



Programa de estudio de Pensamiento computacional

Autores:

Claudia De Anda Quintin
Edwin Ramón Romero Espíritu
Rigoberto Santiago Garzón

Currículo Bachillerato UAS 2024		
Bachillerato General	Modalidad Escolarizada	Opción Presencial
Programa de estudio Cultura digital II		
Clave:	-	Horas semestre 48
Semestre:	IV	Horas semana 3
Grado:	Cuarto	Créditos 6
Currículum fundamental. Recurso sociocognitivo.	Órgano que lo aprueba:	Foro Estatal Reforma de Programas de Estudio 2024
Componente de formación:	Fundamental y extendido	Vigencia: A partir de enero 2026

Mapa del Currículo del Bachillerato UAS 2024 modalidad escolarizada y opción presencial

		Semestre I	Semestre II	Semestre III	Semestre IV	Semestre V	Semestre VI
Componente fundamental y extendido obligatorio	Lengua y comunicación	Lengua y comunicación I (3,6)*	Lengua y comunicación II (3,6)	Lengua y comunicación III (4,8)	Lengua y comunicación IV (4,8)		
	Ingles I	Ingles I (3,6)	Ingles II (3,6)	Ingles III (3,6)	Ingles IV (3,6)		
Pensamiento matemático	Pensamiento matemático I (4,8)	Pensamiento matemático II (4,8)	Pensamiento matemático III (4,8)	Pensamiento matemático III (5,10)	Temas selectos de Matemáticas I (5,10)	Temas selectos de Matemáticas II (3,6)	Temas selectos de Matemáticas III (3,6)
Cultura digital	Cultura digital I (3,6)	Cultura digital II (3,6)	Cultura digital III (3,6)	Pensamiento computacional (3,6)			
Conciencia histórica		Conciencia histórica I (3,6)	Conciencia histórica II (3,6)	Conciencia histórica III (3,6)			
Clencias sociales	Laboratorio de investigación social (3,6)		Ciencias sociales I (3,6)	Ciencias Sociales II (3,6)		Economía, empresa y sociedad (3,6)	Elementos básicos de administración (3,6)
Ciencias naturales, experimentales y tecnología	La materia y sus interacciones (5,10)	Reacciones químicas (5,10)		Conservación de la energía (5,10)	La energía en los procesos de la vida diaria (5,10)		
	Organismos: estructuras y procesos (5,10)	Herencia y evolución biológica (5,10)				Ciencias de la salud (3,6)	Ecosistemas y desarrollo sostenible (3,6)
Humanidades	Humanidades I (3,6)	Humanidades II (3,6)	Humanidades III (3,6)	Humanidades IV (3,6)	Pensamiento literario I (3,6)	Pensamiento literario II (3,6)	
CA	Curriculum ampliado	Formación socioemocional I (1,2)	Formación socioemocional II (1,2)	Formación socioemocional III (1,2)	Formación socioemocional IV (1,2)		
Componente fundamental extendido opcional Fases de preparación específica	Ciencias físmico-matemáticas					Cálculo I (5,10)	Cálculo II (5,10)
						Temas selectos de Mecánica (5,10)	Propiedades de la materia (5,10)
						Electromagnetismo (5,10)	Óptica (5,10)
	Ciencias químico-biológicas					Dibujo I (3,6)	Dibujo II (3,6)
	Ciencias sociales y humanidades					Cálculo I (5,10)	Cálculo II (5,10)
						Electricidad y Óptica (5,10)	Propiedades de la materia (5,10)
						Temas selectos de Química I (5,10)	Temas selectos de Química II (5,10)
						Temas selectos de Biología I (3,6)	Temas selectos de Biología II (3,6)
	Total de horas-clase por semana y créditos	(30,60)	(30,60)	(30,60)	(30,60)	(30,60)	(30,60)

* Indica horas-clase semanales y créditos de cada UAC

Componente de formación fundamental

Componente de formación fundamental extendido (UAC obligatorias)

Componente de formación ampliada (recursos socioemocionales)

Componente de formación fundamental extendido (UAC optativas)

Servicio de apoyo educativo

El semestre consta de 16 semanas (480 hrs. de clases y 120 hrs. de estudio independiente)

Total de horas frente a docente: 2880

Total de horas de estudio independiente: 720

Total de horas de Servicio social estudiantil: 100

Total de horas: 3700

Total de créditos: 370

Curriculum ampliado (programas cocurriculares)		
Actividades físicas y deportivas (100 horas optativas y 10 créditos)	Servicio social estudiantil (100 horas y 10 créditos)	Actividades artísticas y culturales (100 horas optativas y 10 créditos)

Servicios de apoyo educativo		
Programa institucional de tutorías	Orientación Educativa	ADIUSAS

Introducción

El bachillerato de la Universidad Autónoma de Sinaloa (UAS), en las últimas dos décadas ha transitado por importantes modificaciones curriculares, con la finalidad de mejorar la calidad educativa, centrándose en el estudiante y la generación de su aprendizaje; bases del modelo constructivista.

La transición de la UAS hacia una educación focalizada en el aprendizaje, inició en el año 2009, cuando la incorpora al plan de estudio el enfoque por competencias, con el firme propósito de ingresar al Sistema Nacional de Bachillerato (SNB), posteriormente llamado Padrón de Calidad del Sistema Nacional de Educación Media Superior (PC-SiNEMS); lo que generó la necesidad de alinearla al Marco Curricular Común (MCC) derivado de la Reforma Integral de la Educación Media Superior (RIEMS), impulsada por el gobierno federal mexicano.

En el 2015, se modificaron el plan y programas de estudio del bachillerato universitario, para estar en condiciones de dar cumplimiento a lo establecido en el Acuerdo secretarial 656 que reforma los acuerdos 444 y 486 de la RIEMS. Posteriormente, en el plan de estudios 2018, se atendieron los lineamientos del Modelo Educativo para la Educación Obligatoria (MEPEO) el cual promueve aprendizajes claves en cada uno de los campos disciplinares, e integra los planteamientos sobre el desarrollo de habilidades socioemocionales resaltando las actitudes y valores como parte integral de las competencias.

Siguiendo esa línea, en el 2024 la Dirección General de Escuelas Preparatorias (DGEP) de la UAS, puso en marcha el nuevo diseño del Currículo del bachillerato UAS, modalidad escolarizada y opción presencial. La reestructuración de los planes y programas de estudio se orienta con los enfoques humanista y constructivista del Modelo educativo UAS 2022, con los lineamientos de la Nueva Escuela Mexicana (NEM), y por supuesto con el nuevo Marco Curricular Común de la Educación Media Superior (MCCEMS). Este último incorpora los aprendizajes de trayectoria, entendidos, como el perfil de egreso mínimo común de los estudiantes para el Nivel Medio Superior (NMS). Además, se compone por once rasgos que los estudiantes han de lograr progresivamente, pretendiendo alcanzarlos a través de las asignaturas o Unidades de Aprendizaje Curriculares (UAC), que a su vez estructuran a Recursos sociocognitivos (RS), Áreas de conocimiento (AC) y Recursos y ámbitos socioemocionales (RyASE).

Particularmente, el Recurso sociocognitivo Cultura Digital busca fomentar en los estudiantes no solo el uso de aplicaciones digitales, sino la reflexión del empleo y efectos que causan las Tecnologías de la Información, Comunicación, Conocimiento y Aprendizaje Digitales (TICCAD) en su actuar cotidiano, así como la capacidad de adaptarse a la diversidad y disponibilidad de su contexto. (SEP, 2023a).

I. Fundamentación curricular

La Universidad Autónoma de Sinaloa a través de su modelo educativo: UAS 2022, gestiona un proceso de formación centrado en el aprendizaje, potencia las habilidades y aptitudes de los estudiantes, articula los saberes con sus características e intereses para lograr la formación integral de universitarios con sentido humanista, conjuntamente busca generar estudiantes con habilidades de pensamiento crítico, creativo e innovador, donde el centro de su formación sea la comprensión de los fenómenos económicos, políticos, sociales y culturales y contribuir así al desarrollo de la sociedad a la que pertenece. (UAS, 2022).

En este mismo sentido, la NEM tiene como base la idea de promover el aprendizaje bajo ocho principios: fomentar la identidad con México, la responsabilidad ciudadana, honestidad, participación en la transformación de la sociedad, respeto por la dignidad humana, interculturalidad, cultura de la paz y respeto por la naturaleza (SEP, 2019).

Por otro lado, el MCCEMS en su Acuerdo secretarial 09-08-23, establece la base que compone: el currículum fundamental y el currículum ampliado. En el primero, se integran los recursos sociocognitivos, que son los aprendizajes articuladores base del bachillerato e indispensables para la comprensión y construcción permanente del conocimiento; posibilitando al estudiante la oportunidad de acceder a los saberes de las áreas.

Los recursos sociocognitivos Lengua y comunicación, Pensamiento matemático, Conciencia histórica y Cultura digital desempeñan un papel transversal en el currículum para el logro de los aprendizajes de trayectoria. A su vez las áreas de conocimiento sientan las bases de la formación disciplinar del currículum fundamental; constituyendo los aprendizajes de las Ciencias naturales, experimentales y tecnología, Ciencias sociales y humanidades. Desde el currículum ampliado se constituye el eje articulador para la formación social y autónoma que integra los ámbitos socioemocionales.

Cultura digital como recurso sociocognitivo, sustenta tanto al currículum fundamental y como el ampliado, apuntando a la generación de conocimiento en el que, lo digital lo potencia y fortalece. El término cultura digital, refiere al conjunto de sistemas culturales en vínculo con las tecnologías digitales; está relacionada con saberes informáticos de orden operativo e instrumental y con saberes informacionales. En este sentido, al formar estudiantes con una cultura digital se contribuye a que fortalezcan su pensamiento crítico-reflexivo y la capacidad creativa para el dar de soluciones a situaciones de la vida cotidiana, así como el desarrollo

de habilidades de búsqueda y selección de información en diversos formatos (SEPa, 2023).

En virtud de lo anterior, el Plan Bachillerato UAS 2024 incorpora en su malla curricular una signatura más a la propuesta del MCCEMS en el Recurso sociocognitivo de Cultura digital. Una asignatura que integre a la programación estructurada y la robótica educativa para transitar de un consumo tecnológico pasivo a una cultura digital activa y creativa, donde el desarrollo del pensamiento computacional y el enfoque STEAM (Modelo educativo que integra Ciencia, Tecnología, Ingeniería, Artes y Matemáticas) faciliten la resolución de problemas reales mediante la lógica y la experimentación física; dotando al estudiante no solo de competencias técnicas esenciales para la Industria 4.0, sino que actúen como ejes transversales que fortalecen el pensamiento crítico, la interdisciplinariedad y habilidades socioemocionales como la resiliencia y el trabajo colaborativo, formando así ciudadanos capaces de transformar su entorno social con herramientas tecnológicas éticas e innovadoras.

Es así como Pensamiento computacional se integra en el cuarto semestre con un programa de estudios encaminado al logro de aprendizajes de trayectoria y diseño de estrategias didácticas para que los estudiantes adquieran conocimientos y experiencias acordes a las exigencias presentes y futuras que emanan de los veloces cambios tecnológicos que transforman a la sociedad.

Los principios pedagógicos del programa alinean un enfoque educativo colaborativo y adaptable a las realidades y contextos, además promueven un aprendizaje activo y reflexivo planteado a través de las metodologías activas y participativas, basadas en la indagación y el descubrimiento de conocimientos. La propuesta didáctica del MCCEMS es el modelo 5E, aludiendo a Enganchar, Explorar, Explicar, Elaborar y Evaluar, donde cada elemento es análogo a una fase del proceso de aprendizaje. En el transcurrir del tiempo, estas fases ya han sido propuestas desde diferentes modelos educativos pero con otros términos.

La propuesta es iniciar por conocer las actitudes y percepciones de los estudiantes acerca de los temas propuestos, captando el interés del estudiante a través de la motivación. Continuar con momentos de adquisición e integración del conocimiento que puede darse mediante la indagación, la lectura y la escucha activa llevándolo a que organice la nueva información. Seguido por la extensión y refinamiento al profundizar en la comprensión del tema y establecer conexiones con otros conocimientos, procedimientos y habilidades, involucrando experiencias de aprendizaje para dar un uso significativo aplicable en situaciones reales. Finalizando con el desarrollo de habilidades metacognitivas como la planificación, la autorregulación y la evaluación del propio aprendizaje.

II. Aprendizajes de trayectoria

El aprendizaje y desarrollo integral de los estudiantes, en su transitar por el Nivel Medio Superior, se representan a través de los aprendizajes de trayectoria. Con la transversalidad de los saberes y habilidades entrelazados progresivamente en el currículum fundamental y el ampliado es que se lograrán estos aprendizajes que conforman a su vez conforman el perfil de egreso.

Los aprendizajes de trayectoria del Recurso sociocognitivo Cultura digital constituyen un principio articulador de los componentes del modelo educativo, pues se pretende que los estudiantes adopten una cultura digital en el uso de las herramientas digitales.

Particularmente en Pensamiento computacional, las pretensiones referidas son bajo los siguientes aprendizajes:

- Soluciona problemas de su entorno utilizando el pensamiento y lenguajes algorítmico.
- Diseña y elabora contenidos digitales mediante técnicas, métodos, y recursos tecnológicos para fortalecer su creatividad e innovar en su vida cotidiana.

III. Progresiones de aprendizaje

El proceso con el que se pretende el estudiante alcance las metas de aprendizaje está orientado a abordarlo progresivamente durante el desarrollo de los temas y la construcción de su aprendizaje, a su vez se articulan en categorías y subcategorías, que están planteadas de manera global en los contenidos disciplinares.

La asignatura de Pensamiento computacional se estructura en cinco progresiones de aprendizaje enmarcadas por las categorías de: Pensamiento algorítmico y Creatividad digital.

En las siguientes tablas se describe cada una de las progresiones, metas, categorías y sus correspondientes subcategorías, además se presentan algunas orientaciones pedagógico-didácticas para su eficaz implementación y evaluación.

También se sugieren actividades de aprendizaje que atienden los temas de cada progresión.

Progresión 1. Bases del pensamiento computacional	Tiempo estimado: 9 horas
Identifica qué es el pensamiento computacional y lo aplica en la representación de soluciones a problemas cotidianos mediante algoritmos básicos (pseudocódigo, diagramas de flujo), considerando su contexto y recursos disponibles.	
Metas de aprendizaje	
Identifica los principios del pensamiento computacional, su descomposición, abstracción y patrones para diseñar, implementar y evaluar algoritmos de problemas de su vida cotidiana.	
Representa la solución de problemas mediante pensamiento algorítmico seleccionando métodos, diagramas o técnicas.	
Aplica lenguaje algorítmico utilizando medios digitales para resolver situaciones o problemas del contexto.	
Categoría	Subcategoría
Pensamiento algorítmico	<ul style="list-style-type: none"> • Resolución de problemas • Pensamiento computacional y lenguaje algorítmico
Aprendizaje de trayectoria	
Soluiona problemas de su entorno utilizando el pensamiento y lenguaje algorítmico.	
Temas	
1.1 Inicios de la algoritmia <ul style="list-style-type: none"> 1.1.1 Orígenes del pensamiento computacional 1.2 Fases del pensamiento computacional <ul style="list-style-type: none"> 1.2.1 Identificación del problema 1.2.2 Descomposición 1.2.3 Reconocimiento de patrones 1.2.4 Abstracción 1.2.5 Diseño de algoritmo 1.2.6 Implementación 1.2.7 Evaluación 	
Orientaciones pedagógicas específicas de la progresión	
Esta progresión debe ser desarrollada bajo un enfoque activo, contextualizado y reflexivo, donde el estudiante sea protagonista de su aprendizaje, construya conocimiento a partir de su realidad y fortalezca habilidades de resolución de problemas, pensamiento lógico y autorregulación del aprendizaje.	
Inicio: diagnóstico y motivación	
Se sugiere poner interés en conocer las actitudes, percepciones e ideas previas de los estudiantes, y despertar su interés por el pensamiento computacional:	

- Iniciar con preguntas diagnósticas que indaguen qué entienden por problemas, algoritmos, instrucciones o soluciones paso a paso.
- Promover un ambiente de confianza donde el estudiante pueda expresar libremente sus ideas y experiencias cotidianas.
- Utilizar situaciones cercanas a su realidad (rutinas diarias, uso del celular, actividades escolares) para mostrar que el pensamiento computacional ya está presente en su vida.
- Emplear actividades breves de activación, como encontrar patrones en secuencias simples, para captar la atención y generar curiosidad.

Adquisición e integración del conocimiento

Esto es guiar al estudiante para que comprender los conceptos fundamentales del pensamiento computacional y la algoritmia.

- Favorecer la indagación guiada, la lectura comprensiva y la escucha activa mediante: lectura del libro de texto y observación de videotutoriales, infografías o presentaciones interactivas.
- Presentar de manera progresiva los temas clave: Orígenes del pensamiento computacional; Inicios de la algoritmia y Fases del pensamiento computacional: identificación del problema, descomposición, reconocimiento de patrones y abstracción.
- Apoyar la organización de la información usando tablas, esquemas y organizadores gráficos, que ayuden al estudiante a estructurar lo aprendido.

Extensión y refinamiento del aprendizaje

Profundizar en la comprensión del pensamiento computacional y establecer conexiones con otros saberes y habilidades.

- Proponer actividades prácticas donde el estudiante identifique problemas de su entorno, descomponga el problema en partes más pequeñas, reconozca patrones y seleccione información relevante. diseñe algoritmos sencillos.
- Utilizar tablas de trabajo para el diseño, implementación y evaluación de algoritmos.
- Promover el uso del lenguaje algorítmico (pseudocódigo y diagramas de flujo) como una herramienta para representar soluciones.
- Plantear situaciones reales que permitan dar un uso significativo y aplicable al conocimiento adquirido.

Cierre: Metacognición y evaluación

Desarrollar habilidades metacognitivas que permitan al estudiante reflexionar sobre su propio aprendizaje.

- Guiar al estudiante para que planifique cómo resolver un problema, autorregule su proceso al diseñar y aplicar un algoritmo y, evalúe la eficacia de su solución y proponga mejoras.
- Utilizar la elaboración de un algoritmo como evidencia de aprendizaje.
- Fomentar la autoevaluación y la reflexión mediante preguntas como: ¿qué aprendí? ¿qué fue lo más difícil? ¿cómo puedo aplicar esto en otras situaciones?

Otro aspecto importante es promover el trabajo en equipo en la elaboración de algunos productos, donde los estudiantes deban negociar ideas, distribuir tareas y apoyarse mutuamente. Esto desarrolla habilidades de comunicación y resolución de conflictos.

Desde la responsabilidad del docente debe proporcionarse una retroalimentación constante como parte fundamental de la evaluación, que guíe a los estudiantes a mejorar su comprensión y sus productos, no solo señalando errores, sino también ofreciendo caminos para el crecimiento.

Actividades de aprendizaje sugeridas

Diagnóstica	<ul style="list-style-type: none"> • Responder preguntas
Activación	<ul style="list-style-type: none"> • Encontrar patrón en secuencia
Aprendizaje	<ul style="list-style-type: none"> • Preguntas de identificación del problema • Tabla descomposición de problema • Tabla de reconocimiento de patrones • Tabla de abstracción • Diseño de algoritmo • Tabla de implementación y evaluación
Tarea	<ul style="list-style-type: none"> • Lectura del tema en el libro de texto • Observar videotutoriales, infografías y presentación interactiva • Organizador gráfico
Evaluación	<ul style="list-style-type: none"> • Algoritmo

Progresión 2. Algoritmia en IDE

Tiempo estimado: **12 horas**

Resuelve problemas cotidianos y académicos mediante la construcción de algoritmos en IDE, utilizando estructuras de control decisivas e iterativas para automatizar procesos y validar la solución de manera digital.

Metas de aprendizaje

Identifica situaciones de la vida cotidiana que pueden resolverse de manera más eficiente utilizando secuencias y ciclos.

Comprueba la lógica y funcionamiento de algoritmos para representar sus soluciones mediante IDE corrigiendo errores y optimizando el código.

Categoría	Subcategoría
Pensamiento algorítmico	<ul style="list-style-type: none"> • Resolución de problemas • Pensamiento computacional y lenguaje algorítmico
Aprendizaje de trayectoria	
Soluciona problemas de su entorno utilizando el pensamiento y lenguaje algorítmico.	
Temas	
<p>2.1 Aplicación en entorno de desarrollo integrado</p> <p>2.1.1 Interfaz</p> <p>2.1.2 Componentes del pseudocódigo</p> <p> 2.1.2.1 Variables</p> <p> 2.1.2.2 Constantes</p> <p> 2.1.2.3 Tipos de datos</p> <p> 2.1.2.4 Operadores</p> <p> 2.1.2.5 Acciones primitivas secuenciales</p> <p>2.2 Estructuras de control</p> <p> 2.2.1 Secuencial</p> <p> 2.2.2 Condicionales</p> <p> 2.2.2.1 Simple</p> <p> 2.2.2.2 Doble</p> <p> 2.2.2.3 Anidada</p> <p> 2.2.2.4 Según...hacer</p> <p> 2.2.3 Repetitivas</p> <p> 2.2.3.1 Mientras...hacer</p> <p> 2.2.3.2 Repetir ...hasta que</p> <p> 2.2.3.3 Para...hasta...con paso</p>	
Orientaciones pedagógicas específicas de la progresión	
<p>Esta progresión orienta a la resolución de problemas, donde el estudiante consolide el pensamiento algorítmico, fortalezca habilidades digitales y utilice un entorno de desarrollo integrado (IDE) como una herramienta para validar y mejorar sus soluciones.</p> <p>Inicio: Diagnóstico y motivación</p> <p>Iniciar identificando los conocimientos previos del estudiante sobre algoritmos y programación, así como fortalecer su motivación hacia el uso de herramientas digitales, a través de:</p> <ul style="list-style-type: none"> • Preguntas diagnósticas relacionadas con secuencias, decisiones y repeticiones en actividades cotidianas. • Proponer una situación de la vida diaria para que el estudiante identifique la necesidad de automatizar procesos. 	

- Promover la participación y el diálogo para que el estudiante reconozca la utilidad de los algoritmos en contextos reales.
- Presentar el IDE como una herramienta que facilita la construcción, prueba y validación de soluciones.

Adquisición e integración del conocimiento

Guiar al estudiante para que comprenda los elementos básicos del entorno de desarrollo integrado y las estructuras de control.

- Introducir gradualmente el entorno de desarrollo integrado, explicando su interfaz, sintaxis y funcionamiento básico.
- Explicar conceptos fundamentales como: tipos de datos, variables y constantes.
- Guiar al estudiante en la elaboración de pseudocódigos, mostrando la relación entre instrucciones y resultados.
- Apoyar el aprendizaje mediante lecturas del libro de texto, observación de videotutoriales y ejercicios guiados dentro del IDE.
- Favorecer la organización de la información mediante tablas comparativas que distingan los distintos tipos de estructuras de control y sus características.

Extensión y refinamiento del aprendizaje

Profundizar en el uso de estructuras condicionales y repetitivas para resolver problemas de mayor complejidad.

- Proponer actividades donde el estudiante diseñe y ejecute algoritmos utilizando: Estructuras condicionales (simple, doble, anidada y según la opción) y Estructuras repetitivas (mientras, repetir hasta y para).
- Fomentar la experimentación dentro del IDE, permitiendo que el estudiante identifique errores, los corrija y optimice su código.
- Promover el análisis lógico del algoritmo para comprobar su funcionamiento y eficiencia.
- Plantear problemas académicos y cotidianos que requieran automatización de procesos, reforzando el uso significativo del lenguaje algorítmico.

Cierre: Metacognición y evaluación

Desarrollar habilidades metacognitivas que fortalezcan la autonomía del estudiante en la resolución de problemas algorítmicos.

- Invitar al estudiante a reflexionar sobre: la elección de la estructura de control utilizada, las dificultades encontradas y cómo fueron resueltas. la eficiencia de su algoritmo.
- Utilizar la elaboración y ejecución de pseudocódigos en el IDE como evidencia de aprendizaje.

- Promover la autoevaluación y coevaluación mediante preguntas como: ¿qué estructura de control fue más adecuada para resolver el problema? ¿cómo mejoraría mi algoritmo? ¿en qué otras situaciones podrían aplicar este procedimiento?

Considerar la evaluación oportuna durante el desarrollo de las sesiones, pues es significativo motivar que los estudiantes revisen críticamente sus propios trabajos y los de sus compañeros, ofreciendo y recibiendo retroalimentación constructiva para mejorar la calidad de sus producciones y de su aprendizaje.

Actividades de aprendizaje sugeridas	
Diagnóstica	<ul style="list-style-type: none"> • Responder preguntas
Activación	<ul style="list-style-type: none"> • Dar respuesta a una situación cotidiana
Aprendizaje	<ul style="list-style-type: none"> • Pseudocódigo: secuencial • Pseudocódigo: condicional Simple y Doble • Pseudocódigo: condicional Anidada • Pseudocódigo: condicional Según...hacer • Pseudocódigo: repetitiva Mientras...hacer • Pseudocódigo: condicional Repetir...hasta • Pseudocódigo: Repetitiva Para...hasta • Tabla comparativa
Tarea	<ul style="list-style-type: none"> • Lectura del tema en el libro de texto • Observar videotutoriales
Evaluación	<ul style="list-style-type: none"> • Pseudocódigo

Progresión 3. Programación estructurada: estructuras de control

Tiempo estimado: **9 horas**

Codifica instrucciones en un lenguaje de programación estructurada, empleando estructuras de control secuenciales y repetitivas para determinar el orden lógico y eficiente en que se ejecutan en la resolución de problemas.

Metas de aprendizaje

Distingue la sintaxis básica de C++ y la utilidad de las estructuras de control para organizar la ejecución de instrucciones.

Representa soluciones a problemas cotidianos y académicos mediante algoritmos que incorporan estructuras de control secuenciales y repetitivas.

Codifica, compila y ejecuta programas en C++ validando su funcionamiento y corrigiendo errores en el uso de estructuras de control.

Categoría	Subcategoría
Pensamiento algorítmico	Resolución de problemas

	Pensamiento computacional y lenguaje algorítmico
Aprendizaje de trayectoria	
Soluciona problemas de su entorno utilizando el pensamiento y lenguaje algorítmico.	
Temas	
3.1. Lenguajes de programación <ul style="list-style-type: none"> 3.1.1. Historia 3.1.2. Clasificación 3.1.3. Editores de código 3.2. Estructura básica de un programa <ul style="list-style-type: none"> 3.2.1. Programación estructurada 3.2.2. Lenguaje de programación estructurada 3.2.3. Sintaxis y elementos básicos 3.2.4. Variables y tipo de datos 3.2.5. Entrada y salida de datos 3.2.6. Operadores 3.3. Estructuras de control <ul style="list-style-type: none"> 3.3.1. Condicionales <ul style="list-style-type: none"> 3.3.1.1. If 3.3.1.2. If-else 3.3.1.3. If-else-if 3.3.1.4. Switch.case 3.3.2. Repetitivas <ul style="list-style-type: none"> 3.3.2.1. For 3.3.2.2. While 3.3.2.3. Do while 	
Orientaciones pedagógicas específicas de la progresión	
<p>Esta progresión orienta al desarrollo de experiencias de aprendizaje que permitan al estudiante codificar, compilar y ejecutar programas en un lenguaje de programación estructurada, empleando estructuras de control secuenciales, condicionales y repetitivas para resolver problemas de manera lógica y eficiente.</p> <p>Inicio: Diagnóstico y motivación</p> <p>Es importante partir de la identificación de los conocimientos previos del estudiante sobre programación y fortalecer su interés por el uso de lenguajes de programación estructurada.</p> <ul style="list-style-type: none"> • Aplicar preguntas diagnósticas sobre conceptos básicos como instrucciones, variables y secuencias lógicas. • Desarrollar una actividad de activación (resolución de un laberinto) que permita al estudiante reconocer la importancia del orden y la toma de decisiones en la resolución de problemas. • Relacionar la actividad con situaciones reales donde seguir instrucciones claras es indispensable. 	

- Presentar la programación estructurada como una herramienta para organizar soluciones de forma clara y eficiente.

Adquisición e integración del conocimiento

Guiar al estudiante en la comprensión de los fundamentos de la programación estructurada y la sintaxis básica de un lenguaje de programación en particular. Puede llevarse a cabo con:

- La exposición de los lenguajes de programación, su historia y clasificación, resaltando la importancia de los lenguajes estructurados.
- Explicar la estructura básica de un programa, identificando sus elementos principales.
- Abordar de forma gradual las variables y tipos de datos, entrada y salida de datos y el uso de los distintos tipos de operadores.
- Favorecer la comprensión de la sintaxis mediante ejemplos sencillos y programas guiados.
- Complementar el aprendizaje con la lectura del libro de texto y la observación de videos y videotutoriales.

Extensión y refinamiento del aprendizaje

Aplicar las estructuras de control para resolver problemas mediante la programación.

- Guiar al estudiante en la elaboración de programas que incorporen:
 - Estructuras condicionales (if, if-else, if-else-if, switch-case).
 - Estructuras repetitivas (for, while, do while).
- Proponer problemas cotidianos y académicos que requieran el uso de estas estructuras.
- Fomentar la práctica constante de codificación, compilación y ejecución de programas.
- Promover la detección y corrección de errores (debugging) como parte natural del aprendizaje.
- Estimular el análisis del código para mejorar su claridad y eficiencia.

Cierre: Metacognición y evaluación

Para que el estudiante desarrolle la capacidad de reflexionar sobre su proceso de programación y aprendizaje puede ser con orientaciones pedagógicas que:

- Inviten al estudiante a analizar la lógica utilizada para resolver problemas, la estructura de control seleccionada y los errores encontrados y las soluciones aplicadas.
- Elaboren un programa funcional en C++ como evidencia de aprendizaje.
- Promover la autoevaluación mediante preguntas como: ¿qué estructura de control fue más adecuada y por qué? ¿cómo puedo mejorar mi código? ¿en qué otros contextos puedo aplicar lo aprendido?

Para las estructuras de control la evaluación debe validar la lógica de bifurcación y repetición mediante el análisis de flujos y condiciones. Es importante considerar

que la retroalimentación oportuna en la identificación de errores como bucles infinitos o condiciones mal planteadas justo al depurar el código, puesto esto permitirá al estudiante ajustar la jerarquía lógica y optimizar el control del programa de manera inmediata, garantizando un aprendizaje significativo.

Actividades de aprendizaje sugeridas	
Diagnóstica	<ul style="list-style-type: none"> • Responder preguntas
Activación	<ul style="list-style-type: none"> • Resolver laberinto
Aprendizaje	<ul style="list-style-type: none"> • Programa: condicional Simple y Doble • Programa: condicional Anidadas • Programa: repetitiva Para • Programa: repetitiva Haz mientras
Tarea	<ul style="list-style-type: none"> • Lectura del tema en el libro de texto • Observar videos y videotutoriales
Evaluación	<ul style="list-style-type: none"> • Programa

Progresión 4. Programación estructurada: estructuras de datos	Tiempo estimado: 6 horas
Codifica en C++ arreglos unidimensionales para almacenar, procesar y manipular conjuntos de datos, determinando la ejecución de instrucciones de manera organizada y eficiente en la resolución de problemas.	
Metas de aprendizaje	
Identifica la utilidad de los arreglos unidimensionales en la resolución de problemas que requieren manejar múltiples valores del mismo tipo de dato.	
Representa soluciones a problemas cotidianas y académicos mediante el diseño de algoritmos que emplean arreglos unidimensionales.	
Codifica y ejecuta programas en C++ que utilizan arreglos unidimensionales para almacenar, recorrer y procesar datos (suma, promedio, máximo, mínimo, búsqueda lineal y ordenamiento).	
Categoría	Subcategoría
Pensamiento algorítmico	Resolución de problemas Pensamiento computacional y lenguaje algorítmico
Aprendizaje de trayectoria	
Solucciona problemas de su entorno utilizando el pensamiento y lenguaje algorítmico.	
Temas	
4.1 Estructuras de datos	
4.1.1 Arreglos unidimensionales	
4.1.1.1 Declaración	
4.1.1.2 Inserción de datos	

<p>4.1.1.3 Acceso 4.1.1.4 Operaciones</p>	<p>Orientaciones pedagógicas específicas de la progresión</p>
<p>Las orientaciones para la progresión 5 van dirigidas a que el docente diseñe experiencias de aprendizaje que permitan al estudiante codificar arreglos unidimensionales en un lenguaje de programación estructurada para almacenar, procesar y manipular conjuntos de datos, aplicando una ejecución organizada y eficiente de instrucciones en la resolución de problemas cotidianos y académicos.</p>	
<p>Inicio: Diagnóstico y motivación</p>	
<p>Se sugiere iniciar identificando los conocimientos previos del estudiante sobre el manejo de datos y motivar el uso de arreglos como una solución eficiente, por ejemplo:</p>	<ul style="list-style-type: none"> • Aplicar preguntas indagatorias relacionadas con el almacenamiento de información y el manejo de múltiples datos del mismo tipo. • Plantear una situación cotidiana donde el estudiante reconozca la necesidad de organizar varios valores. • Promover el análisis y la participación para que el estudiante identifique las limitaciones de usar variables individuales. • Introducir el concepto de arreglo unidimensional como una estructura que facilita el manejo de conjuntos de datos.
<p>Adquisición e integración del conocimiento</p>	
<p>Con la finalidad de que el estudiante comprenda el concepto, la estructura y la sintaxis de los arreglos unidimensionales en un lenguaje de programación estructurada. Para ello se sugiere:</p>	<ul style="list-style-type: none"> • Explicar de forma clara qué son las estructuras de datos y su importancia en la programación. • Introducir los arreglos unidimensionales, abordando: declaración, inserción de datos y acceso a los elementos. • Mostrar ejemplos sencillos que permitan visualizar el funcionamiento del arreglo y su relación con los índices. • Apoyar el aprendizaje con lectura del libro de texto y observar videos, videotutoriales, infografías y presentaciones interactivas. • Favorecer la organización de la información mediante esquemas o cuadros comparativos.
<p>Extensión y refinamiento del aprendizaje</p>	
<p>Esto puede hacerse aplicando arreglos unidimensionales para resolver problemas que implican el procesamiento de datos, para ello se sugiere:</p>	<ul style="list-style-type: none"> • Guiar al estudiante en la elaboración de programas en un lenguaje de programación estructurada que permitan leer y escribir datos en un arreglo, recorrerlo para realizar operaciones como suma, promedio, valor máximo

- y mínimo e implementar búsqueda lineal y aplicar algoritmos básicos de ordenamiento.
- Promover la resolución de problemas cotidianos y académicos que requieran el manejo eficiente de datos.
 - Fomentar la revisión del código para detectar errores y optimizar su funcionamiento.
 - Establecer conexiones con aprendizajes previos, como el uso de estructuras repetitivas para recorrer arreglos.

Cierre: Metacognición y evaluación

Para desarrollar habilidades metacognitivas relacionadas con el manejo y procesamiento de datos se solicita:

- Invitar al estudiante a reflexionar sobre la utilidad de los arreglos frente a otras alternativas, la lógica empleada para procesar los datos, las dificultades encontradas y cómo fueron superadas.
- Utilizar la elaboración y ejecución de un programa con arreglos unidimensionales en un lenguaje como evidencia de aprendizaje.
- Promover la autoevaluación mediante preguntas como: ¿cómo mejoré la organización de los datos? ¿qué operación fue más compleja y por qué? ¿en qué otras situaciones puedo aplicar el uso de arreglos?

Para evaluar arreglos unidimensionales, debe centrarse la evaluación en la lógica de indexación y gestión de memoria mediante pruebas de escritorio o ejecución de código. La retroalimentación oportuna debe señalar errores sintácticos o de lógica (como el acceso a índices fuera de rango) inmediatamente después de la compilación. Esto permite que el estudiante corrija su estructura de control en el momento, consolidando el manejo correcto de los datos.

Actividades de aprendizaje sugeridas	
Diagnóstica	<ul style="list-style-type: none"> • Responder preguntas
Activación	<ul style="list-style-type: none"> • Resolver una situación cotidiana
Aprendizaje	<ul style="list-style-type: none"> • Programa: lectura y escritura en arreglo • Programa: búsqueda lineal • Programa: ordenamiento
Tarea	<ul style="list-style-type: none"> • Lectura del tema en el libro de texto • Observar videos, videotutoriales, infografías, líneas del tiempo y presentaciones interactivas de los temas
Evaluación	<ul style="list-style-type: none"> • Programa

Progresión 5. Robótica educativa

Tiempo estimado: **12 horas**

Simula sistemas robóticos mediante aplicaciones gráficas y la programación en Arduino, implementando estructuras básicas de código (setup y loop), funciones elementales (pinMode, digitalWrite, delay), control de salidas múltiples mediante

ciclos y retardos, así como el uso de sensores y actuadores para resolver problemas simples de automatización.

Metas de aprendizaje

Identifica la importancia de la robótica educativa como herramienta para comprender la interacción entre hardware y software.

Configura sistemas básicos de automatización en un entorno gráfico controlando las salidas.

Codifica programas con estructuras de control, funciones básicas, bucles, sensores y actuadores en simulaciones.

Categoría	Subcategoría
Pensamiento algorítmico	<ul style="list-style-type: none"> • Pensamiento computacional y lenguaje algorítmico. • Resolución de problemas
Creatividad digital	<ul style="list-style-type: none"> • Desarrollo digital

Aprendizaje de trayectoria

Soluciona problemas de su entorno utilizando el pensamiento y lenguaje algorítmico.

Temas

- 5.1 Introducción a la robótica
 - 5.1.1 Historia
 - 5.1.2 Conceptos básicos de electricidad y electrónica
 - 5.1.3 Aplicaciones
- 5.2 Aplicación simuladora de circuitos por bloques
 - 5.2.1 Interfaz gráfica
 - 5.2.2 Componentes básicos
 - 5.2.3 Componentes de entrada
- 5.3 Plataforma de electrónica para programar robots
 - 5.3.1 Conceptos básicos
 - 5.3.2 Programación básica
- 5.4. Sensores y actuadores
 - 5.4.1 Sensores
 - 5.4.2 Actuadores
 - 5.4.3 Integración de sensores y actuadores

Orientaciones pedagógicas específicas de la progresión

En esta progresión de robótica educativa, se sugiere el docente oriente al estudiante en el diseño de experiencias de aprendizaje que permitan simular sistemas robóticos mediante aplicaciones gráficas y programación básica en una plataforma, integrando sensores y actuadores para resolver problemas simples de automatización, comprendiendo la relación entre hardware y software.

Inicio: Diagnóstico y motivación

Las orientaciones pedagógicas para identificar los conocimientos previos del estudiante sobre robótica, electricidad y automatización, y despertar su interés por la robótica educativa son:

- Aplicar preguntas indagatorias relacionadas con el uso de dispositivos electrónicos, sensores y sistemas automatizados en la vida cotidiana.
- Plantear una situación cotidiana (por ejemplo, encendido automático de luces, control de temperatura o sistemas de seguridad) para que el estudiante reconozca la utilidad de la robótica.
- Promover el diálogo para que el estudiante identifique la robótica como una herramienta para resolver problemas reales.
- Presentar la robótica educativa como un espacio de aprendizaje creativo, práctico y seguro mediante simuladores virtuales.

Adquisición e integración del conocimiento

Acompañar al estudiante en el desarrollo de la progresión de manera que comprenda los conceptos básicos de robótica, electrónica y el funcionamiento de los entornos de simulación, a través de:

- Introducir la robótica abordando su historia, aplicaciones y relevancia actual.
- Explicar de manera sencilla los conceptos básicos de electricidad y electrónica, necesarios para el uso de sensores y actuadores.
- Presentar la aplicación simuladora de circuitos por bloques, explicando: su interfaz gráfica, los componentes básicos y de entrada.
- Introducir la plataforma de electrónica para programar robots, abordando conceptos básicos, estructura del código (setup y loop) y funciones elementales (pinMode, digitalWrite, delay).
- Apoyar el aprendizaje con lecturas, videos, videotutoriales e infografías.

Extensión y refinamiento del aprendizaje

Llevar a que estudiante aplique la programación y la simulación robótica para resolver problemas de automatización.

- Guiar al estudiante en la construcción de sistemas simulados como: el encendido de un LED, la simulación de un semáforo simple, el encendido automático de LED en la oscuridad y simular un ventilador automático con sensor de temperatura.
- Fomentar el uso de estructuras de control y ciclos para manejar salidas múltiples y retardos.
- Integrar sensores y actuadores, promoviendo la experimentación y el análisis del comportamiento del sistema.
- Establecer conexiones con aprendizajes previos, como el uso de condicionales, bucles y arreglos.

Cierre: Metacognición y evaluación

Para fortalecer la reflexión sobre el proceso de aprendizaje y la aplicación de la robótica educativa, se puede:

- Invitar al estudiante a reflexionar sobre la relación entre hardware y software, el funcionamiento del sistema simulado y las mejoras posibles al sistema diseñado.
- Utilizar como evidencia de aprendizaje la simulación de un sistema robótico funcional, como una alarma inteligente.
- Promover la autoevaluación y coevaluación mediante preguntas como: ¿qué problema resolví con mi sistema? ¿cómo funcionan los sensores y actuadores en conjunto? ¿qué cambios haría para mejorar mi diseño?

Es importante recordar que la evaluación debe ser continua y formativa, integrándose en cada fase de la secuencia para monitorear el progreso en tiempo real. Asimismo la retroalimentación debe ser específica, constructiva y oportuna, brindándose justo después de la actividad para que el alumno identifique sus aciertos y áreas de mejora. Ya que este proceso cíclico permite ajustar la enseñanza y empoderar al estudiante en su propio aprendizaje.

Actividades de aprendizaje sugeridas	
Diagnósticas	<ul style="list-style-type: none">• Responder preguntas
Activación	<ul style="list-style-type: none">• Resolver una situación cotidiana
Aprendizaje	<ul style="list-style-type: none">• Sistema simulado: encendido de led• Sistema simulado: semáforo simple• Sistema simulado: encendido led en oscuridad• Sistema simulado: ventilador automático con sensor de temperatura
Tareas	<ul style="list-style-type: none">• Tabla de relación• Lectura del tema en el libro de texto• Observar videos, videotutoriales e infografías de los temas
Evaluación	<ul style="list-style-type: none">• Sistema simulado: alarma inteligente

IV. Transversalidad con otras Áreas de Conocimiento y Recursos Sociocognitivos y Socioemocionales

La transversalidad como estrategia curricular se instituye dentro del nuevo MCC para el NMS para ayudar en el logro de los aprendizajes de trayectoria y en la articulación entre el currículum fundamental y el currículum ampliado, con la pretensión de evitar la segmentación del conocimiento.

Particularmente Cultura digital está establecido como un recurso sociocognitivo transversal a ambos currículums, en virtud de que aborda las tres visiones de la

transversalidad: desarrolla el trabajo con otras disciplinas (Multidisciplinario), entre diferentes disciplinas (Interdisciplinario), a través y más allá de varias disciplinas (Transdisciplinario), todo esto a través del trabajo individual y colaborativo, cumpliendo así con la aplicación y desarrollo de metodologías activas que permiten el aprendizaje integral del estudiante y a la vez considerar su cuidado físico y emocional al resolver situaciones personales y de su comunidad (SEP, 2024c)

De manera que, la integración de Pensamiento computacional con las demás asignaturas de la malla curricular del Bachillerato UAS de manera natural hace posible con la ayuda del docente, llevar a cabo en el proceso de enseñanza-aprendizaje el diseño e implementación, en su planeación didáctica, de las estrategias didácticas y de evaluación donde delineen situaciones contextualizadas que permitan el logro de aprendizajes significativos, asimismo al generar ambientes propicios para el aprendizaje.

Recurso sociocognitivo Lengua y comunicación

En el programa de Pensamiento computacional se establece explícitamente que el docente debe diseñar actividades que contribuyan al desarrollo de habilidades comunicativas, capacidad creativa y pensamiento algorítmico.

Esta articulación se logra con el recurso sociocognitivo de Lengua y comunicación mediante la comunicación de procesos, es decir, el estudiante debe ser capaz de explicar, tanto de forma oral como escrita, la lógica y los pasos seguidos para resolver un problema mediante algoritmos. La creación de pseudocódigo y la documentación de algoritmos son ejercicios de escritura técnica que requieren precisión, orden lógico y claridad comunicativa.

Aunque el programa no detalla una sección exclusiva para inglés, la vinculación es inherente a la disciplina de la computación a través de los lenguajes de programación. La enseñanza de lenguajes como C++ y el uso de plataformas como Arduino requieren que el estudiante comprenda y aplique sintaxis y palabras clave técnicas que provienen del inglés (ej. if, else, while, setup, loop, pinMode, digitalWrite).

Además de la documentación técnica, el programa sugiere la observación de videotutoriales y el uso de simuladores virtuales, donde gran parte de la información técnica y de vanguardia en la Industria 4.0 se encuentra originalmente en este idioma.

Recurso sociocognitivo Pensamiento matemático

Tanto el Pensamiento Matemático como la Cultura Digital (donde se inserta el Pensamiento Computacional) son definidos como recursos sociocognitivos transversales. Su articulación es natural y esencial para evitar la segmentación del

conocimiento, sirviendo como base para que el estudiante acceda y construya saberes en otras áreas.

La conexión principal ocurre a través del desarrollo de capacidades analíticas de Abstracción y Lógica, donde el Pensamiento Computacional utiliza la abstracción, el reconocimiento de patrones y el pensamiento algorítmico, procesos que son intrínsecos al razonamiento matemático; asimismo en el lenguaje algorítmico, la representación de soluciones mediante diagramas de flujo y pseudocódigo requiere de una estructura lógica formal similar a la utilizada en la resolución de problemas matemáticos.

En el desarrollo de las progresiones de aprendizaje, la articulación con el pensamiento matemático se vuelve operativa al utilizar operadores aritméticos y lógicos para la construcción de programas; en la implementación de estructuras de datos, como arreglos unidimensionales, para realizar operaciones matemáticas complejas como sumas, promedios, valores máximos y mínimos; y al gestionar variables y constantes, conceptos fundamentales de la matemática aplicados al entorno digital.

El programa de Pensamiento computacional adopta el enfoque STEAM (Ciencia, Tecnología, Ingeniería, Artes y Matemáticas), el cual integra explícitamente las Matemáticas con la tecnología y la lógica computacional para resolver problemas reales mediante la experimentación física y la programación. Asimismo sugiere a los docentes diseñar estrategias didácticas contextualizadas que vinculen el pensamiento algorítmico con la matemática para que el estudiante adquiera un aprendizaje integral y sea capaz de sistematizar información de manera ética y responsable.

Recurso sociocognitivo Conciencia histórica

La articulación de la UAC de Pensamiento computacional con el recurso sociocognitivo de Conciencia histórica se establece mediante la comprensión de la tecnología como un producto humano que transforma y es transformado por el tiempo.

Desde Conciencia histórica se sitúa a los estudiantes para que comprendan críticamente el pasado, interpreten su presente y proyecten transformaciones futuras. Permite al estudiante comprender que el pensamiento computacional no surgió de forma aislada, sino como respuesta a necesidades históricas específicas. El programa aborda temas como la Industria 4.0, lo que permite analizar cómo las revoluciones industriales previas han reconfigurado la economía, el trabajo y las relaciones sociales a lo largo del tiempo. Entender la evolución desde los algoritmos

mecánicos hasta la inteligencia artificial ayuda al alumno a situarse en su presente y proyectar futuros posibles.

La articulación se refuerza al reflexionar sobre la responsabilidad ética del uso de la tecnología desde una perspectiva histórica. Se vincula la importancia de la preservación de la información (progresiones del programa) con la construcción de la memoria colectiva. El estudiante comprende que lo que genera hoy mediante procesos algorítmicos formará parte del archivo histórico del mañana.

Área del conocimiento Ciencias naturales, experimentales y tecnología

El programa de estudios integra el enfoque STEAM, el cual facilita la resolución de problemas reales mediante la experimentación física y la lógica algorítmica. Este modelo permite que los conceptos de ciencias naturales se materialicen a través de prototipos tecnológicos.

En la Progresión 5, donde propone aprender Robótica educativa, la articulación con las ciencias experimentales se vuelve operativa mediante la simulación de sistemas que responden a variables físicas. La interacción Hardware-Software se da cuando el estudiante utiliza sensores y actuadores para comprender cómo la programación (software) interactúa con componentes electrónicos (hardware).

También se propone programar sistemas simulados que resuelven problemas de automatización basados en fenómenos naturales, tales como control de temperatura con la actividad para simular un ventilador automático con sensor de temperatura y el encendido automático de un LED en la oscuridad (relacionado con la intensidad de luz).

De manera que la asignatura introduce conceptos fundamentales de electricidad y electrónica necesarios para el uso de sensores y actuadores. Esto conecta directamente con los contenidos de física y tecnología, permitiendo al estudiante entender las bases científicas detrás de los dispositivos digitales que utiliza.

También promueven metodologías basadas en la indagación y el descubrimiento. Estas son las mismas herramientas que sustentan el método científico en las ciencias naturales, fomentando que el alumno identifique problemas de su entorno, construya modelos o prototipos (algoritmos o simulaciones robóticas) para probar soluciones y evalúe la eficacia de su solución mediante la experimentación digital y física.

Área del conocimiento Ciencias sociales

Como recurso sociocognitivo, el Pensamiento computacional se vincula con las Ciencias sociales a través de la formación de una ciudadanía digital responsable. En lo concerniente a ética y Legalidad, el programa enfatiza el uso responsable de la información y la tecnología, lo que se alinea con los temas de normas, derechos y obligaciones en la sociedad actual. Para el impacto social de la tecnología se analiza cómo la automatización y la Industria 4.0 transforman las relaciones laborales y la estructura económica de la sociedad.

El pensamiento computacional aporta métodos para desglosar problemas complejos, lo cual es fundamental en las Ciencias Sociales. Permite que el estudiante divida un problema social (como la desigualdad o el crecimiento urbano) en partes más pequeñas y manejables para su estudio y ayuda a identificar tendencias en comportamientos sociales o datos demográficos, facilitando la interpretación de fenómenos colectivos y la identificación de patrones.

Área del conocimiento Humanidades

La articulación de la asignatura de Pensamiento Computacional con el área de conocimiento de Humanidades se establece a partir de una visión crítica y ética sobre el papel de la tecnología en la experiencia humana. Según los lineamientos del programa de la UAS y el MCCEMS, esta relación no es técnica, sino reflexiva y existencial.

Se articula principalmente a través del pensamiento crítico y la naturaleza humana, en Humanidades se busca que el estudiante reflexione sobre quién es y cómo se relaciona con el mundo y el Pensamiento computacional aporta una nueva capa a esta reflexión, el autoconocimiento en la Era digital, lo que permite al estudiante cuestionar cómo los algoritmos y la inteligencia artificial moldean sus deseos, pensamientos y su percepción de la realidad.

Así también en la lógica y Argumentación, donde el rigor lógico necesario para construir un algoritmo (pasos ordenados, coherencia y no ambigüedad) es un reflejo de la lógica formal que se estudia en la filosofía, ayudando a estructurar pensamientos más claros y argumentos más sólidos.

Pensamiento computacional también enfatiza el uso de la tecnología de manera ética y responsable. Esta es la conexión más fuerte con las Humanidades, donde se abordan temas como la discusión sobre qué problemas deben ser resueltos por la tecnología y cuáles requieren una respuesta humana, ética o moral. La sistematización de la información (progresiones del programa) se vincula con el derecho humano a la privacidad y la protección de la identidad en entornos virtuales.

En torno a fomentar la capacidad creativa, en la intersección con las Humanidades, esto se traduce en percibir a la Programación como arte, ver el código no solo como una herramienta funcional, sino como una forma de expresión personal y creativa. Al incluir las Artes en el modelo educativo STEAM, se promueve que el pensamiento computacional se utilice para crear proyectos estéticos o narrativos que exploren la condición humana.

Recurso de Formación socioemocional

La articulación de la UAC de Pensamiento computacional con los Recursos de formación socioemocional es fundamental en el modelo de la Nueva Escuela Mexicana (NEM) y el currículo de la UAS, ya que el aprendizaje de la tecnología y la programación no solo desarrolla capacidades técnicas, sino que fortalece la dimensión humana y social del estudiante.

Implementar la transversalidad con los ámbitos de Formación socioemocional significa formar estudiantes conscientes responsables y empáticos, capaces de autorregularse, establecer relaciones sanas y usar la tecnología de manera ética y reflexiva con su vida cotidiana y académica. Esto mediante el fortalecimiento de la identidad, la autoestima, la toma de decisiones y la convivencia, potenciada por la expresión personal y colectiva en redes sociales y entornos digitales. De manera que enseñar a usar tecnologías con conciencia, ética, respeto y responsabilidad se invita a participar en comunidades digitales seguras e inclusivas.

Aunado a ello, aprender pensamiento computacional implica enfrentarse a errores constantes (fallos en la lógica o sintaxis), esto se articula con el bienestar emocional, el proceso de prueba y error en la robótica educativa y la programación es un escenario ideal para que el estudiante aprenda a gestionar la frustración y desarrolle perseverancia y resiliencia.

Además se busca que el alumno se sienta capaz de enfrentar retos complejos, fortaleciendo su autoestima y seguridad personal al lograr que un prototipo o código funcione. Al involucrarse en proyectos que impactan su entorno (como los sugeridos en las progresiones de automatización), el alumno fortalece su sentido de pertenencia y compromiso con su comunidad.

También se insta a los docentes a fomentar la capacidad creativa del bachiller, y el diseño de soluciones originales a problemas reales permite que el estudiante descubra vocaciones y talentos, conectando sus aprendizajes con su Proyecto de vida.

V. Recomendaciones para el trabajo en el aula y escuela

El programa de estudio de Pensamiento computacional que se fundamenta en los principios del MCCEMS de la NEM y del Modelo Educativo UAS 2022, planteados desde una visión humanista y centrado en la formación integral de los estudiantes, propone que, a través de metodologías activas como el aprendizaje basado en problemas, gamificación, aula invertida, entre otros, los bachilleres se responsabilicen de su proceso de aprendizaje.

Aquí se presentan algunas recomendaciones para el trabajo en el aula y la escuela que facilitan la implementación efectiva del programa:

1. Promover un clima favorable para el aprendizaje, el respeto, la colaboración y la apertura ante la expresión de ideas.
2. Desarrollar estrategias que se basen en la investigación y el descubrimiento de conocimientos y experiencias, desde una postura crítica y reflexiva.
3. Diseñar actividades que contribuyan al desarrollo de habilidades comunicativas, capacidad creativa y pensamiento algorítmico.
4. Utilizar herramientas digitales para crear, seleccionar, procesar, analizar y sistematizar información, asumiendo una postura ética y responsable ante las acciones realizadas.
5. Explorar nuevas formas de trasmitir conocimiento, creando contenido para innovar la práctica docente.
6. Construir en colegiado proyectos multidisciplinarios y transdisciplinarios para coadyuvar al enriquecimiento de la experiencia del estudiante. Considera que Pensamiento computacional se puede vincular con todas las asignaturas y con ello se facilita la integración de toda la comunidad escolar.
7. Implementar estrategias de evaluación para dar seguimiento y mejora continua de los estudiantes. Además, de utilizar diversos tipos de evaluación para lograr una valoración objetiva de su desempeño.
8. Tener en cuenta que la retroalimentación es de carácter motivacional y realizarla de manera oportuna pues es un aspecto clave de la evaluación formativa.
9. Atender oportunamente las necesidades de los estudiantes mediante asesorías.
10. Fomentar en el estudiantado la capacidad de adaptación, esto les permitirá saber actuar ante diversos contextos y circunstancias.

Las recomendaciones anteriores son solo un referente para el docente, por lo que, pueden sentirse en libertad de elegir otras estrategias activas a fin de promover la formación integral de los estudiantes y con ello desarrollar sus capacidades analíticas, críticas y colaborativas.

VI. Evaluación formativa del aprendizaje

En todo programa de estudios, los elementos y procesos que intervienen en el acto educativo recobran especial interés. Uno de ellos es la evaluación de los aprendizajes, con características de ser objetiva, válida, confiable y significativa, tanto para el estudiante como para el docente.

Evaluar la calidad del proceso de aprendizaje, solicita ir más allá de la asignación de una calificación para determinar un buen o mal desempeño del estudiante, el MCCEMS (SEP, 2024) expone la importancia de pasar de una evaluación del aprendizaje a una evaluación para el aprendizaje, es decir, realizar esta acción durante el proceso de aprendizaje y no hasta el final, dando la oportunidad de ajustar la enseñanza considerando las necesidades de los estudiantes. A su vez Frade (2008) señala que la evaluación permite identificar, qué se logró y qué falta por hacer y, sobre todo, en qué se tiene que centrar para que el estudiante mejore su desempeño; es decir, la evaluación es una oportunidad de aprendizaje.

Ante estas premisas queda claro que la evaluación debe ser un proceso continuo, que permita recabar evidencias pertinentes sobre el logro de los aprendizajes, para retroalimentar el proceso de enseñanza-aprendizaje y mejorar sus resultados. Asimismo, es necesario tener en cuenta la diversidad de formas y ritmos de aprendizaje de los estudiantes, para considerar que las estrategias de evaluación atiendan los diferentes estilos de aprendizaje.

El principal propósito pedagógico de la evaluación es el de ayudar al profesor a comprender mejor lo que los estudiantes saben y a tomar decisiones significativas.

La metodología de evaluación de aprendizajes de la UAC Pensamiento computacional incluye la evaluación diagnóstica, formativa y sumativa; es decir una evaluación de contexto, de procesos y resultados (Context Input Process and Products) (Stufflebeam, D. & Shinkfield, A., 1987).

La evaluación diagnóstica

La evaluación diagnóstica o inicial, revela al maestro los logros o las deficiencias de los estudiantes en el proceso de aprendizaje precedente y, le permite determinar las direcciones fundamentales en las que debe trabajarse, así como los cambios en los métodos y estrategias de enseñanza.

Este diagnóstico se hace en diferentes momentos del proceso, ya sea respecto a conocimientos previos necesarios para abordar con éxito una progresión, como para conocer el punto de partida del estudiante y favorecer sus aprendizajes y, en consecuencia, el docente tome decisiones pedagógicas.

La evaluación formativa

La evaluación formativa tiene que ver con la comprensión, regulación y mejora de la situación de enseñanza-aprendizaje; en ese sentido se evalúa para obtener información que permita en un momento determinado saber qué pasó con las estrategias de enseñanza y cómo es que está ocurriendo el aprendizaje de los estudiantes, para que en ambos casos sea posible realizar las mejoras y ajustes necesarios (Díaz Barriga, 2002: p.354), propios de la retroalimentación, tomando en cuenta frecuencia y momentos, toma de decisiones en cuanto a qué y cuáles aspectos a focalizar, diversificación de medios y recursos y, el tipo de retroalimentación, si es individual o colectiva.

La retroalimentación se debe considerar como un actividad necesaria en tanto que le aporta al docente y al estudiante un mecanismo de autocontrol que les permitirá la regulación y el conocimiento de los factores problema que llegan a promover o perturbar dicho proceso (Díaz Barriga, 2002:p.352), una regulación durante todo el periodo en el que se extiende el proceso de enseñanza-aprendizaje, que proporcione ayudas en el momento en que se detectan los problemas, planteando actividades de refuerzo o ampliación, reorientando las secuencias didácticas de aprendizaje donde sea necesario; relacionando todo esto con las metas, como lo establece el MCCEMS haciendo más efectivo el proceso. De acuerdo con Hattie & Timperley (2007) citado por el MCCEMS (SEPb, 2024) la retroalimentación es el elemento central de la evaluación formativa.

La evaluación sumativa o final

La evaluación sumativa o final tiene como objetivo dar cuenta del grado de logro de las metas propuestas, de los resultados globales al finalizar, ya sea una progresión de aprendizaje o de un aprendizaje de trayectoria, con fines que derivan en una calificación, promoción y acreditación. Si bien es cierto la evaluación sumativa tiene la función social de promoción, acreditación de aprendizajes no debemos de descartar que siga siendo un proceso formativo.

Pensamiento computacional ha diseñado un modelo de evaluación tomando en cuenta algunas directrices centrales, en este parte se aborda el cómo de la evaluación, es decir, acerca de cómo los estudiantes saben, interpretan y pueden hacer. El grado en que los alumnos han construido gracias a la ayuda pedagógica recibida y al uso de sus propios recursos cognitivos.

Se implementarán distintos tipos de actividades que guiarán el proceso de aprendizaje. Algunas de ellas servirán de activación para que el estudiante recupere sus saberes previos y sea consciente de los temas que requiere reforzar, otras actividades representarán una ponderación en su evaluación, y algunas de tipo alternativo para mejorar su evaluación en casos particulares. También se facilitan

actividades que llevan al estudiante a la reflexión y autocritica que les ayudaran a ser conscientes de su propio aprendizaje.

En la siguiente tabla se describen solo las evidencias que tienen asignado un valor en el calificador.

Evaluación/calificación				
Aspecto por evaluar	Actividades didácticas sugeridas	Instrumento/ método	Ponderación	Ponderación global
Progresión 1				
Participación en clase	Trabajo colaborativo	Guía de observación	10%	
Actividades de aprendizaje	Preguntas de Identificación del problema	Calificación directa	7%	15%
	Tabla descomposición del problema		8%	
	Tabla reconocimiento de patrones		9%	
	Tabla de abstracción		10%	
	Diseño de algoritmo		15%	
	Tabla implementación y evaluación		11%	
Actividad de evaluación	Algoritmo	Lista de cotejo	30%	
Progresión 2				
Participación en clase	Trabajo colaborativo	Guía de observación	10%	
Actividades de aprendizaje	Algoritmo: secuencial	Calificación directa	5%	25%
	Algoritmo: condicional Simple y Doble		7%	
	Algoritmo: condicional Anidada		7%	
	Algoritmo: condicional Según...hacer		7%	
	Algoritmo: repetitiva Mientras...hacer		7%	
	Algoritmo: condicional Repetir...hasta		7%	
	Algoritmo: Repetitiva Para...hasta		7%	
	Tabla comparativa		3%	
Actividad de evaluación	Algoritmo	Lista de cotejo	40%	
Progresión 3				
Participación en clase	Trabajo colaborativo	Guía de observación	10%	
Actividades de aprendizaje	Programa: condicional Simple y Doble	Calificación directa	11%	20%
	Programa: condicional Anidadas		13%	
	Programa: repetitiva Para		13%	

	Programa: repetitiva Haz mientras		13%	
Actividad de evaluación	Programa	Lista de cotejo	40%	
Progresión 4				
Participación en clase	Trabajo colaborativo	Guía de observación	10%	
Actividades de aprendizaje	Programa: lectura y escritura en arreglo	Calificación directa	10%	15%
	Programa: búsqueda lineal		20%	
	Programa: ordenamiento		20%	
Actividad de evaluación	Programa	Lista de cotejo	40%	
Progresión 5				
Participación en clase	Trabajo colaborativo	Guía de observación	10%	
Actividades de aprendizaje	Tabla comparativa	Calificación directa	2%	25%
	Sistema simulado: encendido LED		10%	
	Sistema simulado: semáforo simple		10%	
	Sistema simulado: encendido LED en oscuridad		14%	
	Sistema simulado: ventilador automático con sensor de temperatura		14%	
Actividad de evaluación	Sistema simulado: alarma inteligente	Lista de cotejo	40%	
Total de evaluación				100%

VII. Recursos didácticos

Los recursos didácticos son medios de apoyo pedagógico que refuerzan la actuación del docente y optimizan el proceso de enseñanza-aprendizaje respondiendo a las exigencias educativas, motivando y despertando el interés de los estudiantes a la vez que fortalecen su aprendizaje. Articulan los contenidos teóricos con las prácticas. Su importancia radica en la influencia que ejercen los estímulos a los órganos sensoriales en quien aprende, en este caso situando al estudiante con el objeto de aprendizaje.

Ahora bien, para que los recursos didácticos sean realmente de utilidad, se deben considerar algunas funciones, como: proporcionar información, cumplir un objetivo, guiar el proceso de enseñanza y aprendizaje, contextualizar a los estudiantes, factibilizar la comunicación entre docentes y estudiantes, acercar las ideas a los sentidos y, motivar a los estudiantes.

En este programa se sugiere para el desarrollo de cada progresión de aprendizaje la realización de diferentes actividades didácticas, que en su mayoría deben ser elaboradas en un equipo de cómputo o algún dispositivo móvil; aunado a ello, la población estudiantil del Bachillerato de la UAS representa una generación marcada por el uso del Internet, lo que lleva a la necesidad de diseñar recursos didácticos que favorezcan la interacción del estudiante con los temas de la asignatura.

Es así como para la óptima implementación de Pensamiento computacional se dispone de varios recursos diseñados por docentes expertos en Tecnologías de la Información, Comunicación, Conocimientos y Aprendizajes Digitales que impartirán esta asignatura.

El repositorio está conformado por videotutoriales, videos, infografías, presentaciones y contenidos interactivos, asimismo de cuestionarios automatizados, que son alojados en la plataforma virtual institucional, en la red social de YouTube, en el sitio web de la Academia de Informática y en otros espacios accesibles para el estudiante, lo que supone un gran avance en la didáctica al permitir procesos de aprendizaje autónomos en los que se consolidan los principios del aprender a aprender.

Es importante resaltar que se consideran herramientas digitales aplicables a diferentes circunstancias, problemáticas y temas que serán el medio de desarrollo de las progresiones, entre ellas, herramientas digitales para la resolución de problemas estructurados cotidianos y para simular sistemas de robótica.

Ofrecer una amplia gama de recursos didácticos que son aplicables a diversas circunstancias y temas se contribuye al aprendizaje en las áreas del conocimiento, en los otros recursos sociocognitivos y en los recursos de formación socioemocional.

VIII. Referencias bibliográficas consultadas para elaborar el programa:

- De Anda, C., Santiago, R., & Romero, E. (2019). Tecnologías de la información 3: Laboratorio de cómputo III. Dirección General de Escuelas Preparatorias-UAS. Ed. Ediciones GYROS, S. A. de C. V. México.
- De Anda, C., Santiago, R., & Romero, E. (2019). Introducción a la programación: Laboratorio de cómputo IV. Dirección General de Escuelas Preparatorias-UAS. Ed. Ediciones GYROS, S. A. de C. V. México.
- Díaz Barriga, A. & Hernández, G. (2002). Estrategias docentes para un aprendizaje significativo. Editorial McGraw-Hill interamericana. México.
- Frade, L. (2008). La evaluación por competencias. Laura Gloria Frade Rubio. México.
- SEP (2019). La Nueva Escuela Mexicana: principios y orientaciones pedagógicas. Subsecretaría Educación Media Superior1 Consultado el 25 de enero del 2024 en:
<https://educacionmediasuperior.sep.gob.mx/work/models/sems/Resource/13516/1/images/NEMprincipiosyorientacionpedagogica.pdf>
- SEP (2023a). Progresiones de aprendizaje del recurso sociocognitivo Cultura digital. SEMS. Secretaría de Educación Pública, Subsecretaría de Educación Media Superior. Segunda edición. Consultado el 18 de diciembre del año 2023 en:
[https://educacionmediasuperior.sep.gob.mx/work/models/sems/Resource/13634/1/images/Progresiones%20de%20aprendizaje%20-%20Cultura%20Digital\(1\).pdf](https://educacionmediasuperior.sep.gob.mx/work/models/sems/Resource/13634/1/images/Progresiones%20de%20aprendizaje%20-%20Cultura%20Digital(1).pdf)
- SEP (2023b). Orientaciones Pedagógicas del recurso sociocognitivo Cultura digital. SEMS. Secretaría de Educación Pública Subsecretaría de Educación Media Superior. Consultado el día 5 de enero del 2024 en:
<https://educacionmediasuperior.sep.gob.mx/work/models/sems/Resource/13634/1/images/Orientaciones%20pedag%C3%A9gicas%20-%20Cultura%20Digital.pdf>
- SEP (2023c). Programa de estudios del Recurso Sociocognitivo de Cultura Digital I. Subsecretaría de Educación Media Superior. Consultado el 10 de diciembre del 2023. en:
<https://educacionmediasuperior.sep.gob.mx/work/models/sems/Resource/13634/1/images/Cultura%20Digital%20I.pdf>
- SEP (2023d). Acuerdo número 09/08/23 por el que se establece y regula el Marco Curricular Común de la Educación Media Superior. Diario Oficial de la Federación.

SEP (2024a). Programa Aula, Escuela y Comunidad PAEC. SEMS. Subsecretaría de Educación Media Superior. Segunda edición. Consultado el 20 de febrero del 2024 en:

[https://educacionmediasuperior.sep.gob.mx/work/models/sems/Resource/13634/1/images/Programa%20Aula,%20Escuela%20y%20Comunidad\(PAEC\),%202da_Edition.pdf](https://educacionmediasuperior.sep.gob.mx/work/models/sems/Resource/13634/1/images/Programa%20Aula,%20Escuela%20y%20Comunidad(PAEC),%202da_Edition.pdf)

SEP (2024b). Evaluación formativa en el MCCEMS. Secretaría de Educación Pública. Subsecretaría de Educación Media Superior. Primera edición Consultado el día 15 de mayo del 2024 en:

https://educacionmediasuperior.sep.gob.mx/work/models/sems/Resource/13634/1/images/Evaluacion_formativa%20en%20el%20MCCEMS.pdf

SEP (2024c) Transversalidad en el MCCEMS. Secretaría de Educación Pública. Subsecretaría de Educación Media Superior. Primera edición. Consultado el 12 de marzo del 2024 en:

https://educacionmediasuperior.sep.gob.mx/work/models/sems/Resource/13634/1/images/La_Transversalidad_en_el_MCCEMS_final.pdf

Stufflebeam, D. & Shinkfield, A. (1987). Evaluación sistémica: Guía teórica y práctica. Ediciones Paidós Ibérica, S.A. España.

UAS (2024). Currículo del Bachillerato DGEP-UAS. Culiacán Rosales, Sinaloa.

UAS (2022). Modelo educativo Universidad Autónoma de Sinaloa.

IX. Anexos

Anexo 1. Instrumentos de heteroevaluación para productos finales por progresión.

Progresión 1. Bases del pensamiento computacional				
Indicador	Si	No	Puntos	
Identifica qué es el pensamiento computacional y lo aplica en la representación de soluciones a problemas cotidianos mediante algoritmos básicos (pseudocódigo, diagramas de flujo), considerando su contexto y recursos disponibles.			1	
Soluciona problemas de su entorno utilizando el pensamiento y lenguaje algorítmico.			1	
Metas	<ul style="list-style-type: none"> Identificar los principios del pensamiento computacional, su descomposición, abstracción y patrones para diseñar, implementar y evaluar algoritmos de problemas de su vida cotidiana. Representar la solución de problemas mediante pensamiento algorítmico seleccionando métodos, diagramas o técnicas. Aplicar lenguaje algorítmico utilizando medios digitales para resolver situaciones o problemas del contexto. 			
Identificar las necesidades, restricciones y objetivos que debe cumplir el reto.			3	
Aplica la fase de descomposición.			2	
Reconoce los patrones del problema.			2	
Realiza la abstracción, conservando únicamente lo esencial del problema.			1	
Diseña el algoritmo adecuado para resolver el reto.			1	
Implementa y evalúa los resultados con una corrida de escritorio según los valores de la tabla.			1	

Progresión 2. Algoritmia en IDE				
Indicador	Si	No	Puntos	
Resuelve problemas cotidianos y académicos mediante la construcción de algoritmos en IDE, utilizando estructuras de control decisivas e iterativas para automatizar procesos y validar la solución de manera digital.			1	
Soluciona problemas de su entorno utilizando el pensamiento y lenguaje algorítmico.			1	
Metas	<ul style="list-style-type: none"> Identificar situaciones de la vida cotidiana que pueden resolverse de manera más eficiente utilizando secuencias y ciclos. Comprobar la lógica y funcionamiento de algoritmos para representar sus soluciones mediante IDE corrigiendo errores y optimizando el código. 			
El algoritmo solicita correctamente los tres datos de entrada (nombre, tasa de aprendizaje, rondas).			1	
Se muestra un mensaje inicial indicando el nombre del modelo y el inicio del entrenamiento.			1	
Se calcula correctamente la precisión en cada ronda considerando los tres factores: avance constante, progreso acumulativo y fatiga.			3	
Se muestra la precisión actual en cada ronda con formato claro y comprensible.			2	
Se incluye una condición final que evalúa si el modelo está listo ($\geq 90\%$) o necesita ajustes.			2	

El código está bien estructurado, con buena identificación y uso adecuado de variables.			1
---	--	--	---

Progresión 3. Programación estructurada en lenguaje: Estructuras de control				
Progresión	Codifica instrucciones en un lenguaje de programación estructurada empleando estructuras de control secuenciales y repetitivas para determinar el orden lógico y eficiente en que se ejecutan en la resolución de problemas.			
Aprendizaje de trayectoria	Soluciona problemas de su entorno utilizando el pensamiento y lenguaje algorítmico.			
Metas	<ul style="list-style-type: none"> Distinguir la sintaxis básica de C++ y la utilidad de las estructuras de control para organizar la ejecución de instrucciones. Representar soluciones a problemas cotidianos y académicos mediante algoritmos que incorporan estructuras de control secuenciales y repetitivas. Codificar, compilar y ejecutar programas en C++ validando su funcionamiento y corrigiendo errores en el uso de estructuras de control. 			
Indicador	Si	No	Puntos	
Aplican correctamente las fases del pensamiento computacional.			1	
Utiliza las librerías adecuadas.			1	
Declara correctamente cada variable de acuerdo con el tipo de datos.			1	
Codifican las instrucciones en C++.			2	
Seleccionaron las estructuras de control adecuadas para resolver de manera óptima el problema.			2	
Utilizan los operadores aritméticos, lógicos y de relación de acuerdo con el algoritmo.			1	
No presenta errores de compilación.			1	
Resuelve de manera óptima el problema mediante lenguaje C++.			1	

Progresión 4. Programación estructurada en lenguaje: Estructuras de datos				
Progresión	Codifica en C++ arreglos unidimensionales para almacenar, procesar y manipular conjunto de datos, determinando la ejecución de instrucciones de manera organizada y eficiente en la resolución de problemas.			
Aprendizaje de trayectoria	Soluciona problemas de su entorno utilizando el pensamiento y lenguaje algorítmico.			
Metas	<ul style="list-style-type: none"> Identificar la utilidad de los arreglos unidimensionales en la resolución de problemas que requieren manejar múltiples valores del mismo tipo de dato. Representar soluciones a problemas cotidianos y académicos mediante el diseño de algoritmos que emplean arreglos unidimensionales. Codificar y ejecutar programas en C++ que utilizan arreglos unidimensionales para almacenar, recorrer y procesar datos (suma, promedio, máximo, mínimo, búsqueda lineal y ordenamiento). 			
Indicador	Si	No	Puntos	
El algoritmo solicita correctamente los datos de entrada: número de jugadores, puntuaciones y puntuación a buscar.			1	

Se muestra un mensaje inicial indicando el propósito del programa (organizar ranking y buscar puntuación).			1
Se ordenan correctamente las puntuaciones de mayor a menor.			3
Se busca correctamente la puntuación deseada y se determina si está en el ranking.			2
Se muestra la posición de la puntuación buscada con un mensaje claro y motivador.			2
Se muestra un mensaje adecuado si la puntuación no se encuentra en el ranking.			1

Progresión 5. Robótica educativa			
Progresión	Simula sistemas robóticos mediante aplicaciones gráficas y la programación en Arduino, implementando estructuras básicas de código (setup y loop), funciones elementales (pinMode, digitalWrite, delay), control de salidas múltiples mediante ciclos y retardos, así como el uso de sensores y actuadores para resolver problemas simples de automatización.		
Aprendizaje de trayectoria	<ul style="list-style-type: none"> Soluciona problemas de su entorno utilizando el pensamiento y lenguaje algorítmico. Diseña y elabora contenidos digitales mediante técnicas, métodos, y recursos tecnológicos para fortalecer su creatividad e innovar en su vida cotidiana. 		
Meta	<ul style="list-style-type: none"> Identificar la importancia de la robótica educativa como herramienta para comprender la interacción entre hardware y software. Configurar sistemas básicos de automatización en un entorno gráfico controlando las salidas. Codificar programas con estructuras de control, funciones básicas, bucles, sensores y actuadores en simuladores. 		
Indicador	Si	No	Puntos
El circuito está correctamente armado en <i>Tinkercad</i> (<i>Arduino</i> , <i>protoboard</i> , sensor y actuadores bien colocados).			2
Las conexiones eléctricas son correctas (polaridad, resistencias, GND/5V, entrada del sensor).			2
El código configura correctamente los pines en <i>setup()</i> y realiza la lectura del sensor.			2
La alarma responde adecuadamente a la condición definida (se activa y desactiva según el sensor).			2
El documento entregado contiene la imagen del circuito, el código completo y una breve explicación del funcionamiento.			2

Anexo 2. Instrumentos de Autoevaluación por bloques.

Bloque 1				
Metas	Criterios	Nivel de dominio		
		Sí lo logro	En proceso	Aún no lo logro
Identifica los principios del pensamiento computacional, su descomposición, abstracción y patrones para diseñar, implementar y evaluar algoritmos de problemas de su vida cotidiana.	Identifico los principios del pensamiento computacional.			
	Aplico cada una de las fases para resolver problemas cotidianos.			
Representa la solución de problemas mediante pensamiento	Organizo los pasos del algoritmo de manera clara y lógica.			

algorítmico seleccionando métodos, diagramas o técnicas.	Represento soluciones mediante pseudocódigo y diagramas de flujo.			
Aplica lenguaje algorítmico utilizando medios digitales para resolver situaciones o problemas del contexto.	Traduzco mis soluciones algorítmicas a pseudocódigo en un entorno digital.			
	Ejecuto mis algoritmos en PSeInt para validar su funcionamiento.			
Identifica situaciones de la vida cotidiana que pueden resolverse de manera más eficiente utilizando secuencias y ciclos.	Reconozco problemas que pueden resolverse con secuencias y ciclos.			
	Selecciono la estructura de control adecuada (secuencia, condicional o repetitiva) en la resolución de problemas.			
Comprueba la lógica y funcionamiento de algoritmos para representar sus soluciones mediante IDE corrigiendo errores y optimizando el código.	Compruebo el funcionamiento de mis algoritmos mediante la ejecución en el IDE.			
	Identifico errores de sintaxis, de lógica y de ejecución en PSeInt.			
Lo mejor que aprendí fue:				
Lo que necesito reforzar es:				
Calificación que doy a mi desempeño:	Excelente	Satisfactorio	En desarrollo	Inicial

Bloque 2				
Metas	Criterios	Nivel de dominio		
		Sí lo logro	En proceso	Aún no lo logro
Distingue la sintaxis básica de C++ y la utilidad de las estructuras de control para organizar la ejecución de instrucciones.	Identifico la sintaxis básica de C++ (declaración de variables, operadores, entradas y salidas).			
	Reconozco la utilidad de las estructuras de control selectivo y repetitivo.			
Representa soluciones a problemas cotidianos y académicos mediante algoritmos que incorporan estructuras de control secuenciales y repetitivas.	Selecciono la estructura de control adecuada según el problema planteado.			
	Explico en qué casos conviene utilizar cada tipo de estructura de control.			
Codifica, compila y ejecuta programas en C++ validando su funcionamiento y corrigiendo errores en el uso de estructuras de control.	Traduzco un algoritmo a código C++ utilizando correctamente las estructuras de control.			
	Compilo y ejecuto programas verificando que funcionen según lo esperado.			
Identifica la utilidad de los arreglos unidimensionales en la resolución de problemas que requieren manejar múltiples valores del mismo tipo de datos.	Reconozco problemas que requieren almacenar muchos valores del mismo tipo.			
	Identifico cuándo es más eficiente usar un arreglo en lugar de variables aisladas.			
Representa soluciones a problemas cotidianos y académicos mediante el diseño de algoritmos que emplean arreglos unidimensionales.	Diseño algoritmos que incluyen la declaración, lectura y uso de arreglos unidimensionales.			
	Estructuro los pasos para recorrer un arreglo (búsqueda, conteo, acumulación, etc.).			
Codifica y ejecuta programas en C++ que utilizan arreglos unidimensionales para almacenar, recorrer y procesar datos (suma, promedio, máximo, mínimo, búsqueda lineal y ordenamiento).	Declaro y utilizo correctamente arreglos unidimensionales en C++.			
	Implemento operaciones de procesamiento: suma, promedio, máximo, mínimo.			
	Realizo búsqueda lineal en un arreglo.			
Lo mejor que aprendí fue:				
Lo que necesito reforzar es:				
Calificación que doy a mi desempeño:	Excelente	Satisfactorio	En desarrollo	Inicial

Metas	Criterios	Nivel de dominio		
		Sí lo logro	En proceso	Aún no lo logro
Identifica la importancia de la robótica educativa como herramienta para comprender la interacción entre hardware y software.	Explico qué es la robótica educativa y su utilidad en el aprendizaje tecnológico.			
	Reconozco la relación entre hardware y software en proyectos de robótica.			
	Describo la forma en que Tinkercad y Arduino representan la interacción hardware-software.			
Configura sistemas básicos de automatización en un entorno gráfico controlando las salidas.	Configuro circuitos básicos en Tinkercad con componentes electrónicos.			
	Conecto correctamente salidas (actuadores) como LED, servo o buzzer en Arduino.			
	Utilizo el simulador para verificar que las salidas reaccionen según el diseño.			
	Comprendo el flujo de energía y señal entre la placa Arduino y los actuadores.			
	Ajusto parámetros (ángulo de servo, intensidad, frecuencia) cuando el circuito lo requiere.			
Codifica programas con estructuras de control, funciones básicas, bucles, sensores y actuadores en simulaciones.	Escribo código Arduino que controla salidas mediante <i>digitalWrite</i> , <i>analogWrite</i> o funciones específicas.			
	Implemento estructuras de control para responder a condiciones del entorno.			
	Programo sensores básicos (LDR, ultrasonido, botón, potenciómetro) e interpreta sus lecturas.			
	Integro sensores y actuadores en una simulación que responda a entradas reales.			
	Depuro y corrojo errores en el código hasta lograr una simulación funcional.			
Lo mejor que aprendí fue:				
Lo que necesito reforzar es:				
Calificación que doy a mi desempeño:	Excelente	Satisfactorio	En desarrollo	Inicial

Anexo 3. Instrumentos de coevaluación por bloques.

Bloque 1			
Buen trabajo (3)	Algo nos faltó (2)	Debemos mejorar (1)	Evaluación
Organizamos el trabajo estipulando tareas, prioridades y plazos.	Se organizó el trabajo, pero no se estipularon tareas, prioridades o el plazo de entrega final.	No hubo organización para realizar nuestros trabajos.	
Cumplimos cada uno con las tareas asignadas en el plazo estipulado.	Casi todos los miembros del equipo cumplimos con las tareas asignadas y el plazo estipulado; teniendo que resolver lo que a otros les fue encomendado.	Un solo miembro del equipo realizó todos los productos.	
Todos participamos activamente en la elaboración de los productos.	Casi todos los miembros del equipo participamos activamente en la elaboración de los productos.	No hubo participación de los miembros del equipo en la elaboración de los productos.	
La calidad de los productos que elaboramos fue la adecuada para su entrega.	La calidad de los productos que elaboramos fue en su mayoría la adecuada para su entrega.	No se cumplió con la calidad adecuada de los productos para su entrega.	
Total			de 12

Bloque 2			
Buen trabajo (3)	Algo nos faltó (2)	Debemos mejorar (1)	Evaluación
Organizamos el trabajo estipulando tareas, prioridades y plazos.	Se organizó el trabajo, pero no se estipularon tareas, prioridades o el plazo de entrega final.	No hubo organización para realizar nuestros trabajos.	
Cumplimos cada uno con las tareas asignadas en el plazo estipulado.	Casi todos los miembros del equipo cumplimos con las tareas asignadas y el plazo estipulado; teniendo que resolver lo que a otros les fue encomendado.	Un solo miembro del equipo realizó todos los productos.	
Todos participamos activamente en la elaboración de los productos.	Casi todos los miembros del equipo participamos activamente en la elaboración de los productos.	No hubo participación de los miembros del equipo en la elaboración de los productos.	
La calidad de los productos que elaboramos fue la adecuada para su entrega.	La calidad de los productos que elaboramos fue en su mayoría la adecuada para su entrega.	No se cumplió con la calidad adecuada de los productos para su entrega.	
Total			de 12

Bloque 3			
Buen trabajo (3)	Algo nos faltó (2)	Debemos mejorar (1)	Evaluación
Organizamos el trabajo estipulando tareas, prioridades y plazos.	Se organizó el trabajo, pero no se estipularon tareas, prioridades o el plazo de entrega final.	No hubo organización para realizar nuestros trabajos.	
Cumplimos cada uno con las tareas asignadas en el plazo estipulado.	Casi todos los miembros del equipo cumplimos con las tareas asignadas y el plazo estipulado; teniendo que resolver lo que a otros les fue encomendado.	Un solo miembro del equipo realizó todos los productos.	
Todos participamos activamente en la elaboración de los productos.	Casi todos los miembros del equipo participamos activamente en la elaboración de los productos.	No hubo participación de los miembros del equipo en la elaboración de los productos.	
La calidad de los productos que elaboramos fue la adecuada para su entrega.	La calidad de los productos que elaboramos fue en su mayoría la adecuada para su entrega.	No se cumplió con la calidad adecuada de los productos para su entrega.	
Total			de 12