pensamiento de orden superior.

El alumnado es el centro del proceso educativo, por lo que se busca su aprendizaje activo mediante el Aprendizaje Basado en Problemas o Proyectos, aplicando una metodología basada en planteamientos de problemas reales como punto de partida y donde el alumnado toma un rol activo participando en aplicar la teoría a veces buscada por él, otras veces por el profesorado, trabajando de forma individual, grupal o colaborativa y fomentando su responsabilidad durante este proceso, cuidando de contar con espacios áulicos donde exista respeto, seguridad y empatía, así, el alumnado se sienta motivado de comentar sus dudas o sugerir ideas dentro del aula.

El proceso de enseñanza aprendizaje de las asignaturas debe orientarse para que el alumnado logre los aprendizajes y propósitos de la materia a través del desarrollo de los contenidos temáticos. Por la naturaleza de la materia, el uso de las Tecnologías de la Información y la Comunicación ha estado presente así, sin nombrarla TIC, por lo que se sugiere utilizar estrategias didácticas que incluyan el uso ético de herramientas para la búsqueda de información, comunicación y colaboración en línea, seguridad informática, plataformas educativas, simuladores, almacenamiento en la nube e inteligencia artificial; con el fin de que el alumnado adquiera, mejores prácticas digitales y desarrolle habilidades tecnológicas aplicadas a la solución de necesidades para todos los ámbitos de su vida, de manera sustentable, transdisciplinaria con equidad e igualdad considerando ese derecho en la interacción donde todos sin importar situación social, género, cultura o diversidad convivamos para buscar un bien para el medio ambiente y la comunidad.

Al integrar conocimiento, métodos y valores se busca generar en el alumnado un pensamiento crítico, reflexivo y computacional, para que partiendo del análisis de un sistema construya el algoritmo de solución que representa el proceso o funcionamiento de éste, el cual se puede llevar a la computadora convertido en una aplicación o programa informático, a través de un lenguaje de programación (Java).

El profesorado debe tener la consigna de enfocar los problemas o proyectos en situaciones reales, proporcionando un contexto motivador que interese al alumnado a lo largo de su desarrollo. Esta orientación asegura que el conocimiento adquirido sea aplicable en un contexto de su formación como cultura general, lo que garantiza su durabilidad y utilidad a largo plazo.

Los temas de sustentabilidad, desarrollo sostenible del medio en el que vivimos, equidad y perspectiva de género en la educación, son elementos importantes a incluir durante el aprendizaje, siempre buscando de manera conjunta, la toma de conciencia del daño que se está generando en nuestro ambiente a nivel físico, económico y social, por lo que es imprescindible fomentar la conciencia social en el alumnado, ya que sus acciones tendrán un gran impacto en su futuro y el de otros.

De ahí que es deseable contar con la formación integral del alumnado, cuidando el equilibrio de los conocimientos, su aplicación y las actitudes a través del trabajo en equipo de forma colaborativa, practicando la tolerancia, la solidaridad y la responsabilidad, la comunicación efectiva y afectiva, la toma de decisiones, la honestidad y creatividad para la solución a problemas a través de alternativas menos centradas en los datos de conocimiento y más ajustadas a los desafíos culturales y contextuales del alumnado. De igual forma, generando oportunidades para incluirlos, creando ambientes de aprendizaje que sean de utilidad para su diversidad en pensamientos, costumbres, hábitos, etcétera. Para lograr este tipo de trabajo, se debe de considerar por igual a los participantes, sin importar su condición social, ideología, raza o preferencia sexual.

Por lo anterior, el Programa de Estudios está diseñado para que el alumnado logre los aprendizajes para obtener los propósitos y el profesorado planea cómo explicar los contenidos temáticos, a través de actividades que le permitan al mismo alumnado apropiarse de esos conocimientos y aplicarlos. Para esta integración, el profesorado debe considerar secuencias didácticas en sesiones de dos horas con la siguiente estructura: Apertura, momento didáctico que corresponde a la bienvenida y presentación del encuadre. Si es el inicio del semestre incluirá la posibilidad de acordar las reglas de convivencia dentro del aula, la forma de organizar las actividades y de evaluarlas, el profesorado realizará una evaluación diagnóstica para valorar los conocimientos previos del alumnado; asimismo, el profesorado provee al alumnado de las observaciones en su avance y del apoyo si es que no lo obtiene con respecto a la norma.

Para las sesiones siguientes, la apertura tendrá la introducción al tema del día y/o retroalimentación a partir de una pregunta detonadora o el planteamiento de un problema para dar introducción al desarrollo. Este momento didáctico permite abordar los contenidos mediante los materiales o recursos didácticos para la apropiación y aplicación del conocimiento, por medio de la exposición, lecturas, indagación, análisis, reflexión, diálogo grupal, consenso de opiniones u otras técnicas con el empleo de las herramientas tecnológicas.

Para el momento didáctico que corresponde al cierre, la consolidación de los contenidos abordados, al refuerzo del o los aprendizajes, se hace un resumen de lo realizado y a la revisión del aprendizaje, es decir, verificar si se logró o no por el alumnado.

Dicha valoración se logra a partir de los distintos tipos de evaluación: diagnóstica, formativa y sumativa, donde el rol del evaluador puede ser: el mismo alumnado, entre pares, o el profesorado en conjunto con el alumnado. En este mismo sentido, se sugiere que las secuencias didácticas tengan un enfoque de conocimiento gradual, es decir, es importante que en cada sesión se construya sobre el conocimiento previo, para que el alumnado tenga una comprensión sólida y progresiva.

Como apoyo para la indagación de la información que se requiere para encontrar definiciones, conceptos teóricos y de la forma de implementar el código para la resolución de problemas o proyectos, el uso de fuentes de consulta por parte del alumnado es indispensable, porque le ayuda a desarrollar su capacidad de autoaprendizaje además de aprender no sólo en su estancia en el Colegio, sino en toda su vida académica, sin embargo, la guía tanto como la supervisión del profesorado es fundamental.

Al final de cada unidad se sugiere bibliografía separada en básica y complementaria, su importancia radica en el hecho que tanto alumnado y profesorado inviertan menos tiempo para localizar la información relacionada con las temáticas a estudiar, ya que las fuentes de consulta incluyen varios temas del programa, se pueden repetir en distintas unidades.

Aunado a lo anterior, es conveniente que el profesorado considere al aprendizaje colaborativo en el desarrollo de sus estrategias, ya que fomenta el trabajo en equipo, la colaboración, el intercambio de ideas, la retroalimentación entre pares, el desarrollo de habilidades comunicativas y el fortalecimiento de valores y actitudes.

Además, el trabajo colaborativo permite la consolidación de la identidad de un grupo, genera espacios de aprendizaje, fomenta la organización diferenciada para compartir, reflexionar y construir conocimientos, con ello, se invita a que el alumnado sume esfuerzos para conseguir un mismo objetivo en comunidad y para la comunidad.

Contribución al Perfil del Egresado

Las asignaturas de Cibernética y Computación I y II propician sustancialmente en el Perfil del Egresado una formación integral de conocimientos, habilidades y actitudes para su desarrollo cognitivo, mejora en sus destrezas en el análisis, diseño y solución a necesidades planteadas para su aprendizaje basado en problemas y el fomento de valores que motivan a actitudes encaminadas a un bienestar y convivencia en equidad.

En cuanto a los conocimientos, giran en torno a conceptos como la cibernética, sistemas, modelos, sistemas de numeración (decimal, binario, octal y hexadecimal) para que reconozca la importancia del carácter digital de una computadora, así como la forma de almacenar y procesar la información en ella. Se continúa con el aprendizaje de una metodología para que describa paso a paso las posibles soluciones a problemas y cree un conjunto de instrucciones que entenderá la computadora para que integre datos de entradas y, a su vez, realice ciertas operaciones para obtener resultados, usando el lenguaje de programación Java, por lo que aprenderá su sintaxis, semántica, operadores, tipos de datos, estructuras de control, definidas en clases, y pueda hacer uso de los objetos en entornos de trabajo en modo consola y gráfico.

Fomenta el pensamiento analítico y científico para la toma de decisiones que le permitan desarrollar propuestas de solución a sistemas que resuelven necesidades transdisciplinarias de forma sustentable a través de algoritmos, simuladores y el lenguaje de programación Java desde dispositivos de cómputo.

Al hacer uso de las TIC, estas propician tanto habilidades como métodos para obtener de forma segura y adecuada información desde el internet para que el alumnado logre comunicarla por redes sociales, mensajería o plataformas educativas de forma responsable, aunado a la habilidad de manejar las TIC en un nivel mayor al de usuario, que le dé la posibilidad de entender como está elaborado el software que utiliza, dotándolo de las herramientas para desarrollar uno a su propia medida y necesidad para cualquier área de conocimiento, ya sea como herramientas para sus actividades académicas, entretenimiento, interacción social, entre otras.

Las habilidades actitudinales buscan en el perfil del alumnado que sea consciente y responsable de su diálogo, pensamiento y acción, a través de la comunicación asertiva para relacionarse con su entorno, desde la libertad, empatía y congruencia, prevaleciendo la equidad, igualdad, trabajo en y para la comunidad, pensando en un bienestar común y sustentable.

Evaluación

La evaluación es un proceso continuo desde inicio a fin de cada asignatura, cuya finalidad es determinar en qué medida se lograron los propósitos y aprendizajes marcados en el Programa de Estudios.

En ella se valora la evolución cognitiva del alumnado, observando el avance en cada una de las actividades que realiza donde conjuntamente alumnado y profesorado van construyendo los conocimientos necesarios para poderlos aplicar, además de considerar las actitudes y valores dentro del salón de clases.

Se plantean tres momentos de evaluaciones: diagnóstica, formativa y sumativa con la posibilidad de utilizar los tipos de evaluación: autoevaluación, coevaluación y/o heteroevaluación.

Evaluación diagnóstica

Es la que se realiza al inicio del curso para recoger datos personales y académicos en la situación de partida. Su propósito es que el profesorado inicie el proceso educativo con un conocimiento real de las características del alumnado, lo que permitirá diseñar estrategias didácticas y acoplar su práctica docente a la realidad del grupo y de sus singularidades individuales.

Esta evaluación diagnóstica se puede contemplar al inicio de las unidades temáticas que requieran la concretización del conocimiento previo. Estas herramientas pueden ser un cuestionario escrito o simplemente en plantear un tema de discusión y dejar que el alumnado opine, a través del análisis de las respuestas dadas, e incluso la elaboración de trabajos extra clase le permitirá al

profesorado darse cuenta de las características personales en lo cognitivo, socioeconómico, actitudinal y valores del alumnado, para ser considerados como factores que influyen en su rendimiento académico.

Evaluación formativa

Es una valoración continua y sistemática que busca reorientar el aprendizaje, a través de la observación, con comentarios mencionados al alumnado sobre las actividades o tareas realizadas, puede ser en forma grupal, en pares, o individual, durante el curso, para mejorar los procesos educativos.

Con respecto a la evaluación formativa, la recomendación es permitir que el alumnado recopile información en forma individual o en equipos y que la discutan primero en sus equipos y después en plenaria, con la finalidad de que sea el mismo alumnado el que obtenga conclusiones de forma consensuada y enriquecida por todos y se dé cuenta de lo aprendido.

Es claro que el alumnado puede llevar a cabo las prácticas, proyectos e investigaciones con éxito siempre y cuando el profesorado le dé lineamientos detallados y que en cada momento lo esté asesorando. Para ello existen varias herramientas, como: rúbricas, lista de verificación, reportes de realización de proyectos, participación individual, portafolio de evidencias. El profesorado tendrá el cuidado de incluir los comentarios del trabajo, al desarrollar las tareas en equipo, valorando el respeto, organización, responsabilidad, contribución y comportamiento del alumnado.

Evaluación sumativa

Consiste en la recolección y ponderación de los datos obtenidos de las dos evaluaciones anteriores al finalizar el curso escolar. Sirve para contrastar los resultados iniciales con respecto a los propósitos y aprendizajes obtenidos durante y al final del curso, permitiendo reflexionar sobre las estrategias didácticas, la forma de transmitir las y de la apropiación del alumnado para ajustar y mejorar su enseñanza. Puede ser el punto de partida o de la evaluación diagnóstica del siguiente periodo escolar.

Concreción en la materia de los principios del Modelo Educativo del Colegio: *aprender a aprender, aprender a hacer, aprender a ser*

Aprender a aprender

Las asignaturas Cibernética y Computación I y II permiten que el alumnado investigue la temática a estudiar a través de la identificación y selección de fuentes de información confiables, las cuales analizará y empleará en forma eficaz, para adquirir habilidades que le permitan construir nuevos conocimientos que lo motive al autoaprendizaje de forma responsable.

Aprender a hacer

Durante el desarrollo de ambas asignaturas, el alumnado asimila conocimientos y desarrolla habilidades procedimentales mediante el ABP, desde un abordaje transdisciplinar, analiza el planteamiento del problema y observa las relaciones para obtener las posibles soluciones, y así descubrir la solución colaborativamente.

Con lo aprendido, el alumnado podrá implementar con las herramientas de desarrollo de software aplicaciones concretas en proyectos interdisciplinarios, integrando en su hacer la equidad, trabajo en y para la comunidad, pensando en un bienestar común y sustentable.

Aprender a ser

Durante el proceso de formación se fomenta el trabajo en y para la comunidad, la pertenencia e identidad, promoviendo el autoconocimiento de sus habilidades, destrezas, aptitudes y actitudes, a través de valores que le permitan ser una persona capaz de escuchar, dialogar de forma reflexiva, expresar sus experiencias e ideas asertivamente, propiciando en todo momento la empatía, tolerancia, respeto a él, los demás y a su medio, responsablemente, con libertad, justicia, honestidad; estas actitudes le permitirán al alumnado una formación integral en su pensar, hacer y ser para emplear los conocimientos y habilidades aprendidos, procurando la sustentabilidad, equidad e igualdad.

Propósitos generales de la materia

Al finalizar, el alumnado:

- Aplicará los conceptos de la cibernética a lo largo del curso para comprender y mejorar los sistemas mediante la tecnología, de forma que se pueda aumentar el rendimiento y eficiencia de los procesos al menor costo económico y ecológico posible.
- Reconocerá a la transdisciplina como la forma de relación entre los elementos de los diferentes sistemas (sociales, económicos, naturales, etcétera) en cualquier área del conocimiento.
- Durante el desarrollo de ambas asignaturas, el alumnado asimilará conocimientos y desarrollará habilidades procedimentales, mediante el ABP, desde un abordaje transdisciplinar, analizará el planteamiento del problema y observará las relaciones para obtener las posibles soluciones, y así descubrir la solución colaborativamente.
- Utilizará algunos elementos de los sistemas de numeración y el álgebra de Boole para resolver problemas a través de la construcción de circuitos lógicos y los representará utilizando un simulador o una *protoboard*, que posteriormente se aplicará en la construcción de expresiones para la toma de decisiones en un lenguaje de programación de alto nivel.

- Obtendrá una metodología con el apoyo de los algoritmos, diagramas de flujo, pseudocódigo y el lenguaje de programación Java para la solución de problemas a través de la computadora.
- Conocerá las características de la programación orientada a objetos, así como las del lenguaje de programación Java y su entorno de desarrollo para la solución de problemas de diferentes áreas del conocimiento.
- Utilizará las estructuras de control de secuencia: incondicionales, condicionales, de ciclo y arreglos a través del uso del lenguaje de programación orientado a objetos para potenciar las estrategias en la solución del sistema a realizar.
- Implementará características de la programación orientada a objetos, como la herencia y el polimorfismo, la comunicación entre clases para la especialización y reutilización del código.
- Realizará interfaces gráficas reutilizando los objetos del paquete Swing, retomando la importancia de los métodos *set* y *get*, así como la comunicación entre ellos para reforzar los aprendizajes adquiridos durante el curso.

A continuación, se presentan las cartas descriptivas por unidades de cada asignatura y el tiempo requerido. Las horas asignadas a cada unidad dependen de la cantidad de los aprendizajes y contenidos que se deben alcanzar, a su grado de dificultad y su ponderación.

Panorama general de las unidades

		Cibernética y Computación I	Cibernética y Computación II
Unidad 1		14 hrs.	10 hrs.
	La cibernética		Fundamentos de programación orientada a objetos con Java.
Unidad 2		18 hrs.	24 hrs.
	Representación de información en sistema binario y álgebra de Boole.		Estructuras de control de secuencia en Java.
Unidad 3		32 hrs.	14 hrs.
	Metodología de solución de problemas e introducción a la programación en Java.		Herencia y poliformismo.
Unidad 4			16 hrs.
			Interfaz gráfica y usuario.
Total		64 hrs.	64 hrs.

The background features a light gray gradient with several overlapping geometric shapes. A large white circle is centered behind the text. To its left, a gray circle overlaps it. In the bottom-left corner, there is a complex pattern of overlapping squares, some containing smaller circles, and horizontal lines. A dark gray triangle is positioned in the top-right corner.

Cibernética y Computación I

PRESENTACIÓN DE LA ASIGNATURA DE CIBERNÉTICA Y COMPUTACIÓN I

En la *Unidad 1. La cibernética*, se realiza un abordaje histórico subrayando el trabajo de las y los autores que definieron a la cibernética como una ciencia que estudia a los mecanismos de control y de comunicación, tanto en los sistemas naturales como en los sistemas artificiales; se resalta el uso de las Tecnologías de la Información y la comunicación (TIC), transdisciplina, sustentabilidad, equidad e igualdad.

En la *Unidad 2. Representación de información en sistemas binario y álgebra de Boole*, se ve la importancia de los sistemas de numeración binario, octal y hexadecimal, ya que son la base de la representación de información en una computadora, mediante el uso y operación matemática de bits. Posteriormente, se destaca el álgebra de Boole para entender la lógica de los circuitos eléctrico-electrónicos que conforman a nuestros dispositivos digitales, y con base en ello, el alumnado pueda desarrollar un proyecto.

En la *Unidad 3. Metodología de solución de problemas e Introducción a la programación en Java*, el alumnado adquiere una metodología que le permite resolver problemas analizando su planteamiento, proponiendo la soluciones a través de algoritmos, diagramas de flujo, pseudocódigos y su codificación en el lenguaje de programación Java.

UNIDAD 1. LA CIBERNÉTICA

Presentación de la unidad

En esta unidad, como introducción a las asignaturas de Cibernética y Computación I y II, pretende que el alumnado se adentre a través de los trabajos de investigación de distintos autores, así como a la historia de los grandes descubrimientos alrededor del control y comunicación de los sistemas, para identificar los elementos que lo conforman y que le permitirán analizarlos mediante su modelamiento para dar paso a la era de las computadoras y que aún en nuestros días sigue en constante innovación, resaltando que tanto la cibernética como la computación interactúan con todas las disciplinas, lo que las hace transdisciplinarias y les permite predecir y tomar decisiones de manera sustentable.

Carta descriptiva

Propósito	Tiempo
<p>Al finalizar la unidad, el alumnado:</p> <p>Modelará un sistema relacionado con un tema de alguna disciplina de su interés, analizando el concepto de cibernética para interrelacionarlo con otras ciencias y los elementos que conforman un sistema, aplicando los principios de sustentabilidad.</p>	14 hrs.

Aprendizajes	Temática	Estrategias sugeridas
<p>El alumnado:</p> <p>Identifica a la cibernética como una ciencia transdisciplinaria.</p>	<p>Definición del concepto de cibernética.</p> <p>Antecedentes de la cibernética.</p> <p>Aplicaciones de la cibernética en la actualidad y la transdisciplina.</p>	<p>Apertura</p> <p>Alumnado: en equipo, investigan los temas relativos a la cibernética, concepto, antecedentes históricos, la relación y aplicación interdisciplinaria.</p> <p>Desarrollo</p> <p>Profesorado: presenta al alumnado materiales didácticos donde se muestren aplicaciones de la cibernética; por ejemplo, la creación y diseño de prótesis, robótica, procesos productivos y administrativos, entre otros. Propicia un intercambio de ideas de forma grupal. El grupo expone sus puntos de vista relativo al material didáctico observado y los conceptos investigados.</p> <p>Cierre</p> <p>Alumnado: contesta un cuestionario de los conceptos relativos a la cibernética y sus aplicaciones.</p> <p>Extra clase</p> <p>Alumnado: busca un video sobre la cibernética y sus aplicaciones y escribe una reseña.</p>

Aprendizajes	Temática	Estrategias sugeridas
<p>El alumnado:</p> <p>Describe el trabajo científico sobre la cibernética desde distintos autores y autoras.</p>	<p>Obra de distintos autores en trabajos científicos sobre la cibernética:</p> <ul style="list-style-type: none"> ▪ Norbert Wiener. ▪ Arturo Rosenblueth. ▪ Claude Shannon. ▪ Ada Lovelace. ▪ Jean Jennings Bartik. 	<p>Apertura</p> <p>Alumnado: en equipo, investigan las aportaciones para el desarrollo de la Cibernética de Norbert Wiener, Arturo Rosenblueth y Claude Shannon, Ada Lovelace y Jean Jennings Bartik, entre otros.</p> <p>Desarrollo</p> <p>Profesorado: propicia un intercambio de ideas, guiando a que el alumnado exponga sus puntos de vista relativo a las aportaciones a la cibernética de los autores investigados.</p> <p>Cierre</p> <p>Alumnado: elabora una línea de tiempo con fechas, autores y aportaciones.</p> <p>Extra clase</p> <p>Alumnado: investiga el concepto, clasificación y ejemplos de sistema.</p>
<p>Describe los componentes de un sistema.</p>	<p>Sistemas. Concepto. Elementos. Ambiente.</p> <ul style="list-style-type: none"> • Clasificación. • Naturales. • Artificiales. • Abiertos. • Cerrados. 	<p>Apertura</p> <p>Alumnado: organizado en equipo, expone el tema relativo a los sistemas: concepto, elementos, ambiente y clasificación.</p> <p>Desarrollo</p> <p>Alumnado: elabora un mapa conceptual sobre la clasificación de los sistemas.</p> <p>Cierre</p> <p>Profesorado: a manera de lluvia de ideas, de forma grupal unifica conceptos y obtiene conclusiones en conjunto con el alumnado.</p> <p>Extra clase</p> <p>Alumnado: investiga y hace un reporte sobre las características de un sistema de control, con lazo abierto, lazo cerrado y retroalimentación..</p>

Aprendizajes	Temática	Estrategias sugeridas
<p>El alumnado:</p> <p>Explica el funcionamiento de los sistemas y el control de estos.</p>	<p>Funcionamiento del control un sistema:</p> <ul style="list-style-type: none"> ▪ Lazo abierto. ▪ Lazo cerrado. ▪ Retroalimentación. 	<p>Apertura</p> <p>Profesorado: guía al grupo para que expongan los reportes elaborados con los conceptos de lazo abierto, lazo cerrado, retroalimentación.</p> <p>Desarrollo</p> <p>Profesorado: cita algunos ejemplos para que el alumnado identifique si son de lazo abierto, cerrado y de este cuál es su retroalimentación, por ejemplo:</p> <ul style="list-style-type: none"> • Un foco eléctrico que ilumina, pero no mide los lúmenes. • Una lavadora automática que funciona por ciclos. • Secador de manos eléctrico el cual no mide la temperatura. • Tostadora de pan temporizada. • Un foco de la calle que mide los lúmenes para ser encendido o apagado. • El volumen de un equipo de sonido. • Un flotador que indica el llenado de un tinaco y apaga la bomba. • El grifo de agua de la cocina. • El mando a distancia de la TV. • Aire acondicionado ajustando automáticamente • temperatura según la temperatura ambiente. <p>Cierre</p> <p>Alumnado: en plenaria obtiene conclusiones y unifican criterios sobre los conceptos.</p> <p>Extra clase</p> <p>Alumnado: en equipo, investigan sobre el concepto de sistemas socio ecológicos, así como de riesgos o crisis socio-ambientales que estén ocurriendo en México y en el mundo. Además, el concepto de desarrollo sustentable.</p>

Aprendizajes	Temática	Estrategias sugeridas
<p>El alumnado:</p> <p>Asocia el concepto de sustentabilidad dentro de los sistemas.</p>	<p>Sustentabilidad</p> <ul style="list-style-type: none"> ▪ Concepto. ▪ Ejemplos de sistemas socio ambientales. ▪ Medio ambiente. ▪ Sistemas de Producción. ▪ Bienestar Social. ▪ Consumo responsable. ▪ Crisis socio ambiental y prácticas encaminadas a mitigarlas. 	<p>Inicio</p> <p>Profesorado: expone el siguiente caso. La Unión Europea sufrió la peor ola de calor desde el renacimiento, en el verano del 2022, con un incremento sustancial en la cantidad de muertes, por dicha razón. Solo Alemania registró 3000 muertes adicionales durante la semana del 18 de julio, en comparación de los últimos 5 años; en promedio, de 6 a 8 millones de personas han muerto a nivel global debido a la contaminación.</p> <p>Lo anterior muestra claramente que necesitamos sustentabilidad ambiental, no solo para mejorar nuestra salud, sino para eliminar una amenaza en nuestras vidas.</p> <p>Desarrollo</p> <p>Profesorado: propicia un intercambio de ideas en donde el alumnado expone su punto de vista relativo al caso, ahora se les pide que realicen una investigación documental en binas y la información encontrada la guarden en un archivo de texto. Entre los puntos fundamentales a buscar son el origen de estas crisis socio ambiental, consecuencias a largo plazo y la forma de mitigarlas.</p> <p>Cierre</p> <p>Alumnado: compara las ideas expuestas sin información, con respecto a las encontradas después de la investigación, para analizar las posibles soluciones sustentables a largo plazo para disminuir el calentamiento global y disminuir sus efectos, lo que aportará a la toma de conciencia sobre los problemas socio ecológicos. Debe hacerse énfasis en el uso racional de los recursos naturales y en las formas de lograr una sociedad equitativa.</p> <p>Extra clase</p> <p>Alumnado: investiga que es un modelo y sus tipos.</p>

Aprendizajes	Temática	Estrategias sugeridas
<p>El alumnado:</p> <p>Identifica el concepto y la importancia del modelo.</p>	<p>Modelos</p> <ul style="list-style-type: none"> ▪ Concepto. ▪ Tipos. ▪ Naturales y artificiales. ▪ Analógicos y digitales. ▪ Matemáticos. ▪ Conceptuales. ▪ Relación. <p>Elementos para modelar un sistema:</p> <ul style="list-style-type: none"> • Entrada y salida. • Proceso. 	<p>Apertura</p> <p>Alumnado: en equipo discuten la información encontrada sobre el tema relativo a modelos concepto, tipos y relaciones.</p> <p>Desarrollo</p> <p>Profesorado: pide que entren a la página del portal académico y de forma individual contesten los ejercicios y la práctica final de la siguiente liga: https://e1.portalacademico.cch.unam.mx/alumno/cibernetica1/unidad1/laCibernetica/modelos</p> <p>Cierre</p> <p>El alumnado: expresa sus comentarios o dudas correspondientes al tema.</p> <p>Extra clase</p> <p>Profesorado: les pide que, en equipo, realicen el modelo de un sistema, por medio de una representación gráfica, (presentación en diapositivas, infografía, etcétera), propuesto por el profesorado o por iniciativa propia. El ejemplo guía puede ser: el funcionamiento del horno de microondas http://www.unamiradaalaciencia.unam.mx/download/pdf_prensa/unamirada_258.pdf</p>
<p>Explica las características de un modelo del sistema, las partes que lo conforman y su funcionamiento.</p>	<p>Desarrolla un modelo de un sistema.</p>	<p>Apertura</p> <p>Profesorado: recapitula los conceptos vistos en la sesión anterior para dar apertura a las exposiciones del alumnado.</p> <p>Desarrollo</p> <p>Alumnado: expone modelos elegidos, enfatizando los elementos del sistema y los tipos en los que se clasificarían.</p> <p>Cierre</p> <p>Alumnado: describe la importancia de saber sobre la Cibernética, los sistemas y los modelos.</p>

Evaluación

En las estrategias sugeridas para cada sesión se proponen actividades que brindan evidencias del nivel del logro de los aprendizajes planteados. La evaluación de cada sesión se integra en la evaluación sumativa. Para evaluar los productos obtenidos en cada sesión se sugieren los siguientes tipos e instrumentos:

Diagnóstica

Conocer las características del grupo para trazar la ruta de implementación.

- Preguntas formuladas al inicio de la unidad y de cada clase
- Análisis de las respuestas dadas en los trabajos extra clase.
- Cuestionarios o actividades interactivas del portal académico.

Formativa

Evaluación para la reorientación para el logro de los aprendizajes.

- Pruebas de ejecución: codificación del programa.
- Observación: de las actividades realizadas en el salón de clases.
- Cuestionarios: para verificar el seguimiento o aprovechamiento del alumnado.

Sumativa

Análisis de los resultados obtenidos para determinar si se alcanzaron los niveles de aprendizajes previstos en el programa de estudios, que permitan asignar una calificación.

- Listas de verificación: para evaluar los trabajos extra clase.
- Rúbrica: para evaluar las investigaciones o exposiciones realizadas.
- Exámenes.

Actitudinal (AR)

- Valoración de las actitudes y valores desarrollados por el alumnado a lo largo del desarrollo de los aprendizajes de la unidad.
- Empatía: el alumnado comprende las necesidades y perspectivas de las personas con quién interactúa.
- Respeto a la diversidad: el alumnado reconoce y valora la diversidad de género, cultura, habilidades y opiniones de los demás.
- Colaboración y trabajo en equipo: el alumnado aprendió a comunicarse de manera efectiva, resolver conflictos y contribuir al éxito del equipo.
- Creatividad e innovación: el alumnado desarrolló soluciones originales a los problemas propuestos por el profesorado y muestra una predisposición positiva a la innovación.
- Tolerancia a la frustración: el alumnado demuestra una actitud positiva al enfrentarse a desafíos técnicos. Aprende de los errores y sigue mejorando.

Nota: la ponderación de cada actividad será asignada por el profesorado.

Referencias

Para el alumnado

Básicas

- Ávila, M. del S., *et al.* (2018). *Paquete didáctico para la asignatura de Cibernética y Computación I*. cch-oriente.unam.mx/pdf/materiales/paqueteCyC.pdf
- Bravo Medina, E. (1974). *Cibernética en la Ingeniería Química* (Tesis para obtener el título de Ingeniero Químico). UNAM, Facultad de Química, <http://132.248.9.195/pmig2017/0165709/0165709.pdf>
- CCH Portal Académico. (2017). *Cibernética y computación I*. <https://e1.portalacademico.cch.unam.mx/alumno/cibernetica1/unidad1/la-Cibernetica/introduccion>
- Maturana, H. y von Foerster, H. (*s/a*). *Historia de la Cibernética*. (<https://www.asc-cybernetics.org/foundations/timeline.htm>)
- TV UNAM. (2023). *Historia gloriosa y casi secreta de la cibernética en México, con Adrián Santuario*. https://youtu.be/rftoHABQQP4?si=zsidi8-nvC-jy_Nv

Complementarias

- Castaños Alés, E. (2000). *Norbert Wiener y el origen de la Cibernética*. Capítulo 2. Los orígenes del arte cibernético en España: el seminario de Generación Automática de Formas Plásticas del Centro de Cálculo de la Universidad de Madrid: (1968-1973). (Tesis de Doctorado). Biblioteca Virtual Miguel de Cervantes, España. http://www.infoamerica.org/documentos_pdf/wiener2.pdf
- Infoamérica. (2013). *Claude Elwood Shannon (1916-2001)*. <http://www.infoamerica.org/teoria/shannon1.htm>
- López Lira Nava, R. (1983). *La aplicación de la cibernética y la utilización de la computadora en las relaciones internacionales* (Tesis para obtener el título de Licenciado en Relaciones Internacionales). UNAM, Facultad de Ciencias Políticas y Sociales
- Quintanilla, S. (2002). "Arturo Rosenblueth y Norbert Wiener: Dos científicos en la historiografía de la educación contemporánea". *Revista Mexicana de Investigación Educativa*, 7(15), 303–329. https://www.infoamerica.org/documentos_pdf/wiener1.pdf
- Wiener, N. (1988). *Cibernética y sociedad*. Editorial Sudamericana.

Para el profesorado

Básicas

- Castaños Alés, E. (2000). *Norbert Wiener y el origen de la Cibernética. Capítulo 2. Los orígenes del arte cibernético en España: el seminario de Generación Automática de Formas Plásticas del Centro de Cálculo de la Universidad de Madrid: (1968-1973)*. (Tesis de Doctorado). Biblioteca Virtual Miguel de Cervantes http://www.infoamerica.org/documentos_pdf/wiener2.pdf
- Distefano, J. et al. (1975). *Retroalimentación y sistemas de control*. McGraw-Hill.
- Gifreu, A. (2014). *Pioneros de la tecnología digital. Ideas visionarias del mundo tecnológico actual*. UOC.
- Jramoi, V. et al. (1971). *Introducción e historia de la cibernética*. Grijalbo.
- Rosenblueth, A. (1981). *Mente y cerebro. Una filosofía de la ciencia*. México. Siglo XXI.
- Wiener, N. (1988). *Cibernética y sociedad*. Editorial Sudamericana.

Complementarias

- Wiener, N. (1967). *Dios y Golem. Comentario sobre ciertos puntos en que chocan cibernética y religión*. México. Siglo XXI.
- Wiener, N. (1985). *Cibernética o el control y comunicación en animales y máquinas*. Tusquets Editores
- Ogata, K. (1980). *Ingeniería de control moderna*. Prentice Hall.
- Raymound, R. (1984). *La cibernética y el origen de la información*. FCE.

UNIDAD 2. REPRESENTACIÓN DE INFORMACIÓN EN SISTEMA BINARIO Y ÁLGEBRA DE BOOLE

Presentación de la unidad

En esta unidad se busca que el alumnado comprenda cuál es lenguaje interno de la computadora, que todo lo que realiza son operaciones o cálculos en nano-segundos y que esos datos están representados por bits, la unidad mínima de información, que son almacenados en espacios de memoria con direcciones en hexadecimal y hasta este momento la arquitectura de la computadora sigue manufacturándose a través de circuitos lógicos diseñados con la teoría del álgebra de Boole. En otras palabras, aprenderá a sistematizar problemas de diferentes disciplinas mediante circuitos lógicos.

Carta descriptiva

Propósito	Tiempo
<p>Al finalizar la unidad, el alumnado:</p> <p>Realizará operaciones aritméticas básicas y conversiones de base entre los sistemas de numeración: binario, octal y hexadecimal para comprender la representación de la información en la memoria de una computadora y en conjunto con el álgebra de Boole para entender la lógica combinacional y diseñar circuitos lógicos.</p>	18 hrs.

Aprendizajes	Temática	Estrategias sugeridas
<p>El alumnado:</p> <p>Comprende las características de los sistemas de numeración decimal y binario, así como sus equivalencias en octal y hexadecimal, para poder representar información en la computadora como tipos de datos primitivos en un lenguaje de programación.</p>	<p>Sistemas de numeración:</p> <ul style="list-style-type: none"> ▪ Binario, octal, decimal y hexadecimal. ▪ Conversiones numéricas entre los sistemas: binario, octal, decimal y hexadecimal. ▪ Representaciones octal y hexadecimal como simplificaciones del sistema binario. 	<p>Apertura</p> <p>Profesorado: inicia el tema con preguntas reflexivas o la descripción de hechos interesantes asociados con los sistemas de numeración.</p> <p>Explica la importancia de los sistemas de numeración en la vida cotidiana y en la historia de las matemáticas, así como su aplicación en el campo de la computación.</p> <p>Muestra algunos videos para comparar los sistemas de numeración acerca de cuáles son posicionales y la importancia del uso del cero y símbolos únicos para los dígitos del sistema, con apoyo de la liga: https://portalacademico.cch.unam.mx/cibernetica1/sistemas-de-numeracion/sistema-de-numeracion-posicional.</p> <p>Desarrollo</p> <p>Alumnado: en equipos o plenaria, realiza una tabla indicando cuáles sistemas de numeración se consideran posicionales y cuáles son sus características.</p> <p>Realiza conversiones de enteros sin signo desde decimal a binario, octal y hexadecimal, y viceversa.</p> <p>Cierre</p> <p>Alumnado y profesorado: ubican los sistema decimal y binario como sistemas de numeración posicionales.</p> <p>Extra clase</p> <p>Profesorado: deja una serie de conversiones para reafirmar el aprendizaje.</p>

Aprendizajes	Temática	Estrategias sugeridas
	<ul style="list-style-type: none"> • Tipos de datos primitivos: enteros con y sin signo, números de punto flotante, caracteres y booleanos. • Tipos de datos con representaciones en memoria, rango y precisión. 	<p>Apertura Alumnado y profesorado: realizan la comprobación de las conversiones de la actividad extra clase previa.</p> <p>Desarrollo Profesorado: pregunta qué es un número entero, un número racional, un número irracional. El alumnado debe investigar en internet si no saben las respuestas. Alumnado: reflexionan si el número Pi se puede representar con todos sus decimales en las estructuras de almacenamiento de la computadora. Menciona ejemplos de relaciones como $3 < 4$ y determinar qué valor lógico resulta, para que entienda la necesidad de asignarle un valor de verdadero o falso en distintas comparaciones numéricas. Investiga la representación de un dato tipo entero, punto flotante, carácter y booleano, en la memoria. Realiza una revisión de las conversiones entre decimal y binario: sin signo, ubica que las operaciones se limitan a un espacio en memoria (8, 16, 32 o 64 bits). https://portalacademico.cch.unam.mx/cibernetica1/sistemas-de-numeracion</p> <p>Cierre Profesorado: propicia un intercambio de ideas, guiando a que el alumnado argumente: ¿cómo una cantidad fija de bits se almacena en la memoria?, ¿cuál es rango de números enteros que se puede representar? y ¿qué precisión se puede tener para números con punto flotante? Alumnado: de manera grupal realizan un cuadro con la clasificación de los tipos de datos primitivos y su almacenamiento en memoria.</p> <p>Extra clase Alumnado: investiga el método para realizar las operaciones de suma, resta, multiplicación y división en sistema de numeración binario.</p>
Realiza operaciones aritméticas con el sistema de numeración binario.	<p>Aritmética del sistema de numeración binario.</p> <p>Operaciones de adición, sustracción, multiplicación y división.</p>	<p>Apertura Profesorado: explica el método para realizar cada operación aritmética con el sistema de numeración binario y el alumnado compara el método encontrado.</p> <p>Desarrollo Alumnado: a manera de competencia pasan al pizarrón para probar su aprendizaje, verificando que todos hayan asimilado el procedimiento.</p> <p>Cierre Alumnado y profesorado: reflexionan sobre las similitudes que tiene respecto a su aritmética en el sistema de numeración binario y el sistema de numeración decimal.</p> <p>Extra clase Alumnado: realizar ejercicios de repaso.</p>

Aprendizajes	Temática	Estrategias sugeridas
	<p>Operaciones aritméticas en octal y hexadecimal como simplificaciones del sistema binario.</p>	<p>Apertura Alumnado y profesorado: revisan los ejemplos de la sesión anterior.</p> <p>Desarrollo Alumnado y profesorado: para los ejemplos vistos en la temática de operaciones con binario, aplicar la sustitución rápida de dígitos binarios de tres en tres para generar el octal y de grupos de cuatro para hexadecimal y ubique cuales son los acarreo que se llevan; utilizan los “relojes” para realizar sumas y obtener acarreo directamente en octal y hexadecimal.</p>  <p>Alumnado y profesorado: aplican los conocimientos a otros ejemplos.</p> <p>Cierre Alumnado y profesorado: identifican en qué momento se generan los acarreo al agrupar bits (3 o 4) para establecer la equivalencia de representaciones binaria, octal y hexadecimal.</p> <p>Extra clase Alumnado: realizan ejercicios de refuerzo y redactan conclusiones.</p>
<p>Obtiene funciones booleanas a partir de sistemas digitales y los reducirá a una expresión equivalente.</p>	<ul style="list-style-type: none"> • Características de una variable booleana. 	<p>Apertura Profesorado: presenta video de <i>Megapprocesador: Computerphile</i> (2017, 2 de mayo). <i>MegaProcessor</i> [video]. https://youtu.be/INa9bQRPM8.</p> <p>Desarrollo Alumnado: en equipos rescatan elementos de teoría de sistemas, sistema binario y anota los términos que resulten desconocidos o novedosos. Alumnado y profesorado: revisan nuevamente el video resaltando lo que se conoce y lo que se revisara en los aprendizajes del álgebra de Boole.</p> <p>Cierre Alumnado y profesorado: identifican en el video los dígitos del sistema binario como variables booleanas, asocia otras disciplinas y situaciones que ocupen variables booleanas: volados, proposiciones lógicas, estados de encendido y apagado, pertenencia en teoría de conjuntos, etcétera.</p> <p>Extra clase Alumnado: revisa el video del sistema binario y las compuertas lógicas: Gentile, N. (2020, 22 de mayo). Sistema binario y puertas lógicas: La base de cómo funciona tu PC [video]. https://youtu.be/RVGIXfC4Xeg</p>

Aprendizajes	Temática	Estrategias sugeridas
	<ul style="list-style-type: none"> Operaciones básicas del álgebra de Boole. 	<p>Apertura Alumnado: en equipos, discute el comportamiento de las compuertas lógicas vistas en el video.</p> <p>Desarrollo Alumnado y profesorado: revisan el comportamiento de las compuertas lógicas básicas: AND y OR en distintas disciplinas (lógica, conjuntos, funciones lógicas, funciones booleanas, circuitos y tablas de verdad). Analizan los conceptos de buffer y negación, aplicándolo para formar las compuertas NAND y NOR. Construyen las compuertas XOR y XNOR a partir de las compuertas anteriores.</p> <p>Cierre Profesorado: con el uso de un simulador o en <i>protoboard</i> de forma física realiza la comprobación de las tablas de verdad.</p>
	<ul style="list-style-type: none"> Concepto de función booleana. Representación de funciones booleanas como expresiones algebraicas. Representa funciones booleanas mediante tablas de verdad. 	<p>Apertura Profesorado: recapitula el tema con una breve revisión acerca de lo que el alumnado investigó sobre las operaciones booleanas (AND, OR, NOT).</p> <p>Desarrollo Profesorado: realiza la explicación teórica acerca de las tablas de verdad y cómo estas representan las salidas de funciones booleanas para las combinaciones posibles en dos entradas, además, muestra cómo la tabla de verdad y la forma algebraica están relacionadas, identifica los comportamientos de las compuertas AND (sólo enciende una combinación) y OR (encienden varias combinaciones). Explica los símbolos lógicos utilizados para representar las diferentes compuertas y su relación con las funciones booleanas. Desarrolla ejemplos prácticos de funciones booleanas mediante la creación de tablas de verdad y analiza cómo estas tablas pueden ser utilizadas para derivar funciones; examinan ejemplos concretos de cómo construir funciones booleanas a partir de las tablas de verdad y su representación del diagrama del circuito lógico.</p> <p>Cierre Alumnado: resuelve problemas que involucren la solución de una función booleana a través de su tabla de verdad y su representación del diagrama del circuito lógico. Fomenta el trabajo colaborativo para la solución de problemas utilizando plataformas tecnológicas o herramientas convencionales no tecnológicas.</p> <p>Extra clase Alumnado: resuelven otros ejercicios para reforzar como tarea. Realizan una investigación sobre la metodología de solución de problemas aplicado a circuitos a partir del modelo de caja negra.</p>

Aprendizajes	Temática	Estrategias sugeridas																				
	<p>Metodología de solución de problemas a partir del modelo de caja negra.</p>	<p>Apertura Alumnado y profesorado: revisan los resultados de la investigación, recuerdan las equivalencias de las funciones booleanas (tablas de verdad, expresión algebraica y circuito) y concluyen que los circuitos equivalentes más reducidos tienen ventajas (simplicidad, menor uso de materiales, menor consumo de energía, etcétera).</p> <p>Desarrollo Alumnado y profesorado: desarrollan un ejemplo que no requiera simplificación, pero que realice el proceso completo. Por ejemplo, una compuerta OR que “pueda verse a simple vista” (planteamiento, análisis de la caja negra, tabla de verdad, obtención de una expresión algebraica equivalente, obtención de un circuito e implementación en un circuito digital, pruebas, correcciones y documentación).</p> <p>Cierre Alumnado y profesorado: concluyen que la aplicación metodológica es aplicable a diversos sistemas y a la construcción de circuitos digitales (se puede simular o implementar en <i>protoboard</i> el resultado).</p> <p>Extra clase Alumnado: investigan los teoremas y postulados del álgebra de Boole.</p>																				
	<p>Funciones booleanas a partir de la tabla de verdad empleando suma de productos (mintérminos).</p>	<p>Apertura Profesorado: introduce el concepto de “mintérminos” como productos que involucran los resultados de la tabla de valor que representan un problema a resolver, por ejemplo, el encendido de una lámpara por medio de un sensor de presencia, variables de entrada a=batería, b=presencia ¿Cuándo enciende la lámpara?</p> <table border="1" data-bbox="571 1221 1359 1632"> <thead> <tr> <th>a</th> <th>b</th> <th>encendido</th> <th>Interpretación</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>No enciende porque no tiene batería ni hay presencia</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>No enciende porque no tiene batería, aunque haya presencia.</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>No enciende porque tiene batería, pero no hay presencia.</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>Si enciende porque hay batería y presencia.</td> </tr> </tbody> </table> <p>Por lo tanto, el resultado de la función que enciende la lámpara: $f(a,b)=ab$</p>	a	b	encendido	Interpretación	0	0	0	No enciende porque no tiene batería ni hay presencia	0	1	0	No enciende porque no tiene batería, aunque haya presencia.	1	0	0	No enciende porque tiene batería, pero no hay presencia.	1	1	1	Si enciende porque hay batería y presencia.
a	b	encendido	Interpretación																			
0	0	0	No enciende porque no tiene batería ni hay presencia																			
0	1	0	No enciende porque no tiene batería, aunque haya presencia.																			
1	0	0	No enciende porque tiene batería, pero no hay presencia.																			
1	1	1	Si enciende porque hay batería y presencia.																			

Aprendizajes	Temática	Estrategias sugeridas
		<p>Desarrollo Profesorado y alumnado: plantean un problema para ser resuelto con más variables.</p> <p>Cierre Profesorado: plantea un problema de “mediana dificultad” como un <i>display</i> de siete segmentos para el despliegue de información (sistema de numeración octal o hexadecimal) desarrollarlo con el grupo.</p> <p>Extra clase Alumnado: identifican patrones a partir de la distributividad para poder simplificar suma de mintérminos.</p>
	Simplificación de funciones booleanas equivalentes con base en postulados y teoremas básicos a partir de la suma de productos de una tabla de verdad.	<p>Apertura Profesorado: plantea que existen diversas formas de aplicar la distributividad (factorización).</p> <p>Desarrollo Profesorado: identifica patrones de reducción de circuitos, identifica términos que puede “clonarse” para agruparlos de diversas formas mediante la aplicación de postulados y teoremas básicos y poder reducir funciones booleanas que redundan en menor cantidad de compuertas lógicas, economiza la construcción del circuito lógico.</p> <p>Cierre Profesorado y alumnado: aplican los postulados y teorema del álgebra de Boole en funciones de circuitos lógicos para obtener un equivalente simplificado.</p> <p>Extra-clase Alumnado: realiza un ejercicio de manera individual aplicando la simplificación de funciones booleanas a partir de los teoremas y postulados del álgebra de Boole.</p>
	Aplicación de mapas de Karnaugh para obtener circuitos lógicos equivalentes.	<p>Apertura Alumnado y profesorado: recapitulan y revisan las soluciones obtenidas.</p> <p>Desarrollo Profesorado: plantea el uso de mapas de Karnaugh como una forma de identificar patrones de simplificación de circuitos digitales y resuelve los problemas planteados en la sesión anterior.</p> <p>Cierre Alumnado: completa la simplificación de los ejercicios realizados por él con el nuevo método.</p> <p>Extra clase Revisa los materiales del portal académico para la construcción de circuitos digitales.</p>

Aprendizajes	Temática	Estrategias sugeridas
<p>Aplica una metodología para la implementación de circuitos lógicos.</p>	<p>Elementos de implementación de circuitos digitales:</p> <ul style="list-style-type: none"> • Fuentes de poder. • Interruptor. • Circuito eléctrico. • Compuerta lógica. • Cables, resistencias y LEDs. • Circuito lógico. 	<p>Apertura Alumnado y profesorado: identifican en la <i>proto-board</i> cómo se implementan los distintos elementos del diagrama lógico.</p> <p>Desarrollo Alumnado y profesorado: mediante un simulador o físicamente, de forma gradual implementa:</p> <ul style="list-style-type: none"> • Interruptores y verifica su correcta polarización, con el uso de un LED. • Comprueba el funcionamiento de compuertas TTL básicas (AND, OR, XOR de dos entradas) enfatizando al alumnado que en cada circuito integrado hay varias compuertas, comprueba cada compuerta con el circuito integrado correspondiente para observar su funcionamiento y compararlo con la tabla del álgebra de Boole correspondiente. <p>Cierre Alumnado: implementa un circuito lógico que involucre el uso de dos circuitos integrados interconectados incluyendo interruptores y LEDs. Comparan en un simulador del circuito digital (iCircuit, Tinkercad u otro) verifican la salida con la tabla de verdad.</p> <p>Extra-clase Alumnado: implementan otros ejemplos, pueden basarse en los provistos en el Portal Académico.</p>
	<p>Construcción o simulación del circuito lógico.</p> <ul style="list-style-type: none"> • Implementa circuitos lógicos en tabletas <i>proto-board</i> como semisumadores, sumadores completos, restadores, otros problemas cotidianos. 	<p>Apertura Profesorado: guía a el alumnado a través de un ejemplo y utilizando un simulador (Tinkercad) o de forma física como se arma el diagrama del circuito lógico para retomarse el ejemplo de la lampara, comparando su comportamiento con el de la tabla de verdad para evaluar su funcionamiento.</p> <p>Desarrollo Profesorado: plantea un problema, para que el alumnado obtenga la función booleana, tabla de verdad y diagrama del circuito lógico implementado en la <i>proto-board</i> o simulador. Alumnado: desarrolla todo el proceso aprendido, y pide ayuda donde tenga dudas al profesorado.</p> <p>Cierre Alumnado: explican ante el grupo como fueron resolviendo el problema planteado.</p> <p>Extra-clase Alumnado: revisan los materiales del portal académico del tema Tableta <i>proto-board</i>.</p>

Aprendizajes	Temática	Estrategias sugeridas
--------------	----------	-----------------------

Relaciona expresiones booleanas con proposiciones lógicas.

Circuitos complejos a partir de circuitos simples (semisumador a sumador completo o restador; sumador y restador de un bit a sumador o restador de dos o más bits, etcétera).

Proposiciones lógicas para la toma de decisiones en un lenguaje de computación mediante variables booleanas.

Apertura

Alumnado y profesorado: plantean el problema de sumar dos dígitos binarios (semisumador) enfatizando el concepto de suma y acarreo y obtienen la tabla de verdad, la función booleana y el diagrama del diseño lógico.

Desarrollo

Alumnado: implementan en la protoboard o el simulador el diagrama del diseño lógico obtenido y comprueban el sumador completo.

Cierre

Alumnado: argumentan de forma grupal que aprendieron y en que lo pueden emplear.

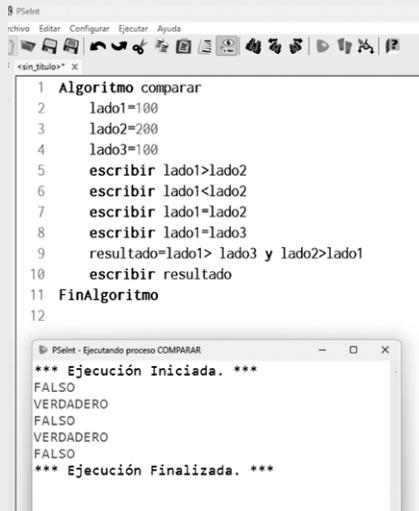
Extra clase

Alumnado: realiza la investigación de los requisitos para tramitar la credencial del INE o de la cartilla, etcétera, cada restricción la ubica como una variable y trata de relacionar al operador lógico que involucre (AND, OR y NOT).

Apertura

Profesorado: establece la analogía entre operadores lógicos con algoritmos con tomas de decisión se puede auxiliar del PSeInt (simulador de algoritmos).

Desarrollo



Alumnado y profesorado: plantean situaciones en las que pueden presentar decisiones (verdadero y falso) y las pueda conectar con las funciones booleanas (AND, OR y NOT), observando el resultado desde el algoritmo.

Cierre

Alumnado y profesorado: revisan el video del megap procesador para identificar elementos que aprendieron en la unidad y obtienen conclusiones.

Profesorado: puede poner a prueba el conocimiento evaluando con un conjunto de reactivos, o pidiéndole al alumnado que desarrolle un "Proyecto" (guiado de una lista de cotejo y con tiempo para su desarrollo).

Evaluación

En las estrategias sugeridas para cada sesión se proponen actividades que brindan evidencias del nivel del logro de los aprendizajes planteados. La evaluación de cada sesión se integra en la evaluación sumativa. Para evaluar los productos obtenidos en cada sesión se sugieren los siguientes tipos e instrumentos:

Diagnóstica

Conocer las características del grupo para trazar la ruta de implementación.

- Preguntas formuladas al inicio de la unidad y de cada clase.
- Análisis de las respuestas dadas en los trabajos extra clase.
- Cuestionarios o actividades interactivas del portal académico.

Formativa

- Evaluación para la reorientación para el logro de los aprendizajes.
- Pruebas de ejecución: codificación del programa.
- Observación: de las actividades realizadas en el salón de clases.
- Cuestionarios: para verificar el seguimiento o aprovechamiento del alumnado.

Sumativa

Análisis de los resultados obtenidos para determinar si se alcanzaron los niveles de aprendizajes previstos en el programa de estudios, que permitan asignar una calificación.

- Listas de verificación: para evaluar los trabajos extra clase.
- Rúbrica: para evaluar las investigaciones o exposiciones realizadas.
- Exámenes.

Actitudinal (AR)

- Valoración de las actitudes y valores desarrollados por el alumnado a lo largo del desarrollo de los aprendizajes de la unidad.
- Empatía: el alumnado comprende las necesidades y perspectivas de las personas con quién interactúa.
- Respeto a la diversidad: el alumnado reconoce y valora la diversidad de género, cultura, habilidades y opiniones de los demás.
- Colaboración y trabajo en equipo: el alumnado aprendió a comunicarse de manera efectiva, resolver conflictos y contribuir al éxito del equipo.
- Creatividad e innovación: el alumnado desarrolló soluciones originales a los problemas propuestos por el profesorado y muestra una predisposición positiva a la innovación.
- Tolerancia a la frustración: el alumnado demuestra una actitud positiva al enfrentarse a desafíos técnicos. Aprende de los errores y sigue mejorando.

Nota: la ponderación de cada actividad será asignada por el profesorado.

Referencias

Para el alumnado

Básicas

Ávila, A. y Baltazar, J. M. (2021). *Sistemas de numeración*. Portal Académico del CCH, UNAM.

<https://portalacademico.cch.unam.mx/cibernetica1/sistemas-de-numeracion>

Mano, M. (2003). *Diseño digital*. Prentice Hall/Pearson.

Pérez, N. A. y Ávila, J. (2021). *Implementación de circuitos lógicos*. Portal Académico del CCH, UNAM.

<https://portalacademico.cch.unam.mx/cibernetica1/implementacion-de-circuitos-logicos>

Complementarias

Mano, M. (2003). *Diseño digital*. Prentice Hall/Pearson.

Computerphile (2017). *MegaProcessor* [video]. <https://youtu.be/lNagbQRPMB8>

Gentile, N. (2020). *Sistema binario y puertas lógicas: La base de cómo funciona tu PC* [video]. <https://youtu.be/RVGIXfC4Xeg>

Pérez, N. A. y Ávila, J. (2021). *Implementación de circuitos lógicos*. Portal Académico del CCH, UNAM.

<https://portalacademico.cch.unam.mx/cibernetica1/implementacion-de-circuitos-logicos>

Calle, D. (2015). *Puertas lógicas AND OR NOT NAND NOR* [Video]. Canal Unicoos. <https://youtu.be/8CRzrOKI96o>

Leal Chapa, C. A., Garza Garza, J. Á., Castillo Castro, J. Á., & Hernández Venegas, J. (2011). *Fundamentos de Diseño Digital. Sistemas Combinacionales*. (1a. edición). Universidad Autónoma de Nuevo León. <https://bit.ly/2GtxEKh>

Les ingenieurs (Director). (2016). *Álgebra booleana-Introducción* [Video]. <https://youtu.be/p58C7OWe3Xk>

Les Ingenieurs (Director). (2016). *Sumador Completo 3 bits* [Video]. <https://youtu.be/qtjohnz4C8c>

Les Ingenieurs (Director). (2019). *Mapa de Karnaugh de 3 variables. Electrónica digital* [Video]. Canal Les Ingenieurs. <https://youtu.be/JAi22PmONnc>

Manso, F. & [Electrónica FP] (Directores). (2019a, 05). *¿Cómo pasar de CIRCUITO a ÁLGEBRA?* [Video]. <https://youtu.be/fokgZU26XHc>

Manso, F., & [Electrónica FP] (Directores). (2019b). *¿Cómo sacar la Tabla de Verdad de un circuito?* [Video]. <https://youtu.be/LPy8okf7PRM>

- Manso, F., & [Electrónica FP] (Directores). (2019c). *Puertas NOT, AND y OR* [Video]. <https://youtu.be/2OH2ahzx6PY>
- Morris, M. M. (2003). *Diseño digital. Tercera Edición* (R. Escalona García, Trad.). Pearson Educación.

Para el profesorado

Básicas

- Ávila, A. y Baltazar, J. M. (2021). *Sistemas de numeración*. Portal Académico del CCH, UNAM. <https://portalacademico.cch.unam.mx/cibernetica1/sistemas-de-numeracion>
- Mano, M. (2003). *Diseño digital*. Prentice Hall/Pearson.
- Pérez, A. y Ávila, J. (2021). *Implementación de circuitos lógicos*. Portal Académico del CCH, UNAM. <https://portalacademico.cch.unam.mx/cibernetica1/implementacion-de-circuitos-logicos>

Complementarias

- Adobe (2021). *Offician Guide to Tinkercad Circuits*. <https://www.tinkercad.com/blog/official-guide-to-tinkercad-circuits>
- Ávila, A. y Baltazar, J. M. (2021). *Sistemas de numeración*. Portal Académico del CCH, UNAM. <https://portalacademico.cch.unam.mx/cibernetica1/sistemas-de-numeracion>
- Computerphile (2017). *MegaProcessor* [video]. <https://youtu.be/lNa9bQRPM8>
- Floyd, L. (1998). *Fundamentos de sistemas digitales*. Prentice–Hall Internacional.
- Garza, J. et al. (2006). *Sistemas digitales y electrónica digital*. Pearson.
- Gentile, N. (2020, 22 de mayo). *Sistema binario y puertas lógicas. La base de cómo funciona tu PC* [video]. <https://youtu.be/RVGIXfC4Xeg>
- Heim, K. (1973). *Álgebra de los circuitos lógicos*. Dossat.
- Mano, M. (2003). *Diseño digital*. Prentice Hall/Pearson.
- Pérez, N. A. y Ávila, J. (2021). *Implementación de circuitos lógicos*. Portal Académico del CCH, UNAM. <https://portalacademico.cch.unam.mx/cibernetica1/implementacion-de-circuitos-logicos>
- Facultad de Contaduría y Administración UNAM. (s/a). *Álgebra de Boole*. http://ecampus.fca.unam.mx/ebook/imprimibles/informatica/arquitectura_computadoras/Unidad_4.pdf

UNIDAD 3. METODOLOGÍA DE SOLUCIÓN DE PROBLEMAS E INTRODUCCIÓN A LA PROGRAMACIÓN EN JAVA

Presentación de la unidad

La Unidad 3 da continuidad de una programación de bajo nivel con unos y ceros que entiende la computadora, para transitar en esta unidad a un lenguaje de alto nivel muy similar al lenguaje en inglés. El alumnado podrá apropiarse de una metodología que lo conduzca a la solución de problemas mediante el desarrollo de un programa informático, utilizando el lenguaje de alto nivel Java.

Carta descriptiva

Propósito	Tiempo
<p>Al finalizar la unidad, el alumnado:</p> <p>Aplicará la metodología de solución de problemas mediante la construcción de algoritmos y la codificación en el lenguaje de programación Java para tener una visión integral del proceso de solución de un problema utilizando la computadora.</p>	32 hrs.

Aprendizajes	Temática	Estrategias sugeridas
<p>El alumnado:</p> <p>Define el concepto de problema. Identifica los elementos de un problema.</p> <p>Identifica los diferentes tipos de problemas: determinísticos, probabilísticos, secuenciales, condicionales y cíclicos.</p>	<p>Definiciones y conceptos generales de un problema. Elementos y relaciones del problema:</p> <ul style="list-style-type: none"> • Entrada. • Proceso. • Salida. <p>Tipos de problema:</p> <ul style="list-style-type: none"> • Determinísticos. • Probabilísticos. • Secuenciales. • Condicionales. • Ciclos. 	<p>Apertura</p> <p>Alumnado: conforma equipos de tres a cuatro personas, el alumnado revisa los objetos de aprendizaje relacionados a la temática que se encuentra en el portal Académico del CCH y realizan las actividades propuestas. Profesorado: presenta ejemplos de problemas abordados, el alumnado determina el tipo de problema e identifican su entrada y salida mediante Diagramas Entrada Proceso Salida (EPS).</p> <p>Desarrollo</p> <p>Alumnado: en binas proponen un tipo de problema de su interés, identifican las causas, insumos o entradas; operaciones o tratamiento efectuados; así como los efectos, productos o salidas para cada problema, el profesorado resuelve dudas y realiza sugerencias</p>

Aprendizajes	Temática	Estrategias sugeridas
		<p>Cierre Alumnado: expone su actividad elaborada en binas y en plenaria se debate, para unificar criterios sobre los conceptos abordados y se obtienen conclusiones.</p> <p>Extra clase Profesorado: propone una serie de problemas y el alumnado selecciona solo uno, determinando el tipo de problema, así como menciona su entrada y salida mediante Diagramas EPS.</p>
<p>Conoce las etapas de la metodología de solución de problemas.</p>	<p>Etapas de la metodología de solución de problemas:</p> <ul style="list-style-type: none"> • Planteamiento del problema. • Análisis del problema. • Diseño de la solución del problema: • Elaboración de algoritmos. • Representación del algoritmo a través de pseudocódigo y diagrama de flujo. • Prueba de escritorio. 	<p>Apertura Profesorado: comienza la clase planteando la importancia de comprender los conceptos fundamentales en programación, el alumnado, con ayuda de una computadora y utilizando fuentes confiables, definirá los siguientes conceptos: algoritmo, diagrama de flujo, pseudocódigo, prueba de escritorio.</p> <p>Desarrollo Profesorado; propone un ejercicio y en conjunto con el grupo se elabora el algoritmo, el diagrama de flujo, el pseudocódigo y la prueba de escritorio, posteriormente el profesorado organiza al alumnado por equipos y les asigna diferentes problemas, cada equipo deberá desarrollar el algoritmo, diagrama de flujo, pseudocódigo y la prueba de escritorio al problema asignado.</p> <p>Cierre Alumnado: cada equipo comparte la solución propuesta de su problema y el diagrama de flujo, para finalizar el profesorado destaca la importancia de estos conceptos como base para el desarrollo de habilidades de programación y resolución de problemas en computación.</p>

Aprendizajes	Temática	Estrategias sugeridas
<p>Conoce el concepto de algoritmo, diagrama de flujo y pseudocódigo.</p> <p>Elabora el algoritmo, diagrama de flujo y pseudocódigo para problemas secuenciales.</p>	<p>Elementos que conforman un algoritmo.</p> <ul style="list-style-type: none"> • Pasos para resolver un problema mediante un algoritmo. • Elementos que conforma un diagrama de flujo. • Transformación del algoritmo a diagrama de flujo. • Elementos de un pseudocódigo. • Transformación de un diagrama de flujo a pseudocódigo. 	<p>Apertura Profesorado: comienza la clase planteando la importancia de comprender los conceptos fundamentales en programación, el alumnado, con ayuda de una computadora y utilizando fuentes confiables, definirá los siguientes conceptos: algoritmo, diagrama de flujo, pseudocódigo, prueba de escritorio.</p> <p>Desarrollo Profesorado; propone un ejercicio y en conjunto con el grupo se elabora el algoritmo, el diagrama de flujo, el pseudocódigo y la prueba de escritorio, posteriormente el profesorado organiza a el alumnado por equipos y les asigna diferentes problemas, cada equipo deberá desarrollar el algoritmo, diagrama de flujo, pseudocódigo y la prueba de escritorio al problema asignado.</p> <p>Cierre Alumnado: cada equipo comparte la solución propuesta de su problema y el diagrama de flujo, para finalizar el profesorado destaca la importancia de estos conceptos como base para el desarrollo de habilidades de programación y resolución de problemas en computación.</p>
<p>Elabora algoritmo, diagrama de flujo y pseudocódigo para problemas condicionales.</p>	<p>Características que conforman a un algoritmo, diagrama de flujo y pseudocódigo para resolver:</p> <ul style="list-style-type: none"> • Problemas tipo condicional simple. • Problemas de tipo de condicionales anidadas. • Problemas de tipo de condicionales múltiples. 	<p>Apertura Profesorado: propone problemas que requieren la evaluación de sentencias condicionales, que involucren estructuras de control de condicionales: para verificar si el dato es válido, ¿hacer ciertas acciones si se cumple algo? ¿qué hacer si se decide entre dos opciones? Comparar valores en tablas, manejo de menús.</p> <p>Desarrollo Profesorado: plantea problemas que requieren el uso de la estructura condicional: por ejemplo, raíces de una ecuación de segundo, el describir el IMC de una personal, identificar qué día no circula, un auto con placas de distinta terminación, registro de documentos para trámite de la credencial del INE. Alumnado en equipo: discuten los problemas y proponen condiciones. Profesorado: formaliza el trazado de las rutas de instrucciones en pseudocódigo correspondientes a cada valor de falso y verdadero. Introducir el uso del símbolo de toma de decisiones (rombo en diagrama de flujo o el uso de las palabras Si-Entonces-FinSi) y resalte la importancia de la indentación.</p> <p>Cierre Alumnado: Identifican las combinaciones de los lazos involucrados en las expresiones booleanas de las sentencias condicionales que se requieren para plantear un plan de pruebas. Optimiza los algoritmos entre el uso de simple, doble (ecuación de segundo grado), anidada (interpretación de tablas). Para situaciones donde los valores son atómicos presente la estructura condicional múltiple (días que circula un auto, presionar teclas en un juego, seleccionar opciones de menú).</p>

Aprendizajes	Temática	Estrategias sugeridas
<p>Construye expresiones lógicas utilizando operadores relacionales y lógicos.</p>	<p>Expresiones y operadores relacionales y lógicos:</p> <ul style="list-style-type: none"> • Operadores relacionales. • Operadores lógicos. • Jerarquía de operadores lógicos. • Evaluación de expresiones lógicas. 	<p>Apertura Profesorado: plantea problemas que responden a preguntas cuyas respuestas son falso o verdadero. Por ejemplo: ¿una persona en las próximas elecciones puede votar si es menor de edad?</p> <p>Desarrollo Profesorado: pregunta al grupo pregunta al grupo por qué serán importantes las operaciones relaciones o las lógicas. Alumnado: en plenaria discute las respuestas y concluye donde se usan los operadores lógicos y relacionales de forma cotidiana. Profesorado: plantea distintas oraciones para que el alumnado identifique en qué momento se utiliza cada operador. Profesorado: plantea algunas expresiones relacionales para que el alumnado las compare con las tablas booleanas revisadas en la Unidad 2.</p> <p>Cierre Plantear ejemplos que combinen ambos operadores.</p>
<p>Construye expresiones aritméticas utilizando la jerarquía de las operaciones y los operadores apropiados en el lenguaje de programación Java.</p>	<p>Expresiones y operadores aritméticos:</p> <ul style="list-style-type: none"> • Asignación. • Operadores aritméticos. • Jerarquía de operadores aritméticos. • Evaluación de expresiones aritméticas. • Funciones aritméticas básicas de la clase <i>Math</i>. 	<p>Apertura Profesorado: presenta los diferentes operadores aritméticos, así como la jerarquía de operadores, el uso correcto de paréntesis y los métodos básicas de la biblioteca <i>Math</i>.</p> <p>Desarrollo Profesorado: organiza equipos, a cada equipo se les facilitará una lista de problemas, donde tendrán que evaluar un conjunto de expresiones aritméticas. Al finalizar los equipos seleccionan un problema y tendrán que codificarlo en el lenguaje de programación Java, se contribuirán sus respuestas con los diferentes equipos.</p> <p>Cierre Profesorado: con su guía y la orientación, participan en la construcción de fórmulas matemáticas para obtener expresiones utilizando solo los operadores aritméticos correspondientes en el lenguaje de programación Java.</p>

Aprendizajes	Temática	Estrategias sugeridas
<p>Elabora el algoritmo, diagrama de flujo y pseudocódigo para resolver problemas cíclicos.</p>	<p>Estructuras de control de secuencia repetitivas:</p> <ul style="list-style-type: none"> • Componentes básicos de una estructura repetitiva: Sentencia de asignación del valor o condición inicial, evaluación de la expresión booleana, sentencia de modificación de estado, sentencias (que se repiten). • Concepto de bandera. • Concepto de contador. • Concepto de acumulador. • Elaboración de algoritmos de ciclo. • Representación de algoritmos de ciclo a través de diagramas de flujo. • Elaboración de algoritmos de ciclo. • Representación de algoritmos de ciclo a través de diagramas de flujo. 	<p>Apertura Profesorado: presenta un problema que requiera el uso de un ciclo de repetición y en plenaria con el alumnado, se buscará diversas formas de solucionarlo.</p> <p>Desarrollo Profesorado: presenta los diferentes conceptos importantes sobre los ciclos de repetición y la forma de representarlos en un algoritmo, diagrama de flujo y pseudocódigo. El profesorado asignará a el alumnado en equipos y les facilitará un problema que requiera el uso de ciclos. Cada equipo tendrá que elaborar el algoritmo, diagrama de flujo y pseudocódigo.</p> <p>Cierre Profesorado resalta las diferencias entre los diferentes tipos de ciclo que existen y la manera de controlarlos.</p> <p>Extra clase Alumnado: identificará un problema que requiera el uso de un ciclo de repetición y elaborará su algoritmo, diagrama de flujo y pseudocódigo.</p>

Aprendizajes	Temática	Estrategias sugeridas
<p>Conoce las generalidades de los lenguajes de programación más utilizados.</p> <p>Conoce la historia del lenguaje de programación Java.</p> <p>Conoce las características básicas del lenguaje de programación Java.</p>	<p>Lenguaje de programación:</p> <p>Lenguajes de programación más utilizados.</p> <ul style="list-style-type: none"> • Paradigmas de programación. • Requerimientos funcionales y no funcionales. • Conjunto de reglas de léxico y de sintaxis. • Código fuente. • Código máquina y objeto. • Compiladores, interpretes. <p>Lenguaje de programación Java:</p> <ul style="list-style-type: none"> • Historia del lenguaje. • Características: <ul style="list-style-type: none"> • Tipos de aplicaciones. • Arquitectura neutral. • Lenguaje orientado a objetos. • Disponibilidad de un amplio conjunto de bibliotecas. • Interpretado. • Robusto. • Distribuido. 	<p>Apertura</p> <p>Profesorado: organiza al alumnado en equipos y en forma aleatoria les asigna alguno de los siguientes temas a investigar: historia, características y entorno de desarrollo del lenguaje de programación Java.</p> <p>Desarrollo</p> <p>Alumnado: en plenaria los equipos exponen la investigación realizada y obtienen conclusiones.</p> <p>Cierre</p> <p>Profesorado: proporciona al alumnado, una guía donde se muestran los pasos que deberán de seguir para realizar la instalación de las herramientas del entorno de desarrollo del lenguaje de programación Java (Eclipse, Netbeans, JCreator, etcétera).</p> <p>Extra clase</p> <p>Alumnado: instala el JDK e IDE en su computadora personal o aplicaciones para el desarrollo de aplicaciones Java en su smartphone o tableta (Jvndroid, Dcoder, etcétera).</p>

Aprendizajes	Temática	Estrategias sugeridas
<p>Conoce el entorno de desarrollo para el lenguaje de programación Java.</p>	<p>Entorno de desarrollo del lenguaje de programación:</p> <ul style="list-style-type: none"> • JDK (kit de desarrollo de Java). • JVM (Máquina virtual de Java). • IDE (interfaz de entorno de desarrollo). • Pasos para implementar un programa con el lenguaje de programación Java y el entorno de desarrollo. • Creación de un proyecto. • Declaración de la clase. • Método <i>main</i>. • Empleo de los métodos <code>System.out. print</code> y <code>System.out.println</code>. • Errores sintácticos y lógicos. 	<p>Apertura Profesorado: realiza una explicación del entorno de desarrollo que se va a utilizar dentro del aula (BlueJ, NetBeans, Eclipse, Replit).</p> <p>Desarrollo Profesorado: Desarrolla en el IDE un programa que permita salida de datos con <code>System.out.println("Mensaje")</code>.</p> <p>Cierre Alumnado: realiza salida de los datos que lleva una caratula de algún trabajo escolar. Pidiendo al profesorado que lo oriente.</p> <p>Extra clase Contestan un cuestionario sobre la temática vista.</p>
<p>Realiza programas empleando la clase <i>scanner</i> para la entrada de datos.</p>	<p>Introducción de datos desde el teclado. La Clase <i>Scanner</i> y sus principales métodos. Instanciación del objeto de la clase <i>Scanner</i>. Entrada estándar de datos por consola (<i>System.in</i>) Tipos de datos primitivos. Enteros: <i>byte, short, int, long</i>. Reales: <i>float, double</i>. Booleanos: <i>boolean</i>. Caracteres: <i>char</i>.</p>	<p>Apertura Alumnado y profesorado realizan un primer programa de prueba en Java. El profesorado borrará elementos del programa inicial para provocar errores de sintaxis que el alumnado deberán de corregir en plenaria.</p> <p>Desarrollo Profesorado: mediante una aplicación de Java presentará a la clase el uso de los diferentes tipos de datos primitivos del lenguaje. Por ejemplo, un programa para calcular el área de un triángulo en donde los valores para la base y la altura sean declarados e inicializados. El alumnado codificará en Java sus primeros algoritmos secuenciales haciendo uso de los tipos de datos primitivos de Java.</p> <p>Cierre Alumnado: con ayuda del profesorado optimizarán el uso de la memoria que hace la aplicación, dependiendo de los valores que tomen las variables de sus programas. Por ejemplo, si las variables toman valores positivos menores a 255 se recomendará usar el tipo <code>byte</code>.</p>

Aprendizajes	Temática	Estrategias sugeridas
Codifica algoritmos condicionales en el lenguaje de programación Java.	Sentencias condicionales: <ul style="list-style-type: none"> • Simples: <i>if</i>. • Doble: <i>if-else</i>. • Anidadas: <i>if-else-if</i>. • Múltiples: <i>switch</i>. 	<p>Apertura Profesorado presenta las sentencias condicionales de Java y las relaciona con sus respectivos diagramas de flujo y pseudocódigos. Revisar los operadores lógicos y relacionales. Completando la tabla de operadores relacionales en Java.</p> <p>Desarrollo Profesorado muestra un ejemplo de codificación de algoritmos secuenciales en Java. El profesorado y el alumnado codifican los algoritmos condicionales elaborados previamente. Identificar sentencias nulas, simples y múltiples (bloques) y cómo opera en estos casos el terminador de sentencia “;”.</p> <p>Cierre Alumno traduce las sentencias lógicas condicionales de sus diagramas de flujo en sentencias de Java identificando la más conveniente para cada tipo de problema: <i>if</i>, <i>if-else</i>, <i>if-else-if</i> o <i>switch</i>.</p> <p>Extra-clase Profesorado: proponer nuevos problemas para codificar directamente en Java y, de forma inversa, obtener sus diagramas de flujo y pseudocódigos.</p>
Codifica algoritmos cíclicos en el lenguaje de programación Java.	Estructuras de control de ciclo. Sentencia <i>for</i> . Sentencia <i>while</i> . Sentencia <i>do-while</i> .	<p>Apertura Profesorado: presenta las sentencias cíclicas de Java y las relaciona con sus respectivos diagramas de flujo y pseudocódigos.</p> <p>Desarrollo Profesorado: muestra un ejemplo de codificación de algoritmos cíclicos en Java. Alumnado: codifican los algoritmos cíclicos previamente elaborados. Identifica variables de control, sus tres ubicaciones dentro de las estructuras (valor inicial, evaluación de la expresión booleana para continuar y la modificación del estado) así como la sentencia que del cuerpo de repetición.</p> <p>Cierre Alumnado: traduce los diagramas de flujo previamente elaborados en sentencias de Java identificando la más conveniente para cada tipo de problema: <i>for</i>, <i>while</i>, <i>do-while</i>. El alumnado propone nuevos problemas para codificar directamente en Java y, de forma inversa, obtener sus diagramas de flujo y pseudocódigos.</p> <p>Extra-clase: Revise el video del Megaprocessador¹ visto en la unidad 2 e identifique las estructuras de control que se plantean entre el minuto 3:20 y 5:35.</p>

¹ Computerphile (2017). *MegaProcessor* [video]. <https://youtu.be/INa9bQRPM8&t=210>

Evaluación

En las estrategias sugeridas para cada sesión se proponen actividades que brindan evidencias del nivel del logro de los aprendizajes planteados. La evaluación de cada sesión se integra en la evaluación sumativa. Para evaluar los productos obtenidos en cada sesión se sugieren los siguientes tipos e instrumentos:

Diagnóstica

Conocer las características del grupo para trazar la ruta de implementación.

- Preguntas formuladas al inicio de la unidad y de cada clase.
- Análisis de las respuestas dadas en los trabajos extra clase.
- Cuestionarios o actividades interactivas del portal académico.

Formativa

Evaluación para la reorientación para el logro de los aprendizajes.

- Pruebas de ejecución: codificación del programa.
- Observación: de las actividades realizadas en el salón de clases.
- Cuestionarios: para verificar el seguimiento o aprovechamiento del alumnado.

Sumativa

Análisis de los resultados obtenidos para determinar si se alcanzaron los niveles de aprendizajes previstos en el programa de estudios, que permitan asignar una calificación.

- Listas de verificación: para evaluar los trabajos extra clase.
- Rúbrica: para evaluar las investigaciones o exposiciones realizadas.
- Exámenes.

Actitudinal (AR)

- Valoración de las actitudes y valores desarrollados por el alumnado a lo largo del desarrollo de los aprendizajes de la unidad.
- Empatía: el alumnado comprende las necesidades y perspectivas de las personas con quién interactúa.
- Respeto a la diversidad: el alumnado reconoce y valora la diversidad de género, cultura, habilidades y opiniones de los demás.
- Colaboración y trabajo en equipo: el alumnado aprendió a comunicarse de manera efectiva, resolver conflictos y contribuir al éxito del equipo.
- Creatividad e innovación: el alumnado desarrolló soluciones originales a los problemas propuestos por el profesorado y muestra una predisposición positiva a la innovación.

- Tolerancia a la frustración: el alumnado demuestra una actitud positiva al enfrentarse a desafíos técnicos. Aprende de los errores y sigue mejorando.

Nota: la ponderación de cada actividad será asignada por el profesorado.

Software auxiliar

- BlueJ. <https://www.bluej.org/>
- Replit. <https://replit.com/>
- Eclipse. <https://www.eclipse.org/>
- NetBeans. <https://netbeans.apache.org/>

Referencias

Para el alumnado

Básicas

Deitel P. y Deitel H. (2016). *Cómo programar en Java*. 10a. edición. Pearson Education.

Eckel, G. (2022). *Piensa en Java* (5a. ed.). Prentice Hall.

Joyanes, L. (2003). *Fundamentos de la programación. Algoritmos, estructura de datos y objetos*. McGraw-Hill.

Joyanes, L. (1990). *Problemas de metodología de la programación, 468 problemas resueltos*. McGraw-Hill.

Complementarias

Computerphile (2017). *MegaProcessor* [video].

<https://youtu.be/INa9bQRPMB8&t=210>

Ávila, J. y Ávila, A. (2024). *Estructuras repetitivas*. Portal Académico del CCH, UNAM. <https://portalacademico.cch.unam.mx/cibernetica2/estructuras-repetitivas> (en construcción)

Ávila, J. y Baltazar, J. M. (2023). *Estructuras condicionales*. Portal Académico del CCH, UNAM.

<https://portalacademico.cch.unam.mx/cibernetica2/estructuras-condicionales>

- Ávila, J. (2022). *Lenguaje de programación orientado a objetos con Java*. Portal Académico del CCH, UNAM.
<https://portalacademico.cch.unam.mx/cibernetica2/programacion-java>
- Bailón, J. y Baltazar, J. M. (2021). *Algoritmos y codificación*. Portal Académico del CCH, UNAM.
<https://portalacademico.cch.unam.mx/cibernetica1/algoritmos-y-codificacion>
- Ávila, J. y Bailón, J. (2022). *Análisis y diseño en POO*. Portal Académico del CCH, UNAM.
<https://portalacademico.cch.unam.mx/cibernetica1/analisis-y-diseno-en-poo>

Para el profesorado

- Amhed, A. y Prasad, B. (2016). *Foundations of Software Design*. CRC Press.
- Ávila, J. y Hernández, A. (2012). *Informe del seminario institucional para la elaboración de material didáctico interactivo de Cibernética y Computación I*. ENCCH-UNAM.
- Barnes, D.J. y Kölling, M. (2007). *Programación orientada a objetos con Java, una introducción práctica usando BlueJ*. Pearson Educación.
- Coad, P. y Yourdon, E. (1991). *Object Oriented Analysis*. Yourdon Press.
- Deitel P. y Deitel H. (2016). *Como programar en Java*. 10a. edición. Pearson Education. Capítulos 20-34. https://www.pearsonenespanol.com/mexico/educacion-superior/deitel_index/como-programar-en-java-10e
- Hernández, A. y Ávila, J. (2013). *Informe del seminario institucional para la elaboración de material didáctico interactivo de Cibernética y Computación II*. ENCCH-UNAM.



Cibernética y
Computación II

PRESENTACIÓN DE LA ASIGNATURA DE CIBERNÉTICA Y COMPUTACIÓN II

En la *Unidad 1. Fundamentos de programación orientada a objetos con Java*, se describen las características del lenguaje de programación Java y la metodología para el desarrollo de aplicaciones orientadas a objetos, definiendo clases, atributos y métodos para la implementación de programas en un Entorno de Desarrollo Integrado (IDE por sus siglas en inglés).

En la *Unidad 2. Estructuras de control de secuencia en Java*, se utilizan las estructuras de control de secuencia para desarrollar los métodos de una clase que den la solución a problemas con un lenguaje de programación orientado a objetos Java.

En la *Unidad 3. Herencia y polimorfismo*, se elaboran programas en Java utilizando herencia y polimorfismo como mecanismos de reutilización de código y especialización o personalización de clases.

En la *Unidad 4. Interfaz Gráfica de Usuario*, el alumnado retoma todos los aprendizajes adquiridos a lo largo del curso para resolver problemas y/o proyectos en un ambiente de modo gráfico utilizando las clases del paquete Swing. Se recomienda que los programas en Java a resolver sean apropiados al nivel de estudios del alumnado.

El curso pretende crear un pensamiento computacional, construyendo las bases para que en sus estudios posteriores puedan profundizar sus conocimientos.

En general, se sugiere que en cada sesión de dos horas se obtenga un aprendizaje, desarrollando la temática y haciendo uso de las estrategias que incluyen actividades de apertura, desarrollo, cierre y trabajo extra-clase. Dado que la evaluación es un proceso que permite conocer de manera sistemática el logro de los aprendizajes, las estrategias sugeridas proponen actividades que propician que el alumnado se auto evalúe, y a la vez, le dé información al profesorado sobre el logro de los aprendizajes.

UNIDAD 1. FUNDAMENTOS DE PROGRAMACIÓN ORIENTADA A OBJETOS CON JAVA

Presentación de la unidad

En la unidad el alumnado podrá conocer el paradigma de la programación orientada a objetos POO, sus pilares fundamentales: encapsulamiento, abstracción, herencia y polimorfismo y de forma detallada los primeros dos, para que pueda diseñar bajo el Lenguaje de Modelado Unificado UML e implementar dichos conceptos dentro de un código en lenguaje java y así crear sus programas orientados a objetos.

Carta descriptiva

Propósito	Tiempo
<p>Al finalizar la unidad, el alumnado:</p> <p>Conocerá las características de la programación orientada a objetos en el lenguaje Java, para resolver problemas desde la definición de las clases hasta la instanciación de objetos.</p>	10 hrs.

Aprendizajes	Temática	Estrategias sugeridas
<p>El alumnado:</p> <p>Conoce los conceptos básicos de la programación orientada a objetos.</p> <p>Conoce la organización general de un programa en Java como lenguaje orientado a objetos.</p>	<p>Conceptos básicos de la programación orientada a objetos.</p> <p>Paquete.</p> <p>Clase.</p> <p>Atributos o características.</p> <p>Métodos o comportamiento.</p> <p>Objeto.</p> <ul style="list-style-type: none"> - Identidad. - Atributos. - Métodos. <p>Características de la programación orientada a objetos.</p> <p>Abstracción.</p> <p>Encapsulamiento.</p> <p>Herencia.</p> <p>Polimorfismo.</p>	<p>Apertura</p> <p>Profesorado: le pide al alumnado crear equipos de trabajo y solicita que realicen un organizador gráfico a su gusto sobre los conceptos a abordar.</p> <p>El alumnado presenta sus organizadores gráficos y llegan a conclusiones sobre los conceptos.</p> <p>Desarrollo</p> <p>Profesorado: solicita al alumnado de manera individual identifiquen los atributos (características) y métodos (acciones) comunes a una clase de objetos, como pueden ser: celulares, tabletas, computadoras, consolas de videojuegos, televisores, etcétera.</p> <p>Alumnado: plenaria presenta su trabajo y con base en dichos ejemplos, el profesorado generaliza la información relevante de las diferentes clases de los objetos expuestos, mediante diagramas de clase y de objetos (abstracción).</p> <p>Cierre</p> <p>Alumnado y profesorado: concluyen y se retroalimentan de lo aprendido en clase. Se sugiere realizar una actividad interactiva con algún software o documento impreso (consultar las referencias).</p> <p>Extra clase</p> <p>El alumnado: elabora de forma similar a la trabajada en clase los diagramas para los siguientes casos: alumnado, trabajadores y trabajadoras, figuras geométricas, datos de un paciente, etcétera.</p>

Aprendizajes	Temática	Estrategias sugeridas
<p>Conoce los estándares del Lenguaje Unificado de Modelado (UML) para la elaboración de un diagrama de clase y de objeto.</p> <p>Elabora los diagramas de clase y de objetos usando el Lenguaje Unificado de Modelado (UML).</p>	<p>Estructura del diagrama de objetos: Identificación. Datos. Métodos. Estructura del diagrama de clase: Nombre de la clase. Atributos. Métodos. Modificadores de acceso (Niveles de visibilidad): De los atributos: - Privados. - Públicos. - Protegidos. - Estáticos. De los métodos: - Privados. - Públicos. - Protegidos. - Estáticos. - Abstractos.</p>	<p>Apertura El alumnado, usando de preferencia las fuentes de consulta de este documento, realiza una investigación documental sobre los estándares del Lenguaje de Modelado Unificado (UML por sus siglas en inglés) para la representación gráfica de una clase de objetos.</p> <p>Desarrollo Alumnado: con base en los primeros diagramas elaborados por ellos de forma extra clase (trabajadores, figuras geométricas, datos de un paciente, etcétera), de forma conjunta alumnado y profesorado utilizan el UML para la elaboración de las representaciones gráficas de las clases y de los objetos de uno de los casos. Profesorado: ejemplifica el uso de los modificadores de acceso para manipular los niveles de visibilidad de los atributos y métodos de una clase. Alumnado: integra a sus diagramas de clase los símbolos relacionados a cada modificador de acceso empleado en cada atributo y método.</p> <p>Cierre Alumnado: de forma individual el alumnado elabora los diagramas de clase y de objetos usando el UML del resto de los ejercicios extra clase.</p> <p>Extra clase Alumnado: realizará investigación documental sobre los métodos constructores, <i>getters</i> y <i>setters</i> de una clase.</p>
<p>Implementa la codificación en Java del modelo de clase, utilizando el diagrama de clase (atributos y los 3 métodos constructor, <i>setter</i> y <i>getter</i>).</p>	<p>Implementa el modelo de clase:</p> <ul style="list-style-type: none"> • Declaración de la clase. • Declaración de atributos. • Declaración e implementación de los métodos constructor, <i>setters</i> y <i>getters</i>. 	<p>Apertura Alumnado: en plenaria con base en la investigación documental obtienen conclusiones sobre los métodos constructores, <i>getter</i> y <i>setters</i> de una clase.</p> <p>Desarrollo Profesorado: Con base en los diagramas de clase desarrollados ejemplifica la codificación de una de esas clases en Java, especificando: declaración de la clase, declaración de atributos e implementación de los métodos <i>getter</i>, <i>setter</i> y constructores.</p> <p>Cierre Alumnado: elabora de forma individual la codificación en Java de las clases de los ejercicios restantes.</p> <p>Extra clase Alumnado: realiza una investigación documental sobre la arquitectura modelo-vista-controlador.</p>

Aprendizajes	Temática	Estrategias sugeridas
<p>Emplea la arquitectura modelo-vista-controlador para instanciar una clase.</p> <p>Escribe el código del controlador para la instanciación de la clase.</p>	<p>Arquitectura modelo-vista-controlador: Modelo. Vista. Controlador.</p> <p>Diagrama Clase-Objeto. Instanciación. Crear objetos. Hacer llamada a los métodos (notación punto).</p>	<p>Apertura Profesorado: explicará al alumnado un proyecto orientado a objetos en Java, utilizando el IDE que tiene como herramienta didáctica en el centro de cómputo, de su preferencia desde la creación de un proyecto, el paquete, y los archivos para la clase de los objetos y la clase principal o controladora. Alumnado: de forma similar a la mostrada en clase creará el proyecto para uno de los casos previamente trabajados (trabajadores, figuras geométricas, pacientes, etcétera).</p> <p>Desarrollo Alumnado: con base en el proyecto ejemplo, el profesorado mostrará la instanciación de objetos desde la clase controladora usando el <i>operador new</i> y los constructores de la clase. El alumnado de forma individual codificará la instanciación de distintos modelos de clases creadas hasta el momento.</p> <p>Cierre Profesorado: utilizando el programa ejemplo utilizará los métodos <i>getters</i> y <i>setters</i> de la clase para manipular la información de cada uno de los objetos instanciados (encapsulamiento). El alumnado modificará el código de la clase controladora para que desde ella se pueda actualizar la información de sus objetos.</p> <p>Extra clase Alumnado: realizan una investigación documental sobre la declaración en Java de los métodos de clase y de instancia</p>
<p>Elabora la solución a un problema definiendo la clase e instanciando el objeto.</p>	<p>Métodos de clase y de instancia. Componentes de un método de clase y de instancia: Modificador de acceso. Tipo de dato de retorno. Identificador. Parámetros. Bloque de sentencias.</p>	<p>Apertura Alumnado: con base en la investigación documental elaborada y los proyectos de las sesiones previas. El profesorado y el alumnado generalizan la declaración de un método en Java (modificador de acceso, tipo de dato de retorno, identificador, parámetros y bloque de sentencias).</p> <p>Desarrollo Profesorado: ejemplificará la codificación que dé solución a una problemática en Java, implementado todo lo resuelto hasta este momento, modelo de la clase, atributos, métodos (constructor, <i>getter</i>, <i>setter</i> y de función) para finalizar con el método de instancia, por ejemplo, el problema puede ser el cálculo del salario bruto, el cálculo de la nómina.</p> <p>Cierre Alumnado: desde otro problema planteado se le pide que de forma individual realice el modelo de la clase y su instancia.</p> <p>Extra clase Alumnado: elabora y documenta un proyecto entregable orientado a objetos en Java aplicando todos los aprendizajes de la unidad.</p>

Evaluación

En las estrategias sugeridas para cada sesión se proponen actividades que brindan evidencias del nivel del logro de los aprendizajes planteados. La evaluación de cada sesión se integra en la evaluación sumativa. Para evaluar los productos obtenidos en cada sesión se sugieren los siguientes tipos e instrumentos:

Diagnóstica

Conocer las características del grupo para trazar la ruta de implementación.

- Preguntas formuladas al inicio de la unidad y de cada clase.
- Análisis de las respuestas dadas en los trabajos extra clase.
- Cuestionarios o actividades interactivas del portal académico.
- Formativa.
- Evaluación para la reorientación para el logro de los aprendizajes.
- Pruebas de ejecución: codificación del programa.
- Observación: de las actividades realizadas en el salón de clases.
- Cuestionarios: para verificar el seguimiento o aprovechamiento del alumnado.

Sumativa

Análisis de los resultados obtenidos para determinar si se alcanzaron los niveles de aprendizajes previstos en el programa de estudios, que permitan asignar una calificación.

- Listas de verificación: para evaluar los trabajos extra clase.
- Rúbrica: para evaluar las investigaciones o exposiciones realizadas.
- Exámenes.

Actitudinal (AR)

- Valoración de las actitudes y valores desarrollados por el alumnado a lo largo del desarrollo de los aprendizajes de la unidad.
- Empatía: el alumnado comprende las necesidades y perspectivas de las personas con quién interactúa.
- Respeto a la diversidad: el alumnado reconoce y valora la diversidad de género, cultura, habilidades y opiniones de los demás.
- Colaboración y trabajo en equipo: el alumnado aprendió a comunicarse de manera efectiva, resolver conflictos y contribuir al éxito del equipo.
- Creatividad e innovación: el alumnado desarrolló soluciones originales a los problemas propuestos por el profesorado y muestra una predisposición positiva a la innovación.

- Tolerancia a la frustración: el alumnado demuestra una actitud positiva al enfrentarse a desafíos técnicos. Aprende de los errores y sigue mejorando.

Nota: la ponderación de cada actividad será asignada por el profesorado.

Software auxiliar

- BlueJ. <https://www.bluej.org/>
- Replit. <https://replit.com/>
- Eclipse. <https://www.eclipse.org/>
- NetBeans. <https://netbeans.apache.org/>

Referencias

Para el alumnado

Básicas

Ávila, M. del S., et al. (2018). *Paquete didáctico para la asignatura de Cibernética y Computación I*. cch-oriente.unam.mx/pdf/materiales/paqueteCyC.pdf

Henao, C. (2013). *Conceptos Básicos de Programación Orientada a Objetos*. <https://codejavu.blogspot.com/2013/05/conceptos-de-programacion-orientada.html>

Jiménez, J., Jiménez E. M. y Alvarado L. N: (2014). *Fundamentos de programación, diagramas de flujo, diagramas, diagramas N-S, pseudocódigo y Java*. Alfaomega. <https://content.e-bookshelf.de/media/reading/L-18448289-5aeff52fbc.pdf>

Joyanes, L. y Zahonero, M. (2014). *Programación en c, c++, Java*. McGraw-Hill Interamericana. <https://idoc.pub/documents/programacion-en-c-c-java-y-uml-2da-edicion-luis-joyanes-aguilar-librosvirtualcom-2nv59vro3rlk>

Joyanes, L. y Zahonero, M. (2011). *Programación en Java: algoritmos, programación orientada a objetos e interfaz gráfica de usuarios*. McGraw-Hill Interamericana RUA.

Moisset, D. (2016). *Curso de programación Java*. <https://tutorialesprogramacionya.com/javaya/>

Pérez G. (2008), *Aprendiendo Java y POO*.

<https://www.cartagenag9.com/recursos/programacion/apuntes/AprendiendoJava.pdf>

Sznajdleder. P. (2012). *Java a fondo, estudio del lenguaje y desarrollo de aplicaciones*. Alfaomega.

Complementarias

Bernal, J. (2012). *Programación orientada a objetos con Java*. Universidad Politécnica de Madrid. Máster Universitario en Ingeniería Web. https://www.etsisi.upm.es/sites/default/files/curso_2013_14/MASTER/MIWJEE.POOJ.pdf

Ceballos, F. (2015). *Java. Interfaces gráficas y aplicaciones para Internet*. Ra-Ma

Martín, A. (2014). *Programador Java certificado: curso práctico*. Ra-Ma.

Deitel H. (2016). *Como programar en Java*. 10a. edición. Pearson Education.

Ossio, S. W., Prudencio, A. M., Salgado I. y Valda, F. S. (2007). *Programación II Guía de Java + ejercicios*. Universidad Católica Boliviana “San Pablo”.

Para el profesorado

Básicas

Blasco, F. (2019). *Programación orientada a objetos en Java*. Ediciones de la U. BIDI

Cairó, O. (1995). *Metodología de la programación, algoritmos, diagrama de flujo y programas*. Computec.

Cardona, S. (2008), *Introducción a la programación en java*. Editorial Elizcom.

Casanova, A. *Empezar a programar usando Java*.

<http://site.ebrary.com.pbidi.unam.mx:8080/lib/bibliodgbsp/detail.action?docID=10831789>

Deitel P. y Deitel H. (2016). *Cómo programar en Java*. 10a. edición. Pearson Education.

Joyanes, L. & Zahonero, M. (2014). *Programación en c, c ++, Java y uml*. McGraw-Hill Interamericana.

López, L. (2013). *Metodología de la programación orientada a objetos*. 2a. edición. Editorial Alfaomega.

Malik, D.S. (2013). *Programación jaca del análisis de problemas al diseño de programa*. Quinta edición (Trad. León, J.). Cengage Learning Editores. BIDI

Moreno, J. (2015). *Programación orientada a objetos*. Ra-Ma.

Rivas, M. (2021). *Cuadernillo de ejercicios para trabajo en casa. Módulo: Programación Orientada a Objetos*. Conalep-Xalapa.

<https://www.conalepveracruz.edu.mx/iniciobackup/wp-content/uploads/03/2021/>

UNIDAD 2. ESTRUCTURAS DE CONTROL DE SECUENCIA EN JAVA

Presentación de la unidad

Dentro de un lenguaje de programación se podría ir haciendo instrucción tras instrucción de forma secuencial, sin embargo, al conocer las estructuras de control el alumnado tiene la posibilidad de alterar, controlar o modificar el orden del flujo en el que se ejecutan las instrucciones, al tomar decisiones, creando ciclos de repetición, guardando datos en arreglos, dando así, mayor versatilidad a la forma de resolver problemáticas.

Carta descriptiva

Propósito	Tiempo
<p>Al finalizar la unidad, el alumnado:</p> <p>Utilizará las estructuras de control de secuencia para la resolución de problemas a través del lenguaje de programación orientado a objetos con Java.</p>	24 hrs.

Aprendizajes	Temática	Estrategias sugeridas
<p>El alumnado:</p> <p>Desarrolla programas que involucren las estructuras condicionales simples, compuestas, anidadas y múltiples en los métodos de una clase.</p>	<p>Estructuras condicionales:</p> <p>Simple: <i>if</i>.</p> <p>Compuestas: <i>if-else</i></p> <p>Anidadas</p> <p>Operador ternario.</p> <p>Implementación.</p> <p>Estructura condicional múltiple:</p> <p><i>Switch</i>.</p> <p>Implementación.</p>	<p>Apertura</p> <p>Alumnado: conformados en equipo, investigan los conceptos relativos a las estructuras condicionales.</p> <p>Desarrollo</p> <p>Profesorado: plantea problemas que empleen estructuras condicionales simples, compuestas, anidadas y múltiples en métodos de una clase, haciendo énfasis en la importancia de los distintos tipos de operadores relacionales y en la identificación de cuándo debe utilizarse qué tipo de condición.</p> <p>Cierre</p> <p>Alumnado: propone un problema similar a los vistos anteriormente y elabora el programa de solución.</p> <p>Extra clase</p> <p>El profesorado plantea nuevos problemas en cuya solución se empleen estructuras condicionales simples, compuestas, anidadas y múltiples, e invita al alumnado a subirlos a la plataforma.</p>

Aprendizajes	Temática	Estrategias sugeridas
<p>Identifica y codifica excepciones a través de las sentencias <i>try</i> y <i>catch</i>.</p>	<p>Concepto de excepción <i>try</i> y <i>catch</i>.</p>	<p>Apertura Alumnado: conforman equipos, para investigar los conceptos relativos al manejo de excepción <i>try</i> y <i>catch</i>.</p> <p>Desarrollo Profesorado: tomando en cuenta la investigación previa realizada por el alumnado, explica el concepto de excepción relativa a los casos. Conversión a número entero una cadena (<i>string</i>) que no contiene valores numéricos. Operación de la división, si el divisor toma el valor de cero. Intentar acceder o modificar atributos con valores fuera de rango. Explica el uso de la sentencia <i>try</i> y <i>catch</i> en los métodos de una clase, además de plantear problemas para su aplicación, el alumnado bajo la estricta supervisión del profesorado los desarrolla.</p> <p>Cierre Alumnado y profesorado: en plenaria exponen la solución de los problemas planteados, en forma ordenada y respetuosa, sus compañeros opinan y comparan su solución.</p>
<p>Codifica programas para resolver problemas que involucren estructuras repetitivas <i>for</i>, <i>while</i> y <i>do-while</i>.</p>	<p>Estructura repetitiva</p> <ul style="list-style-type: none"> • <i>for</i>. • <i>while</i>. • <i>do-while</i>. 	<p>Apertura Alumnado: conforma equipos, para realizar una investigación sobre los conceptos relativos a las estructuras repetitivas <i>for</i>, <i>while</i> y <i>do-while</i>.</p> <p>Desarrollo Profesorado: explica los conceptos relativos a las estructuras repetitivas <i>for</i>, <i>while</i> y <i>do-while</i>. Plantea problemas que empleen las estructuras <i>for</i>, <i>while</i> y <i>do-while</i> en métodos de una clase. En cada momento el profesorado supervisa y hace comentarios al alumnado para que comprenda los temas.</p> <p>Cierre Alumnado: en plenaria exponen la solución de los problemas planteados, en forma ordenada y respetuosa, sus compañeros opinan y comparan su solución.</p> <p>Extra clase Profesorado: plantea problemas adicionales en cuya solución se empleen las estructuras repetitivas, e invita al alumnado a resolverlas empleando algunas de las estructuras vistas en clase <i>for</i>, <i>while</i> y <i>do-while</i>, solicita al alumnado a que justifique por qué selecciono la estructura de ciclo, el trabajo desarrollado lo deberán subir a la plataforma.</p> <p>Nota. Se recomienda plantear problemas relativos a:</p>

Aprendizajes	Temática	Estrategias sugeridas
		<ul style="list-style-type: none"> • Asignación del monto de beca con base al promedio obtenido durante el semestre. • Determinar el importe por la compra de diversos productos, considerando que se otorga un porcentaje de descuento con base en la compra realizada. • Convertir la calificación de número a letra. <p>Solución de:</p> <ul style="list-style-type: none"> • Ecuación de primer grado. • Sistema de ecuaciones simultáneas. • Ecuación de segundo grado. • En un verficentro hay que determinar el promedio de puntos contaminantes de los vehículos atendidos durante un día, así como el vehículo que menos y más contamina. • Operación de un cajero automático. • Con base al sueldo diario, días y horas extras trabajadas, determinar el sueldo bruto de un empleado. • Índice de masa corporal. • Promedios de índice de masa corporal para un grupo de personas.
<p>Desarrolla programas que involucren la estructura repetitiva <i>while</i> en los métodos de una clase.</p>	<p>Estructura repetitiva:</p> <ul style="list-style-type: none"> • <i>while</i>. • Implementación. 	<p>Apertura Profesorado: explica el uso de la estructura repetitiva <i>while</i> en los métodos de una clase, tomando como base la investigación previa realizada por el alumnado, y desarrolla ejemplos de su aplicación.</p> <p>Desarrollo Profesorado: plantea problemas que empleen la estructura repetitiva <i>while</i> en los métodos de una clase. El alumnado elabora los programas correspondientes.</p> <p>Cierre Alumnado: en plenaria exponen la solución a los problemas resueltos, opinan y comparan su solución.</p> <p>Extraclase Alumnado: investiga los conceptos relativos a la estructura repetitiva <i>do-while</i>.</p>

Aprendizajes	Temática	Estrategias sugeridas
<p>Desarrolla programas que involucren la estructura repetitiva <i>do-while</i> en los métodos de una Clase.</p>	<p>Estructura repetitiva:</p> <ul style="list-style-type: none"> • <i>do-while</i>. • Implementación. 	<p>Apertura Profesorado: ejemplifica el uso de la estructura repetitiva <i>do-while</i> en los métodos de una clase, tomando en cuenta la investigación previa realizada por el alumnado.</p> <p>Desarrollo Profesorado: plantea problemas que empleen la estructura repetitiva <i>do-while</i> en los métodos de una clase, el alumnado elabora los programas correspondientes.</p> <p>Cierre Alumnado: en plenaria exponen la solución a los problemas resueltos, opinan y comparan su solución.</p> <p>Extra clase Alumnado: en equipo, investigan los conceptos relativos a arreglos unidimensionales.</p>
<p>Resuelve problemas que involucren el uso de los arreglos unidimensionales en los métodos de una clase.</p> <p>Conoce el concepto de estructuras de datos en Java.</p> <p>Conoce el concepto de estructura de datos estática.</p> <p>Describe el concepto de los arreglos unidimensionales en los métodos de una clase.</p>	<p>Arreglos unidimensionales:</p> <p>Concepto. Declaración. Tipos. Uso. Recorrido.</p>	<p>Apertura Profesorado: tomando en cuenta la investigación previa realizada por el alumnado, explica el uso de arreglos unidimensionales en los métodos de una clase.</p> <p>Desarrollo Profesorado: plantea problemas en los que se utilicen arreglos unidimensionales en los métodos de una clase, explica las estructuras de datos estática, el uso de arreglos unidimensionales en los métodos de una clase y el uso de la estructura repetitiva <i>foreach</i>, el alumnado elabora los programas bajo la guía del profesorado.</p> <p>Cierre Alumnado: exponen la solución a los problemas resueltos, opinan y comparan su solución en forma ordenada y respetuosa.</p> <p>Extra clase Alumnado: resuelven problemas, planteados por el profesorado, que empleen arreglos unidimensionales en los métodos de una clase.</p>

Aprendizajes	Temática	Estrategias sugeridas
<p>Codifica programas que involucren el uso de los arreglos unidimensionales en los métodos de una clase que involucren la estructura repetitiva <i>for- each</i>.</p>	<p>Estructura repetitiva:</p> <ul style="list-style-type: none"> • <i>for-each</i> • Implementación. 	<p>Apertura Profesorado: propone, al menos, 3 problemas que se resuelvan empleando arreglos unidimensionales.</p> <p>Desarrollo Alumnado: en equipos de trabajo, elige alguno de los problemas planteados relativos a arreglos unidimensionales en los métodos de una clase. Bajo la guía del profesorado, el alumnado los desarrolla.</p> <p>Cierre Alumnado: en plenaria los equipos de trabajo exponen la solución de los problemas planteados. En forma ordenada y respetuosa, sus compañeros opinan y comparan su solución.</p> <p>Nota. Se recomienda plantear problemas relativos a:</p> <ul style="list-style-type: none"> • Calcular el promedio de un grupo de 10 estudiantes. <p>Con base a las calificaciones obtenidas de los 10 estudiantes:</p> <ul style="list-style-type: none"> • Acomodarlas. • Menor a mayor. • Mayor a menor <p>Con base al primer apellido de los 10 estudiantes, acomodarlos en orden alfabético.</p>
<p>Describe los conceptos de los arreglos bidimensionales en los métodos de una clase.</p>	<p>Arreglos bidimensionales:</p> <ul style="list-style-type: none"> • Concepto. • Declaración. • Uso. • Acceso. • Recorrido. 	<p>Apertura Alumnado: en equipo, investiga los conceptos relativos a los arreglos bidimensionales.</p> <p>Desarrollo Profesorado: tomando en cuenta la investigación previa del alumnado, explica el uso de arreglos bidimensionales en los métodos de una clase y plantea problemas relativos a arreglos bidimensionales en los métodos de una clase; se desarrolla lo supervisado por el profesorado.</p> <p>Cierre Alumnado: en reunión plenaria expone la solución de los problemas planteados. En forma ordenada y respetuosa, sus compañeros opinan y comparan su solución.</p>

Aprendizajes	Temática	Estrategias sugeridas
<p>Codifica programas que involucren el uso de los arreglos bidimensionales en los métodos de una clase.</p> <p>Elabora un proyecto que utilice las sentencias vistas hasta el momento, incluyendo arreglos unidimensionales y / o bidimensionales.</p>	<p>Arreglos bidimensionales:</p> <ul style="list-style-type: none"> • Implementación. 	<p>Apertura Profesorado: propone al menos 3 problemas que se resuelvan empleando arreglos bidimensionales.</p> <p>Desarrollo Alumnado: en equipos de trabajo, eligen alguno de los problemas planteados relativos a arreglos bidimensionales en los métodos de una clase, bajo guía del profesorado el alumnado los desarrolla.</p> <p>Cierre Alumnado: en reunión plenaria, los equipos de trabajo exponen la solución de los problemas planteados. En forma ordenada y respetuosa, sus compañeros opinan y comparan su solución.</p> <p>Nota. Se recomienda plantear problemas relativos a:</p> <ul style="list-style-type: none"> • Suma y resta de matrices. • Sumatoria de la diagonal principal de una matriz. <p>Con base en la producción trimestral de maíz, frijol, o cualquier otro producto de 10 diferentes estados de la República Mexicana, determinar:</p> <ul style="list-style-type: none"> • Producción anual de la República Mexicana. • Estado de la República Mexicana con menor y mayor producción anual. • Trimestre con menor y mayor producción. • Con base en el sueldo bruto de un empleado, calcular el impuesto sobre la renta a retener.

Aprendizajes	Temática	Estrategias sugeridas
<p>Conoce el concepto de estructura de datos dinámica.</p> <p>Describe el concepto de ArrayList en los métodos de una clase.</p> <p>Codifica programas que involucren el uso de los ArrayList en los métodos de una clase.</p>	<p>ArrayList:</p> <ul style="list-style-type: none"> • Concepto. • Declaración. • Añadir elementos. • Eliminar elementos. • Acceso a los elementos. • Recorrido. • Implementación. 	<p>Apertura Alumnado: en equipo investigan los conceptos relativos a las estructuras de datos dinámicas en Java: ArrayList.</p> <p>Desarrollo Profesorado: toma en cuenta la investigación previa realizada por el alumnado, explica el uso de los ArrayList en los métodos de una clase y plantea problemas relativos a ArrayList. Bajo la supervisión del profesorado, el alumnado los desarrolla.</p> <p>Cierre Alumnado: en reunión plenaria exponen la solución de los problemas planteados. En forma ordenada y respetuosa, sus compañeros opinan y comparan su solución.</p> <p>Nota. Se recomienda plantear problemas relativos a: Listado de personas con sus respectivos índices de masa corporal. Para ello, es necesario crear dos clases (“Persona” y “Listado_IMC”); en esta última clase se implementará un ArrayList de objetos persona. Programa de registro biológico: para ello se debe plantear la clase “Especie” con sus atributos nombre, tipo y población, además de crear la clase “Registro Biológico” que haga uso de la clase ArrayList, con las funcionalidades de agregar y visualiza. Tienda virtual de libros: modelar las clases “Libro”, “Inventario” con sus respectivos atributos y métodos, así como la clase “Tienda_Virtual”.</p>

Evaluación

En las estrategias sugeridas para cada sesión se proponen actividades que brindan evidencias del nivel del logro de los aprendizajes planteados. La evaluación de cada sesión se integra en la evaluación sumativa. Para evaluar los productos obtenidos en cada sesión se sugieren los siguientes tipos e instrumentos:

Diagnóstica

Conocer las características del grupo para trazar la ruta de implementación.

- Preguntas formuladas al inicio de la unidad y de cada clase.
- Análisis de las respuestas dadas en los trabajos extra-clase.
- Cuestionarios o actividades interactivas del portal académico.
- Formativa.
- Evaluación para la reorientación para el logro de los aprendizajes.
- Pruebas de ejecución: codificación del programa.
- Observación: de las actividades realizadas en el salón de clases.
- Cuestionarios: para verificar el seguimiento o aprovechamiento del alumnado.

Sumativa

Análisis de los resultados obtenidos para determinar si se alcanzaron los niveles de aprendizajes previstos en el programa de estudios, que permitan asignar una calificación.

- Listas de verificación: para evaluar los trabajos extra clase.
- Rúbrica: para evaluar las investigaciones o exposiciones realizadas.
- Exámenes.

Actitudinal (AR)

- Valoración de las actitudes y valores desarrollados por el alumnado a lo largo del desarrollo de los aprendizajes de la unidad.
- Empatía: el alumnado comprende las necesidades y perspectivas de las personas con quién interactúa.
- Respeto a la diversidad: el alumnado reconoce y valora la diversidad de género, cultura, habilidades y opiniones de los demás.
- Colaboración y trabajo en equipo: el alumnado aprendió a comunicarse de manera efectiva, resolver conflictos y contribuir al éxito del equipo.
- Creatividad e innovación: el alumnado desarrolló soluciones originales a los problemas propuestos por el profesorado y muestra una predisposición positiva a la innovación.
- Tolerancia a la frustración: el alumnado demuestra una actitud positiva al enfrentarse a desafíos técnicos. Aprende de los errores y sigue mejorando.

Nota: la ponderación de cada actividad será asignada por el profesorado.

Software auxiliar

BlueJ. <https://www.bluej.org/>

Replit. <https://replit.com/>

Eclipse. <https://www.eclipse.org/>

NetBeans. <https://netbeans.apache.org/>

Referencias

Para el alumnado

- Ávila, J. y Baltazar, J. M. (2023). *Estructuras condicionales*. Portal Académico del CCH, UNAM.
<https://portalacademico.cch.unam.mx/cibernetica2/estructuras-condicionales>
- Ávila, J. (2022). *Sintaxis y sentencias*. En *Lenguaje de programación orientado a objetos con Java*. Portal Académico del CCH, UNAM. <https://portalacademico.cch.unam.mx/cibernetica2/programacion-java/sintaxis-sentencias>
- Absolute (Director). (2020a). *La Lógica de Herencia explicada con Minecraft* (Vol. 3) [Video]. <https://youtu.be/yh8bTKqCotU&t=37>
- Absolute (Director). (2020b). *La Lógica de Polimorfismo explicada con Minecraft* (Vol. 2) [Video]. <https://www.youtube.com/watch?v=bblFTvuk4pY>
- Blasco, F. (2019). *Programación orientada a objetos en Java*. Ediciones de la U. <http://www.ebooks7-24.com.pbidi.unam.mx:8080/?il=10026>
- Joyanes, A. L., & Zahonero, M. I. (2014). *Programación en c, c++, java y UML*. (2a. ed.). McGraw-Hill Interamericana.
<https://ebookcentral.proquest.com/lib/bibliodgbmhe/detail.action?docID=3225314>
- Malik, D. S. (2013). *Programación java del análisis de problemas al diseño de programas* (5a. ed.). CENGAGE Learning.
<https://ebookcentral.proquest.com/lib/bibliodgbsp/detail.action?docID=3430513>
- Olsson, M. (2013). *Java quick syntax reference*. Berkeley, Apress.
- Sznaidleder, P. (2020). *Java a Fondo*. (4a. ed.). Alfaomega.
<https://fadunam-bibliotecasdigitales-com.pbidi.unam.mx:2443/read/9786075386102/index?page=163>
- Wu, C. (2008). *Programación en Java*. McGraw Hill.

Para el profesorado

- Eckel, B. (2007). *Piensa en Java*. Pearson–Prentice Hall.
- Chávez, C. P. (2017). *Informática 2*. Pearson Hispanoamérica
<https://bookshelf-ref.vitalsource.com/books/9786073239073>
- Chiesa, G. (s/f). *Programación Orientada a Objetos con Java* [MOOC]. Coursera.
<https://www.coursera.org/learn/programacion-orientada-a-objetos-con-java>
- García, L., et al. (2003). *Construcción lógica de programas. Teoría y problemas resueltos*. Alfaomega.
- Gutiérrez, J & Ávila Nicolás, S. (s.f.) *Paquete didáctico para la Asignatura de Cibernética y Computación II*.
<http://www.cch-oriente.unam.mx/pdf/materiales/paqueteCyCII.pdf>
- Joyanes, L. (1990). *Problemas de metodología de la programación, 468 problemas resueltos*. Mc Graw–Hill.
- Prieto, N., Marqués, F., & Llorens, M. (2013). *Empezar a programar usando java* (2a. ed.). Editorial de la Universidad Politécnica de Valencia. <https://ebookcentral.proquest.com/lib/bibliodgbsp/detail.action?docID=3217647>

UNIDAD 3. HERENCIA Y POLIMORFISMO

Presentación de la unidad

El tema de estos dos pilares importantes dentro de la programación orientada a objetos se pospuso hasta esta unidad, esperando que el alumnado fuera aprendiendo los conceptos básicos, para lograr reutilizar los objetos creados y darle un mayor dinamismo y simplificación al código a través de la implementación de la herencia y el polimorfismo.

Carta descriptiva

Propósito	Tiempo
<p>Al finalizar la unidad, el alumnado:</p> <p>Implementará programas en Java utilizando herencia, polimorfismo, y colaboración de clases para aprovechar las bondades de la programación orientada a objetos como mecanismos de reutilización de código y especialización de clases.</p>	14 hrs.

Aprendizajes	Temática	Estrategias sugeridas
<p>El alumnado:</p> <p>Conoce el concepto de herencia como mecanismo de especialización y reutilización de código.</p>	<p>Herencia y jerarquía de objetos: clase, superclase y subclase (diferencias de <i>super</i> y <i>this</i>).</p> <ul style="list-style-type: none"> Agregación de atributos y métodos en clases descendientes. 	<p>Apertura Profesorado: explica el concepto de polimorfismo y su implementación.</p> <p>Desarrollo Alumnado: en equipo resuelven un problema planteado por el profesorado, tomando en cuenta las características del polimorfismo y constructores.</p> <p>Cierre Profesorado y alumnado: en plenaria, los equipos presentan y explican la forma de cómo resolvieron el problema.</p> <p>Extra clase Alumnado: plantean y resuelven un problema que requiera la inicialización de atributos (constructor).</p> <p>Apertura Profesorado y alumnado: Revisar la clase <code>Object</code> y los wrappers de Java: <code>boolean</code>, <code>float</code>, <code>double</code>, <code>integer</code> en la documentación en línea de Java y señala las diferencias con los tipos primitivos. (Las clases wrapper y su diferencia con los tipos primitivos).</p>

Aprendizajes	Temática	Estrategias sugeridas
		<p>Desarrollo</p> <p>Profesorado y alumnado: en plenaria se revisan conceptos básicos para identificar elementos de clases y de instancia, atributos y métodos; se revisa la congruencia en la nomenclatura (mayúsculas, minúsculas, capitalización); armar un árbol de clases de acuerdo con la jerarquía considerando los elementos nuevos y comunes. Introducir los símbolos UML para herencia.</p> <p>Profesorado y alumnado (en equipo o individual): realizan la escritura de varios objetos realizados en clase y de tipos de datos primitivos, por ejemplo, <code>System.out.println(new Integer(5));</code>; aplican algo parecido para las clases realizadas a lo largo del semestre.</p> <p>Profesorado: (opcionalmente) agregar un arreglo polimórfico de <i>object</i>, realizar el recorrido de los objetos escribiéndolos a la pantalla.</p> <pre>for(Object o : miArreglo) System.out.println(o.getClass().getName()+' '+o);</pre> <p>Cierre</p> <p>Profesorado: introducir los términos: clase, objeto (instancia), superclase, subclase, clase base, super y this.</p> <p>Profesorado y alumnado: concluir la ubicación de las clases creadas previamente a lo largo del curso. ¿Por qué es posible utilizar el método <code>getClass()</code> en todos los objetos que nos son <i>object</i>?, ¿Por qué los objetos aparecen en un mismo formato que parece no proporcionar información del contenido del objeto?</p> <p>Profesorado: Proponer una jerarquía utilizando las clases previas en el curso.</p> <p>Extra clase</p> <p>Alumnado: agregar métodos o atributos a subclases a partir de las clases planteadas en el curso, por ejemplo, agregar desviación estándar y otra información estadística a clase Estadística para aprovechar los datos almacenados previamente y el valor del promedio; círculo y triángulo, agregar área o perímetro, lo que no tengan; a partir de figuras 2D, crear figuras 3D agregando nuevos atributos y cálculos como el del volumen.</p>

Aprendizajes	Temática	Estrategias sugeridas
<p>Comprende la colaboración de clases.</p>	<p>Colaboración de clases.</p>	<p>Apertura Profesorado y alumnado: revisan la actividad extra clase y aclarar dudas.</p> <p>Desarrollo Profesorado y alumnado: crean clases de prueba, utilizan arreglos polimórficos a partir de Object: Object[] arregloDeObjetos = new Object[10]; Object[0] = new Float(6.0); Object[1] = new TrianguloEquilatero(1); //... O simplemente: Object unObjeto = new TriánguloEquilatero(1);</p> <p>Enfatizar cómo cambian los comportamientos cuando se trata de una super clase y su(s) subclase(s). Introducir los símbolos UML para colaboración.</p> <p>Cierre Profesorado y alumnado: concluyen que los objetos pueden estar compuestos por atributos de datos primitivos o bien contener objetos agregados al mismo.</p> <p>Extra clase Alumnado: colocan las variables declaradas dentro del método como atributos (privados) y observar el comportamiento, dividir las acciones de creación. Asignar valores para pruebas y escritura en métodos por separado y observar que el alcance en toda la clase. Realizar la escritura de las conclusiones.</p>
<p>Comprende el concepto de polimorfismo como mecanismo de especialización y automatización.</p>	<p>Concepto de polimorfismo</p> <ul style="list-style-type: none"> • Mecanismo especialización, • Automatización. 	<p>Apertura Alumnado y profesorado: en plenaria, discuten el concepto de sobrecarga estática; se explica el término de firma de un método y su importancia en la sobrecarga (polimorfismo estático).</p> <p>Desarrollo Profesorado: selecciona un problema que involucre la creación de una clase que requiera múltiples constructores. En una clase principal se crearán objetos utilizando los diferentes constructores de la clase. En conjunto responderán las siguientes preguntas: ¿Por qué debe(n) implementarse constructores en las subclases? ¿Qué utilidad tiene sobrecargar constructores? ¿Cuál es el pilar de la POO que permite que una clase pueda contener múltiples constructores? Para crear múltiples constructores, ¿en qué se deben diferenciar? ¿Existen diferencias entre los objetos creados con los diversos constructores? ¿Cuáles? Profesorado: selecciona una clase en la cual se deba sobrecargar uno de sus métodos para que funcione de manera distinta de acuerdo con su firma del método.</p> <p>Cierre Profesorado: resalta la importancia de la sobrecarga el polimorfismo dentro de la POO.</p>

Aprendizajes	Temática	Estrategias sugeridas
	<p>Polimorfismo o sobreescritura, métodos con firmas iguales en distintas clases de una jerarquía y uso de <i>super</i> y <i>this</i> en el contexto de la sobre carga dinámica.</p>	<p>Apertura Alumnado: en plenaria revisa la actividad extra clase.</p> <p>Desarrollo Profesorado: revisa el ejemplo de escritura de objetos: <pre>for(Object o : miArreglo) System.out.println(o.getClass().getName() +' '+o);</pre> Revisa que en la práctica escribir un objeto puede hacerse también con <code>o.toString()</code> Revisa qué clases implementan <code>toString()</code> (<i>Float, Double, Integer, ...</i>) Implementa en un par de clases desarrolladas en el semestre el método <code>toString()</code> y vea el comportamiento de la escritura, ¿se sobrescribe? A este mecanismo se conoce como polimorfismo (dinámico).</p> <p>Cierre Alumnado y profesorado: codifican en el resto de las clases su método Implementación del método <code>toString()</code>.</p> <p>Extra clase Alumnado: implementa otros métodos como redefinición de otros métodos: <code>getArea()</code> en <code>Óvalo</code>, <code>getInterpretaciónIMC()</code> en <code>Persona</code> y subclases.</p>
	<p>Clases de interfaz y de implementación (métodos abstractos).</p>	<p>Apertura Alumnado y profesorado: revisan los ejercicios realizados en extra clase.</p> <p>Desarrollo Profesorado y alumnado: crean o recuperan, de los ejercicios previos, una Función $f(x)$ para tabular o para calcular una integral de Riemann o la derivada por aproximaciones. Establecer qué debe modificarse ($f(x)$) para poder realizarse las mismas acciones con otra función. Determinan que si se redefine $f(x)$ el algoritmo empleado es el mismo, implementa $f(x)$ como un método abstracto para forzar a las subclases a implementarla. Codifica la implementación como una clase aparte, como una clase contenida dentro de la clase principal y como una clase anónima embebida dentro del código.</p> <p>Cierre Profesorado y alumnado: agregan otro método a la clase base y que haga uso de $f(x)$. Observar cómo las clases derivadas utilizan las relaciones de polimorfismo.</p> <p>Extra clase Alumnado: implementa subclases para redefinir la descripción del índice de masa corporal de adultos a estudiantes de 15 años, hombres y mujeres para implementarlo en otras poblaciones.</p>

Aprendizajes	Temática	Estrategias sugeridas
	Interpretación y modelado de una jerarquía de objetos.	<p>Apertura Profesorado y alumnado: Revisar los ejercicios realizados en extra clase, aclarar dudas y generar conclusiones.</p> <p>Desarrollo Profesorado y alumnado en equipos: realizar la jerarquía de clases (desde Object) de todas las clases desarrolladas en el curso.</p> <p>Cierre Profesorado y alumnado: realizar conclusiones de qué principios de la programación orientada a objetos se implementan. Indicar en qué casos hay sobrecarga y sobrescritura (polimorfismo), qué diferencias hay entre usar this o super y a qué método se llama en cada clase dependiendo de la jerarquía.</p>

Evaluación

En las estrategias sugeridas para cada sesión se proponen actividades que brindan evidencias del nivel del logro de los aprendizajes planteados. La evaluación de cada sesión se integra en la evaluación sumativa. Para evaluar los productos obtenidos en cada sesión se sugieren los siguientes tipos e instrumentos:

Diagnóstica

Conocer las características del grupo para trazar la ruta de implementación.

- Preguntas formuladas al inicio de la unidad y de cada clase.
- Análisis de las respuestas dadas en los trabajos extra clase.
- Cuestionarios o actividades interactivas del portal académico.
- Formativa.
- Evaluación para la reorientación para el logro de los aprendizajes.
- Pruebas de ejecución: codificación del programa.
- Observación: de las actividades realizadas en el salón de clases.
- Cuestionarios: para verificar el seguimiento o aprovechamiento del alumnado.

Sumativa

Análisis de los resultados obtenidos para determinar si se alcanzaron los niveles de aprendizajes previstos en el Programa de Estudios, que permitan asignar una calificación.

- Listas de verificación: para evaluar los trabajos extra clase.
- Rúbrica: para evaluar las investigaciones o exposiciones realizadas.
- Exámenes.

Actitudinal (AR)

- Valoración de las actitudes y valores desarrollados por el alumnado a lo largo del desarrollo de los aprendizajes de la unidad.
- Empatía: el alumnado comprende las necesidades y perspectivas de las personas con quien interactúa.
- Respeto a la diversidad: el alumnado reconoce y valora la diversidad de género, cultura, habilidades y opiniones de los demás.
- Colaboración y trabajo en equipo: el alumnado aprendió a comunicarse de manera efectiva, resolver conflictos y contribuir al éxito del equipo.
- Creatividad e innovación: el alumnado desarrolló soluciones originales a los problemas propuestos por el profesorado y muestra una predisposición positiva a la innovación.
- Tolerancia a la frustración: el alumnado demuestra una actitud positiva al enfrentarse a desafíos técnicos. Aprende de los errores y sigue mejorando.

Nota: la ponderación de cada actividad será asignada por el profesorado.

Software auxiliar

- BlueJ. <https://www.bluej.org/>
- Replit. <https://replit.com/>
- Eclipse. <https://www.eclipse.org/>
- NetBeans. <https://netbeans.apache.org/>

Referencias

Para el alumnado

Básicas

- Deitel P. y Deitel H. (2016). *Cómo programar en Java*. 10a. edición. Pearson Educación. México.
- Eckel, G. (2022). *Piensa en Java* (5a. ed.). Prentice Hall.
- Gutiérrez, J & Ávila Nicolás, S. (s.f.) *Paquete didáctico para la Asignatura de Cibernética y Computación II*.
<http://www.cch-oriente.unam.mx/pdf/materiales/paqueteCyCII.pdf>
- Joyanes, L. (2003). *Fundamentos de la programación. Algoritmos, estructura de datos y objetos*, Mc. Graw–Hill.
- Joyanes, L. (1990). *Problemas de metodología de la programación, 468 problemas resueltos*. Mc Graw–Hill.
- Martínez, Jorge (2015). *Fundamentos de programación en Java*. Universidad Complutense de Madrid.
<https://www.ingenieriasystems.com/2015/12/Fundamentos-de-programacion-en-Java-Autor-Jorge-Martinez-Ladron-de-Guevara.html>

Complementarias

- Absolute (Director). (2020a). *La Lógica de Herencia explicada con Minecraft* (Vol. 3) [Video]. <https://youtu.be/yh8bTKqCotU&t=37>
- Absolute (Director). (2020b). *La Lógica de Polimorfismo explicada con Minecraft* (Vol. 2) [Video].
https://www.youtube.com/watch?v=I848HdWjLMo&ab_channel=Absolute
- Ávila, J., & Ruiz, E. (2022). *Lenguaje de programación orientado a objetos con Java*. Portal Académico del CCH, UNAM.
<https://portalacademico.cch.unam.mx/cibernetica2/programacion-java>
- Ávila, J., & Ruiz, E. (2022). *Principios de la Programación Orientada a Objetos*. Portal Académico del CCH.
<https://portalacademico.cch.unam.mx/cibernetica2/principios-programacion-orientada-a-objetos>
- Berzal Galiano, F. (s.f). *Programación en Java: Fundamentos de programación y principios de diseño*. <http://elvex.ugr.es/decsai/java/>
- Chiesa, G. (s/f). *Programación Orientada a Objetos con Java*. Coursera.
<https://www.coursera.org/learn/programacion-orientada-a-objetos-con-java>

- Moya, Ricardo. (2014). "Herencia en Java, ejemplo Selección española de futbol". *Jarroba*.
<https://jarroba.com/herencia-en-la-programacion-orientada-a-objetos-ejemplo-en-java/>
- Taboada Sánchez, A. Programación A. (2017). 74. *Programación en Java || POO || Herencia en POO* [Video]. https://youtu.be/LZPafq9Ve_Q

Para el profesorado

Básicas

- Chiesa, G. (s/f). *Programación Orientada a Objetos con Java* [MOOC]. Coursera.
<https://www.coursera.org/learn/programacion-orientada-a-objetos-con-java>
- Deitel P. y Deitel H. (2016). *Cómo programar en Java*. 10a. edición. Pearson Education.
- Diagramando UML con PlantUML. *Guía de referencia del lenguaje*. (2017).
http://s.plantuml.com/PlantUML_Language_Reference_Guide_ES.pdf
- Eckel, G. (2022). *Piensa en Java* (5a. ed.). Prentice Hall.
- Gutiérrez, J & Ávila Nicolás, S. (s.f.) *Paquete didáctico para la Asignatura de Cibernética y Computación II*. <http://www.cch-orientado.unam.mx/pdf/materiales/paqueteCyCII.pdf>
- Oracle. (2017). *Lesson: Object-Oriented Programming Concepts*.
<https://docs.oracle.com/javase/tutorial/java/concepts/index.html>

Complementarias

- Amhed, A. y Prasad, B. (2016). *Foundations of Software Design*. CRC Press.
- Ávila, J., & Ruiz, E. (2022). *Lenguaje de programación orientado a objetos con Java*. Portal Académico del CCH, UNAM.
<https://portalacademico.cch.unam.mx/cibernetica2/programacion-java>
- Ávila, J., & Ruiz, E. (2022). *Principios de la Programación Orientada a Objetos*. Portal Académico del CCH.
<https://portalacademico.cch.unam.mx/cibernetica2/principios-programacion-orientada-a-objetos>
- Barnes, D.J. y Kölling, M. (2007). *Programación orientada a objetos con Java, una introducción práctica usando BlueJ*. Pearson Educación.
- Coad, P. y Yourdon, E. (1991). *Object Oriented Analysis*. Yourdon Press.
- Deitel P. y Deitel H. (2016). *Cómo programar en Java*. 10a. edición. Pearson Education. Capítulos 20-34.
https://www.pearsonenespanol.com/mexico/educacion-superior/deitel_index/como-programar-en-java-10e

- Hernández, A. y Ávila, J. (2013). *Informe del seminario institucional para la elaboración de material didáctico interactivo de Cibernética y Computación II*. CCH-UNAM.
- Jiménez, C. (2015). *UML: Aplicaciones en Java y C++*.
https://cch-oriente.bibliotecas.unam.mx:81/cgi-bin/koha/opac-detail.pl?biblionumber=29481&query_desc=kw%2Cwrdl%3A%20programación%20orientada%20a%20objetos
- Oviedo, E.M. (2015). *Lógica de programación orientada a objetos*. Ecoe Ediciones. https://cch-oriente.bibliotecas.unam.mx:81/cgi-bin/koha/opacdetail.pl?biblionumber=26262&query_desc=su%3A%22Programación%20orientada%20a%20objetos%20Computación%22
- Reyes Gómez, J. P., & Hernández Hoyos, M. (s/f). *Introducción a la programación orientada a objetos en Java—Módulo 4: Herencia* [MOOC]. Coursera.
<https://www.coursera.org/learn/introduccion-programacion-java/home/week/4>
- Villalobos S., J. A. (2008). *Mecanismos de reutilización y desacoplamiento. Introducción a las estructuras de datos. Aprendizaje activo basado en casos*. (pp. 221–282). Pearson Prentice Hall.
- Weiss, M. A. (2013). *Estructuras de datos en JAVA*, 4ª edición. Pearson Educación.
<https://bookshelf-ref.vitalsource.com/>

UNIDAD 4. INTERFAZ GRÁFICA DE USUARIO

Presentación de la unidad

En virtud de que el software usado por el alumnado está compuesto de ventanas, íconos, barras de herramientas, menús, cuadros de diálogos, es importante que cambie de un modo consola a un modo gráfico. Lo programado anteriormente en modo consola lo transformará a través de la creación de objetos del paquete Swing para obtener un ambiente gráfico.

Carta descriptiva

Propósito	Tiempo
<p>Al finalizar la unidad, el alumnado:</p> <p>Implementará programas con colaboración de clases en Java para aplicar sus conocimientos de programación orientada a objetos, utilizando interfaces gráficas de usuario con formularios y su comparación con las interfaces de consola.</p>	16 hrs.

Aprendizajes	Temática	Estrategias sugeridas
<p>El alumnado:</p> <p>Aplicará los mecanismos de colaboración entre clases para implementar interfaces de usuario (presentación) con las clases de la lógica (modelado).</p>	<p>Interfaces de usuario: consola e interfaces formularios.</p> <p>Diagramas de secuencia (usuario, programa principal y clases de lógica).</p>	<p>Apertura Profesorado: explica el concepto de interfaz de usuario y los tipos de interfaces (consola y formulario).</p> <p>Desarrollo Profesorado: plantea el diagrama de flujo de un problema sencillo donde el usuario interactúe con el programa principal y este con sus clases. Alumnado: planteará el esquema del diagrama de secuencia para la interfaz tipo consola, después planteará el esquema del diagrama de secuencia para la interfaz tipo formulario para observar las diferencias y las necesidades para crear un formulario.</p> <p>Cierre Profesorado: explicará los elementos fundamentales de los diagramas de secuencia para ambas interfaces.</p> <p>Extra clase Alumnado: revisan (en modo consola) qué instrucciones involucran interacción en el diagrama de secuencia y/o de flujo una interacción entre el programa principal y el usuario, y entre el programa principal y las clases de la lógica. Preparan conclusiones para la siguiente sesión.</p>

Aprendizajes	Temática	Estrategias sugeridas
	<p>Diferencias entre mecanismos de interacción en las interfaces de consola y formularios.</p> <ul style="list-style-type: none"> • Sistema de coordenadas en pantalla. 	<p>Apertura Alumnado y profesorado: concluyen que las sentencias de lectura y escritura son interacciones del programa principal con el usuario, y los cálculos se quedan entre la lógica y el programa principal. Asociar que por cada instrucción de lectura (en modo consola) se requiere escribir algo en pantalla y leer un dato. Detallar el mecanismo de declaraciones, instanciación y uso de métodos de un objeto de la clase <i>Scanner</i> para tener acceso a los datos proporcionados por el usuario mediante <i>System.in</i> y <i>System.out</i></p> <p>Desarrollo Profesorado: muestra algunos formularios de ejemplo. Profesorado y alumnados: identifican los tipos de componentes de los formularios e indican qué tipo de información pueden recuperar (cadenas y booleanos). Analizan qué elementos comunes y distintivos hay entre cada componente en una interfaz gráfica de usuario. Alumnado: realizan una investigación para ver los mecanismos de conversión entre cadenas y otros tipos de datos. Profesorado: realiza un resumen de acciones que pueden realizarse con las cadenas para realizar conversiones de valores primitivos y arreglos.</p> <p>Cierre Alumnado y profesorado: realizan ejemplos de conversiones de datos y revisan qué tipo de excepciones pueden generarse. Identifican que un <i>get</i> entre el usuario y el programa principal se traduce en un setter con la clase de la lógica y viceversa.</p> <p>Extra clase Alumnado: realiza ejercicios de conversión entre datos primitivos y cadenas; hacen un listado de tipos de componentes que pueden encontrarse en una interfaz de usuario gráfica, características comunes y distintivas de cada componente.</p>

Aprendizajes	Temática	Estrategias sugeridas
--------------	----------	-----------------------

Aplicará los conceptos de herencia para modelar e interpretar la jerarquía de clases de AWT/Swing para la construcción de interfaces gráficas de usuario (GUI).

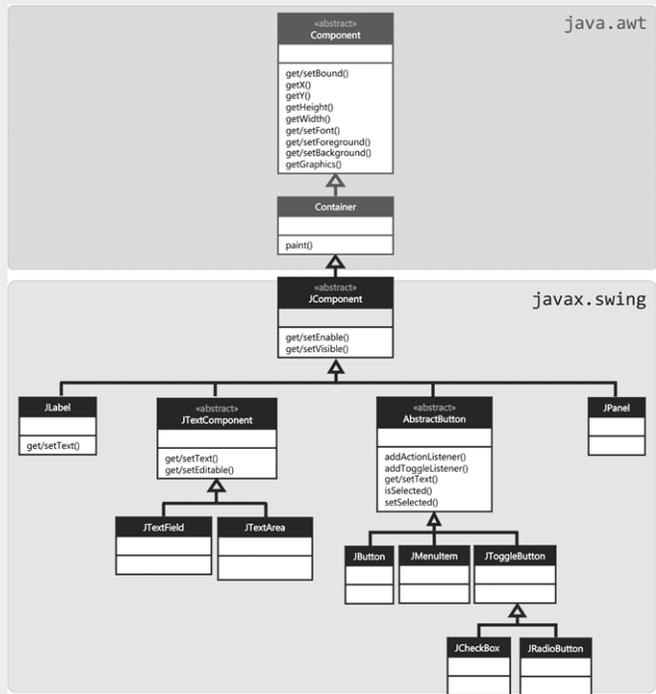
Jerarquía de clases de Swing.

Apertura

Alumnado y profesorado: realizan una revisión de la actividad extra clase para identificar los tipos de componentes y proponer una jerarquía de clases que los describa.

Desarrollo

Profesorado: introduce el siguiente diagrama y en plenaria ubican los elementos comunes y distintivos.



Profesorado y alumnado: realizan una revisión rápida de la documentación para ver otros métodos disponibles en cada clase.

Aprendizajes	Temática	Estrategias sugeridas
<p>Elabora programas con interfaz gráfica que capture y presente texto con JLabel, JTextField y JTextArea.</p>	<p>Contenedores: <i>JFrame</i> y <i>JPanel</i>.</p> <p>Captura y presentación de texto:</p> <ul style="list-style-type: none"> • JLabel. • JTextField. • JTextArea. 	<p>Apertura</p> <p>Alumnado: guiados por el profesorado hacen una revisión conceptual de las interacciones GUI usando clases AWT/Swing de <i>JFrame</i>, <i>JPanel</i>, <i>JLabel</i>, <i>JTextField</i> y <i>JTextArea</i> utilizando la documentación oficial de Java que se encuentra en línea.</p> <p>Desarrollo</p> <p>Profesorado: explica en plenaria los conceptos clave: <i>JFrame</i> como contenedor principal, <i>JPanel</i> como subcontenedor, <i>JLabel</i> como herramienta para mostrar texto o imágenes de forma estática, <i>JTextField</i> como forma de entrada de texto y <i>JTextArea</i> como una entrada de texto multi renglón editable.</p> <p>Profesorado: desarrolla un ejemplo sencillo donde se crea una ventana principal mediante un <i>JFrame</i> y además crea un contenedor mediante el uso de <i>JPanel</i>, posteriormente agrega elementos <i>JLabel</i>, <i>JTextFeld</i> y <i>JTextArea</i>.</p> <p>Alumnado: en equipo el alumnado desarrolla pequeños programas haciendo énfasis en usar su creatividad para hacer variaciones del ejemplo desarrollado por el profesorado.</p> <p>Cierre</p> <p>Alumnado: exponen por equipo el ejemplo que desarrollaron.</p> <p>Extra clase</p> <p>Alumnado: de forma individual realicen otro programa con la finalidad de afianzar los conocimientos vistos en esta sesión, en caso de dudas el profesorado podrá tener una retroalimentación con ellos mediante la plataforma de comunicación que h aya establecido (Teams, Classroom, Telegram, WhatsApp, etcétera).</p>

Aprendizajes	Temática	Estrategias sugeridas
<p>Elabora programas con interfaz gráfica que realice acciones a través de <i>JButton</i>.</p>	<p>Ejecución de acciones por medio de botones:</p> <ul style="list-style-type: none"> ▪ <i>JButton</i>. <p>Manejo de eventos</p> <ul style="list-style-type: none"> ▪ <i>ActionListener</i>, ▪ <i>KeyListener</i>. 	<p>Apertura Profesorado y alumnado: en plenaria discuten los eventos y los escuchadores (<i>listener</i>), los tipos y su utilidad en las interfaces de usuario en Java.</p> <p>Desarrollo Profesorado: propone un ejemplo donde se tenga que accionar un evento que involucre hacer clic en un botón, mediante un <i>ActionListener</i>. Se debe permitir captar el evento y realizar una acción, por ejemplo, al presionar un botón de la interfaz, se tiene que cambiar el color del fondo del <i>JPanel</i>. Profesorado: propone un ejemplo donde se tenga que presionar una tecla y mediante un <i>KeyListener</i>, se debe permitir captar el evento y realizar una acción, por ejemplo, al presionar una tecla, se tiene que visualizar en un <i>JLabel</i> todas las letras presionadas por el usuario. Alumnado: se organizan en equipos y resuelven diferentes ejercicios que involucren el uso de eventos y escuchadores.</p> <p>Cierre Alumnado: exponen sus resultados al resto del grupo. En plenaria se comenta la importancia de las acciones y escuchadores dentro de las interfaces de usuario en Java y genera sus conclusiones.</p> <p>Extra clase: Profesorado: propone un conjunto de ejercicios para reforzar los conocimientos que involucren acciones y escuchadores dentro de una Interfaz gráfica de usuario.</p>
<p>Elabora programas con interfaz gráfica aplicando las clases: <i>JCheckBox</i>, <i>JRadioButton</i>.</p>	<p>Ejecución de selección de datos a través de:</p> <ul style="list-style-type: none"> ▪ <i>JCheckBox</i>, ▪ <i>JRadioButton</i>. 	<p>Apertura Alumnado y profesorado: en plenaria se discuten los conceptos básicos, funcionalidad y características de elementos interactivos en Java, <i>JCheckBox</i>, <i>JButton</i> y <i>JRadioButton</i>.</p> <p>Desarrollo Profesorado: de manera guiada, apoya al alumnado a codificar un ejemplo de interfaz que contenga tres secciones. Selección de alternativas con <i>JCheckBox</i>, selección de una alternativa entre varias con <i>JRadioButton</i> y una última sección de procesamiento o confirmación con <i>JButton</i>. Se utiliza el método <i>addActionListener()</i> y <i>isSelected()</i> de <i>JRadioButton</i> para detectar las acciones del usuario.</p> <p>Cierre Alumnado: discute en plenaria los usos, ventajas y desventajas de los tres elementos.</p> <p>Extraclase Alumnado: plantea y elabora un proyecto con interfaz gráfica que integre las clases vistas en esta sesión más las estudiadas anteriormente.</p>

Aprendizajes	Temática	Estrategias sugeridas
Elabora programas con interfaz gráfica aplicando las clases: <i>JMenuBar</i> , <i>JMenu</i> , <i>JMenuItem</i> .	Clase <i>Swing</i> : <ul style="list-style-type: none"> ▪ <i>JMenuBar</i>. ▪ <i>JMenu</i>. ▪ <i>JMenuItem</i>. 	<p>Apertura Profesorado: expone el uso de las clases <i>JMenuBar</i>, <i>JMenu</i>, <i>JMenuItem</i>, para crear programas con interfaz gráfica de usuario.</p> <p>Desarrollo Alumnado: organizados en equipo resuelven problemas, sugeridos por el profesorado, empleando las clases expuestas.</p> <p>Cierre Alumnado: ejecutan los programas para verificar los resultados.</p>

Evaluación

En las estrategias sugeridas para cada sesión se proponen actividades que brindan evidencias del nivel del logro de los aprendizajes planteados. La evaluación de cada sesión se integra en la evaluación sumativa. Para evaluar los productos obtenidos en cada sesión se sugieren los siguientes tipos e instrumentos:

Diagnóstica

Conocer las características del grupo para trazar la ruta de implementación.

- Preguntas formuladas al inicio de la unidad y de cada clase
- Análisis de las respuestas dadas en los trabajos extra clase.
- Cuestionarios o actividades interactivas del portal académico.

Formativa

Evaluación para la reorientación para el logro de los aprendizajes.

- Pruebas de ejecución: codificación del programa.
- Observación: de las actividades realizadas en el salón de clases.
- Cuestionarios: para verificar el seguimiento o aprovechamiento del alumnado.

Sumativa

Análisis de los resultados obtenidos para determinar si se alcanzaron los niveles de aprendizajes previstos en el programa de estudios, que permitan asignar una calificación.

- Listas de verificación: para evaluar los trabajos extra-clase.
- Rúbrica: para evaluar las investigaciones o exposiciones realizadas.
- Exámenes.

Actitudinal (AR)

- Valoración de las actitudes y valores desarrollados por el alumnado a lo largo del desarrollo de los aprendizajes de la unidad.
- Empatía: el alumnado comprende las necesidades y perspectivas de las personas con quién interactúa.
- Respeto a la diversidad: el alumnado reconoce y valora la diversidad de género, cultura, habilidades y opiniones de los demás.
- Colaboración y trabajo en equipo: el alumnado aprendió a comunicarse de manera efectiva, resolver conflictos y contribuir al éxito del equipo.
- Creatividad e innovación: el alumnado desarrolló soluciones originales a los problemas propuestos por el profesorado y muestra una predisposición positiva a la innovación.
- Tolerancia a la frustración: el alumnado demuestra una actitud positiva al enfrentarse a desafíos técnicos. Aprende de los errores y sigue mejorando.

Nota: La ponderación de cada actividad será asignada por el profesorado.

Software auxiliar

- BlueJ. <https://www.bluej.org/>
- Replit. <https://replit.com/>
- Eclipse. <https://www.eclipse.org/>
- NetBeans. <https://netbeans.apache.org/>

Referencias

Para el alumnado

Básicas

- Deitel P. y Deitel H. (2016). *Cómo programar en Java*. 10a. edición. Pearson Education.
- Eckel, G. (2022). *Piensa en Java* (5a. ed.). Prentice Hall.
- Joyanes, L. (2003). *Fundamentos de la programación. Algoritmos, estructura de datos y objetos*. Mc. Graw–Hill
- Gutiérrez, J & Ávila Nicolás, S. (s.f.) *Paquete didáctico para la Asignatura de Cibernética y Computación II*.
<http://www.cch-oriente.unam.mx/pdf/materiales/paqueteCyCII.pdf>
- Joyanes, L. (1990). *Problemas de metodología de la programación, 468 problemas resueltos*. México: Mc Graw–Hill.
- Martínez, Jorge (2015). *Fundamentos de programación en Java*. Universidad Complutense de Madrid.
<https://www.ingenieriasystems.com/2015/12/Fundamentos-de-programacion-en-Java-Autor-Jorge-Martinez-Ladron-de-Guevara.html>

Complementarias

- Absolute (Director). (2020a). *La Lógica de Herencia explicada con Minecraft* (Vol. 3) [Video]. <https://youtu.be/yh8bTKqCotU&t=37>
- Absolute (Director). (2020b). *La Lógica de Polimorfismo explicada con Minecraft* (Vol. 2) [Video]. <https://youtu.be/yh8bTKqCotU>
- Ávila, J., & Ruiz, E. (2022). *Lenguaje de programación orientado a objetos con Java*. Portal Académico del CCH, UNAM.
<https://portalacademico.cch.unam.mx/cibernetica2/programacion-java>
- Ávila, J., & Ruiz, E. (2022). *Principios de la Programación Orientada a Objetos*. Portal Académico del CCH.
<https://portalacademico.cch.unam.mx/cibernetica2/principios-programacion-orientada-a-objetos>
- Berzal Galiano, F. (s.f). *Programación en Java: Fundamentos de programación y principios de diseño*.
<http://elvex.ugr.es/decsai/java/>
- Chiesa, G. (s/f). *Programación Orientada a Objetos con Java* [MOOC]. Coursera. <https://www.coursera.org/learn/programacion-orientada-a-objetos-con-java>

- Moya, Ricardo. (2014). "Herencia en Java, ejemplo Selección española de futbol". *Jarroba*.
<https://jarroba.com/herencia-en-la-programacion-orientada-a-objetos-ejemplo-en-java/>
- Taboada Sánchez, A. *Programación A*. (2017). 74. *Programación en Java || POO || Herencia en POO* [Video]. https://youtu.be/LZPafq9Ve_Q

Para el profesorado

Básicas

- Chiesa, G. (s/f). *Programación Orientada a Objetos con Java [MOOC]*. Coursera.
<https://www.coursera.org/learn/programacion-orientada-a-objetos-con-java>
- Deitel P. y Deitel H. (2016). *Cómo programar en Java*. 10a. edición. Pearson Education.
- Eckel, G. (2022). *Piensa en Java* (5a. ed.). Prentice Hall.
- Diagramando UML con PlantUML. Guía de referencia del lenguaje. (2017). http://s.plantuml.com/PlantUML_Language_Reference_Guide_ES.pdf
- Gutiérrez, J & Ávila Nicolás, S. (s.f.) *Paquete didáctico para la Asignatura de Cibernética y Computación II*.
<http://www.cch-orientado.unam.mx/pdf/materiales/paqueteCyCII.pdf>
- Joyanes, L. (2003). *Fundamentos de la programación. Algoritmos, estructura de datos y objetos*. Mc. Gra-Hill
- Joyanes, L. (1990). *Problemas de metodología de la programación, 468 problemas resueltos*. Mc. Graw-Hill.
- Oracle. (2017). *Lesson: Object-Oriented Programming Concepts*.
<https://docs.oracle.com/javase/tutorial/java/concepts/index.html>

Complementarias

- Amhed, A. y Prasad, B. (2016). *Foundations of Software Design*. CRC Press: EEUU.
- Ávila, J., & Ruiz, E. (2022). *Lenguaje de programación orientado a objetos con Java*. *Portal Académico del CCH, UNAM*. <https://portalacademico.cch.unam.mx/cibernetica2/programacion-java>
- Ávila, J., & Ruiz, E. (2022). *Principios de la Programación Orientada a Objetos*. *Portal Académico del CCH*.
<https://portalacademico.cch.unam.mx/cibernetica2/principios-programacion-orientada-a-objetos>
- Barnes, D.J. y Kölling, M. (2007). *Programación orientada a objetos con Java, una introducción práctica usando BlueJ*. Pearson Educación: Madrid.
- Coad, P. y Yourdon, E. (1991). *Object Oriented Analysis*. Yourdon Press.

- Deitel P. y Deitel H. (2016). *Como programar en Java*. 10ma. edición. Pearson Education. Capítulos 20-34.
https://www.pearsonenespanol.com/mexico/educacion-superior/deitel_index/como-programar-en-java
- Hernández, A. y Ávila, J. (2013). *Informe del seminario institucional para la elaboración de material didáctico interactivo de Cibernética y Computación II*. ENCCH-UNAM.
- Jiménez, C. (2015). *UML: Aplicaciones en Java y C++*.
https://cch-oriente.bibliotecas.unam.mx:81/cgi-bin/koha/opac-detail.pl?biblionumber=29481&query_desc=kw%2Cwrdl%3A%20programación%20orientada%20a%20objetos
- Oviedo, E.M. (2015). *Lógica de programación orientada a objetos*. Ecoe Ediciones. https://cch-oriente.bibliotecas.unam.mx:81/cgi-bin/koha/opacdetail.pl?biblionumber=26262&query_desc=su%3A%22Programación%20orientada%20a%20objetos%20Computación%22
- Reyes Gómez, J. P., & Hernández Hoyos, M. (s/f). *Introducción a la programación orientada a objetos en Java—Módulo 4: Herencia* [MOOC]. Coursera.
<https://www.coursera.org/learn/introduccion-programacion-java/home/week/4>
- Weiss, M. A. (2013). *Estructuras de datos en JAVA*, 4ª edición. Pearson Educación.
<https://bookshelf-ref.vitalsource.com/>
- Villalobos S., J. A. (2008). "Mecanismos de reutilización y desacoplamiento". *Introducción a las estructuras de datos. Aprendizaje activo basado en casos*. (pp. 221–282). Pearson Prentice Hall.



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Dr. Leonardo Lomelí Vanegas

RECTOR

Dra. Patricia Dolores Dávila Aranda

SECRETARIA GENERAL

Mtro. Hugo Alejandro Concha Cantú

ABOGADO GENERAL

Mtro. Tomás Humberto Rubio Pérez

SECRETARIO ADMINISTRATIVO

Dra. Diana Tamara Martínez Ruiz

SECRETARIA DE DESARROLLO INSTITUCIONAL

Lic. Raúl Arcenio Aguilar Tamayo

SECRETARIO DE PREVENCIÓN Y SEGURIDAD UNIVERSITARIA

Mtro. Néstor Martínez Cristo

DIRECTOR GENERAL DE COMUNICACIÓN SOCIAL



**ESCUELA NACIONAL
COLEGIO DE CIENCIAS Y HUMANIDADES**

Dr. Benjamín Barajas Sánchez
DIRECTOR GENERAL

Lic. Mayra Monsalvo Carmona
SECRETARIA GENERAL

Lic. Rocío Carrillo Camargo
SECRETARIA ADMINISTRATIVA

Lic. María Elena Juárez Sánchez
SECRETARIA ACADÉMICA

QBP. Taurino Marroquín Cristóbal
SECRETARIO DE SERVICIOS DE APOYO AL APRENDIZAJE

Mtra. Dulce María E. Santillán Reyes
SECRETARIA DE PLANEACIÓN

Mtro. José Alfredo Núñez Toledo
SECRETARIO ESTUDIANTIL

Mtra. Araceli Mejía Olguín
SECRETARIA DE PROGRAMAS INSTITUCIONALES

Lic. Héctor Baca Espinoza
SECRETARIO DE COMUNICACIÓN INSTITUCIONAL

Ing. Armando Rodríguez Arguijo
SECRETARIO DE INFORMÁTICA

DIRECTORES DE PLANTELES

AZCAPOTZALCO

Mtra. Martha Patricia López Abundio

NAUCALPAN

Mtro. Keshava Quintanar Cano

VALLEJO

Lic. Maricela González Delgado

ORIENTE

Mtra. María Patricia García Pavón

SUR

QFB. Susana de los Ángeles Lira de Garay



**PROGRAMAS
DE ESTUDIO
2024**

Escuela Nacional Colegio de Ciencias y Humanidades.
Los Programas de Estudio del Área de Matemáticas
se terminaron de imprimir en el mes de julio de 2024.



**PROGRAMAS
DE ESTUDIO
2024**

