A picture containing toy

Description automatically generated

**PLAN DE CLASES**

**Título:** La Hora del Código 2021 (TimeCraft)

**Edades:** de 7 en adelante

**Materias:** ciencias de la computación, matemáticas, artes lingüísticas, historia, artes visuales, ciencias, ingeniería

**Habilidades necesarias para el siglo XXI:**creatividad, pensamiento crítico, resolución de problemas

**Idiomas compatibles:** inglés (Estados Unidos), inglés (Reino Unido), alemán, español (España), español (México), francés (Francia), francés (Canadá), italiano, japonés, coreano, portugués (Brasil), portugués (Portugal), ruso, chino (simplificado), chino (tradicional), turco, polaco, indonesio, ucraniano, griego, búlgaro, húngaro, finlandés, danés, sueco, noruego, eslovaco, checo, holandés.

**Multijugador/Un jugador:** Un jugador

**Nivel de experiencia con Minecraft requerido (docente):** principiante

**Descripción de la clase:** tutorial divertido y creativo de introducción a las ciencias de la computación. Viaja en el tiempo y utiliza el código basado en bloques o el código de Python para resolver los percances misteriosos que se presentan en la línea de tiempo de la Tierra. Busca las pistas y usa tus habilidades de pensamiento creativo para deducir quién es el culpable en esta aventura a través del tiempo. Los estudiantes también exploran la integración de las ciencias de la computación en diferentes carreras e intereses personales.

**Tiempo total requerido:** 1 hora

**Estándares educativos:**

|  |  |  |
| --- | --- | --- |
| Estándares | | |
| Estados Unidos | Australia | Reino Unido - Inglaterra |
| Puedo desarrollar modelos de procesos cotidianos al crear y seguir algoritmos para completar tareas  **1A-AP-08** | Puedo seguir, describir y representar una secuencia de pasos y decisiones (algoritmos) necesarios para resolver problemas simples  **ACTDIP004** | Puedo crear y depurar programas simples  **Etapa clave 1** |
| Puedo desarrollar programas con secuencias y bucles simples a fin de expresar ideas o resolver un problema  **1A-AP-10** | Puedo definir problemas simples, así como describir y seguir una secuencia de pasos y decisiones (algoritmos) necesarios para resolverlos  **ACTDIP010** | Puedo entender lo que son los algoritmos, la manera en que se implementan como programas en dispositivos digitales, y el hecho de que los programas se ejecutan al seguir instrucciones precisas e inequívocas  **Etapa clave 1** |
| Depurar (identificar y corregir) errores en un algoritmo o un programa que incluye secuencias y bucles simples.  **1A-AP-14** | Diseñar, modificar y seguir algoritmos simples que incluyen secuencias de pasos, ramificaciones e iteración (repetición)  **ACTDIP019** | Puedo utilizar el razonamiento lógico para predecir el comportamiento de los programas simples  **Etapa clave 1** |
| Puedo comparar la manera en que las personas viven y trabajan, antes y después de la implementación o la adopción de una nueva tecnología de cómputo  **1A-IC-16** | Diseñar algoritmos representados en forma de diagrama y en inglés, y trazar algoritmos para predecir el resultado de una entrada determinada e identificar errores  **ACTDIP029** | Puedo utilizar el razonamiento lógico para explicar la manera en que funcionan algunos algoritmos simples, y detectar y corregir errores en algoritmos y programas  **Etapa clave 2** |
| Puedo crear programas que incluyen secuencias, eventos, bucles y condicionales.  **1B-AP-10** | Puedo definir problemas simples, así como describir y seguir una secuencia de pasos y decisiones (algoritmos) necesarios para resolverlos.  **ACTDIP01** | Puedo utilizar la secuencia, la selección y la repetición en los programas, además de trabajar con variables y varias formas de entradas y resultados  **Etapa clave 2** |
| Puedo comparar y perfeccionar varios algoritmos que realizan la misma tarea y determinar cuál es el más adecuado  **1B-AP-08** | Puedo implementar soluciones digitales simples en forma de programas visuales con algoritmos que incluyen ramificación (decisiones) y entradas de los usuarios  **ACTDIP011** | Puedo diseñar, escribir y depurar programas que alcanzan objetivos específicos, lo que incluye controlar o simular sistemas físicos, y resolver problemas al dividirlos en partes más pequeñas  **Etapa clave 2** |
| Probar y depurar (identificar y corregir errores) un programa o algoritmo para asegurarse de que funcione como se espera.  **1B-AP-15** | Puedo implementar soluciones digitales simples en forma de programas visuales simples que incluyen ramificación iteración (repetición) y entradas de los usuarios  **ACTDIP020** | Puedo entender varios algoritmos clave que reflejan el pensamiento computacional, y utilizar el razonamiento lógico para comparar la utilidad de otros algoritmos como alternativas para el mismo problema  **Etapa clave 3** |
| Puedo hablar sobre las tecnologías de cómputo que han cambiado el mundo y expresar las maneras en que esas tecnologías influyen en, y se ven influidas por, las prácticas culturales  **1B-IC-20** | Puedo implementar y modificar programas con interfaces de usuario que incluyen ramificación, iteración y funciones en un lenguaje de programación para propósitos generales  **ACTDIP030** | Puedo utilizar 2 o más lenguajes de programación, al menos uno de los cuales es textual, para resolver una gran variedad de problemas de cómputo, utilizar adecuadamente las estructuras de datos, y diseñar y desarrollar programas modulares que utilizan procedimientos o funciones  **Etapa clave 3** |
| Puedo utilizar diagramas de flujo o pseudocódigo para visualizar los problemas complejos como algoritmos    **2-AP-10** | Puedo recopilar, utilizar y presentar diferentes tipos de datos por medio de software simple para crear información y resolver problemas  **ACTDIP009** |  |
| Puedo diseñar y desarrollar de forma iterativa programas que combinen diferentes estructuras de control, incluyendo bucles anidados y condicionales compuestas  **2-AP-12** |  |  |
| Puedo dividir los problemas y subproblemas en partes para facilitar el diseño, la implementación y la revisión de programas  **2-AP-13** |  |  |

**Preparación para docentes:**

* Instala Minecraft: Education Edition desde <https://aka.ms/HourofCode2021>
* Mira [el video de introducción](https://aka.ms/HOC2021Trailer) del tema de La Hora del Código de este año.
* Mira [el video de demostración](https://aka.ms/HOC2021Walkthrough) para entender mejor la clase y aprender a utilizar el mundo de Minecraft.
* Descarga la [Guía para docentes](file:///C:\Users\laylahbulman\Downloads\aka.ms\HOC2021educatorguide), que está diseñada para proporcionarte toda la información que necesitas a fin de ayudar a los estudiantes a aprovechar la Hora del Código de forma agradable y exitosa, incluyendo todos los recursos externos y las instrucciones sugeridas para los entornos de aprendizaje guiado, independiente e híbrido/remoto
* Utiliza las [diapositivas de presentación](C:\\Users\\laylahbulman\\Downloads\\aka.ms\\HOC2021presentationslides) para estudiantes a fin de obtener las instrucciones paso a paso
* Descarga una copia de las [soluciones de código](https://aka.ms/HOC2021CodingSolutions).
* ¿Tienes alguna otra duda? Consulta las [preguntas frecuentes](https://aka.ms/HOC2021FAQ).

**Resumen del tema:**

Eres especialista en ciencias de la computación y trabajas para el Instituto de los Grandes Errores en el Tiempo. Tu trabajo es corregir las misteriosas "divisiones del tiempo" que aparecen en la historia y averiguar quién (o qué) las está provocando.

¿Podrás ayudar a reparar las divisiones del tiempo y salvar la historia con tus superpoderes de programación?

¿Lograrás averiguar quién o qué está provocando estos cambios alocados en la historia?

En tu misión de TimeCraft, deberás hacer lo siguiente:

* Viajar al pasado para visitar algunos momentos emocionantes en la historia del mundo
* Programar a tu Agent del tiempo para que repare las divisiones del tiempo
* Utilizar las pistas para identificar al culpable (es decir, la persona o el objeto que está provocando las divisiones del tiempo)

Los estudiantes pueden utilizar código basado en bloques o código de Python.

**Objetivos de aprendizaje:**

* Entender la importancia y los beneficios de las ciencias de la computación en todos los aspectos de la vida.
* Analizar y resolver problemas mediante el pensamiento algorítmico y la división de problemas.
* Practicar diferentes conceptos de las ciencias de la computación, como secuencias, eventos, bucles y depuración.
* Crear soluciones de código para completar una tarea o resolver un problema de manera exitosa.
* Reconocer las conexiones expandidas que ofrecen las ciencias de la computación para otras carreras.

**Actividades de los estudiantes:**

**Actividades de introducción (5 minutos):**

1. Los estudiantes pensarán en las siguientes tres preguntas:

* ¿Qué son las ciencias de la computación?
* ¿Cómo se usan las ciencias de la computación (o las habilidades de informática) en la escuela?
* ¿Cómo se usan las ciencias de la computación en el trabajo (en diferentes empleos)?

1. Los estudiantes revisarán algunos términos clave del glosario que les servirán para entender mejor la experiencia de aprendizaje.

**Actividades de programación (30-40 minutos)**

Los estudiantes comenzarán su recorrido de programación en el portal de la Efímera Uno, que es la nave espacial principal del Instituto de Monitoreo de Errores en el Tiempo. Los estudiantes son especialistas en ciencias de la computación, y deberán trabajar con TARRA, el robot de inteligencia artificial (IA) que administra la nave, y con sus Agents del tiempo, que son robots que pueden viajar en el tiempo hacia diferentes puntos de la historia, para corregir las divisiones del tiempo mediante el uso de código.

Se les pedirá a los estudiantes que seleccionen entre el código basado en bloques y Python para sus actividades de programación, y luego deberán seleccionar a su Agent del tiempo. Nuestra recomendación es que los principiantes comiencen con la programación por bloques.

Después de seleccionar a su Agent, los estudiantes deberán completar dos desafíos de programación iniciales.

DESAFÍOS DE PROGRAMACIÓN INICIALES

Desafío 1: Llama a tu Agent del tiempo. Habla con TARRA para que te entregue tu dispositivo de comunicación TALK, y llama a tu Agent del tiempo hacia tu ubicación.

Desafío 2: Mover al Agent (w). Mueve a tu Agent hacia adelante para que se pare sobre el bloque verde.

DESAFÍOS DE PROGRAMACIÓN DE LA LÍNEA DE TIEMPO

Los estudiantes recibirán el primer desafío, "Orquesta de jazz", el cual será una experiencia de programación guiada. Los estudiantes podrán utilizar dos pistas para resolver cada actividad de programación antes de que se les muestre la solución. Después de cada experiencia de programación, participarán en una búsqueda de pistas breve para identificar al culpable.

Después de completar el desafío División del tiempo 1: Orquesta de jazz, los estudiantes se teletransportarán de regreso hacia el vestíbulo principal para seleccionar otro desafío. Los estudiantes pueden seleccionar cualquier división del tiempo, pero se recomienda que sigan las divisiones del tiempo que se muestran en la pantalla gigante, sobre la línea de tiempo.

|  |  |
| --- | --- |
| **División del tiempo 1**  Orquesta de jazz  (Este es el desafío guiado) | ¡El gran músico de jazz perdió su trompeta favorita y la reemplazó con un kazoo! Programa a tu Agent del tiempo para que recorra el laberinto y recupere la trompeta del músico para salvar el jazz. |
| **División del tiempo 2**  Las grandes pirámides de Giza | ¡Las grandes pirámides ahora tienen forma de cubos!  Programa a tu Agent del tiempo para que ayude al diseñador a crear una estructura segura que dura miles de años: la pirámide. |
| **División del tiempo 3**  Misión en la Luna | Los astronautas necesitan tu ayuda. ¡Utiliza a tu Agent del tiempo para completar un laberinto de fe y entregar los cálculos que necesitan los astronautas para aterrizar en la Luna! |
| **División del tiempo 4**  La Gran Muralla China | La Gran Muralla aún no es tan grandiosa... ¡De hecho, es muy pequeña!  Los pandas hambrientos se comen todos los andamios de bambú, por lo que los trabajadores no pueden seguir construyendo la muralla y hacerla tan grande como quisieran. ¡Programa un jardín de bambú para distraer a los pandas hambrientos! |
| **División del tiempo 5**  La Mona Lisa | La Mona Lisa ya no sonríe. De hecho, está muy enojada. El culpable del tiempo pisoteó todo el jardín de la Mona Lisa y convirtió su famosa sonrisa en una mueca de disgusto. Programa a tu Agent del tiempo para que le levante el ánimo al plantar más flores. |
| **División del tiempo 6**  Los primeros vuelos | El culpable llenó de agujeros la pista de despegue. En lugar de aviones, ahora solo tenemos globos aerostáticos. ¡Programa a tu Agent del tiempo para que repare la pista a fin de que los inventores puedan hacer su primer vuelo! |
| **División del tiempo 7**  La primera programadora | El primer programa de ciencias de la computación tocaba una canción, pero el culpable arruinó el código. ¡Repara el código con tu Agent del tiempo para que se pueda reproducir la música! |
| **División del tiempo 8**  El mejor amigo del ser humano | El mejor amigo del ser humano ya no es el perro... ¡es un oso! Utiliza a tu Agent del tiempo para hacer amistad con un cachorro y llévalo de regreso con los humanos para que sigan siendo los mejores amigos. |
| **División del tiempo 9**  Acertijo de paleontología | ¡Oh, no! El culpable viajó al pasado y se robó algunos fósiles, por lo que el cuello del braquiosaurio ahora está más corto. ¡Utiliza a tu Agent del tiempo para ayudar a reemplazar los fósiles robados a fin de que los paleontólogos puedan construir al braquiosaurio con su cuello largo! |
| **División del tiempo 10**  Elementos de descubrimiento | El culpable se metió al laboratorio de ciencias y escondió algunos elementos. ¡Utiliza a tu Agent del tiempo para encontrar los elementos escondidos y ayuda al científico a hacer su gran descubrimiento! |

Al completar tres actividades de programación, los estudiantes resolverán el primer ciclo del juego e identificarán al culpable. Podrán extender sus actividades de programación al seleccionar un nuevo Agent del tiempo y programar otras divisiones del tiempo.

**Evaluación:**

* ¿Cuál fue tu parte favorita de la Hora del Código?
* ¿Cuál fue la parte más difícil de la Hora del Código?
* ¿Cómo utilizaste tus habilidades de ciencias de la computación el día de hoy?
* ¿Qué cosa nueva aprendiste hoy?
* ¿Por qué son importantes las ciencias de la computación para todas las personas?
* ¿Te gustaría volver a jugar Minecraft: Education Edition?

**Diferenciación:**

Con el objetivo de ofrecer un mayor acceso a la Hora del Código para los estudiantes, tenemos tres experiencias de aprendizaje diferentes: en el salón de clases con un docente facilitador, en el salón de clases como experiencia autoguiada, o como aprendizaje remoto (virtual). Cada una de estas experiencias de aprendizaje tiene un nivel diferente de soporte de los docentes y modificación, con la finalidad de ayudar a los estudiantes a participar y tener éxito en la Hora del Código.

**Docente facilitador (cara a cara)**

***Este tipo de experiencia de aprendizaje ofrece el mayor nivel de soporte del docente para los estudiantes***. Tu labor como docente es proporcionar instrucciones y modelos explícitos antes de dejar que los estudiantes resuelvan las actividades de programación de la Hora del Código de forma independiente. Utilizarás la [presentación de la Hora del Código](https://aka.ms/HOC2021Presentation) para guiarte y guiar a los estudiantes por los primeros pasos del juego, las actividades de programación y la reflexión sobre lo aprendido.

**Estudiantes autoguiados**

***Este tipo de experiencia de aprendizaje ofrece el menor nivel de soporte del docente para los estudiantes***. Tu labor como docente es asegurarte de que los estudiantes puedan iniciar sesión en la plataforma de Minecraft: Education Edition. Deberás proporcionar a los estudiantes la presentación de la Hora del Código para que la utilicen como referencia durante su experiencia de juego autoguiada a su propio ritmo. La presentación les proporcionará a los estudiantes instrucciones explícitas sobre cómo participar en la Hora del Código. También debes proporcionarles a los estudiantes una copia del glosario visual, el cual se encuentra al final de esta guía para docentes. Se recomienda que les entregues estos materiales a los estudiantes en un medio digital de tu elección, o que imprimas copias en papel para que las puedan utilizar cuando lo necesiten.

**Experiencia de aprendizaje remoto**

Este tipo de experiencia de aprendizaje se puede facilitar de forma sincrónica o asincrónica.

***Si tus estudiantes participan en una sesión virtual sincrónica (en vivo)***, deberás facilitar la sesión como si fuera una experiencia en el salón de clases (con facilitación por parte del docente). Tu labor como docente es proporcionar instrucciones y modelos explícitos antes de dejar que los estudiantes resuelvan las actividades de programación de la Hora del Código de forma independiente. Utilizarás la presentación de la Hora del Código para guiarte y guiar a los estudiantes por los primeros pasos del juego, las actividades de programación y la reflexión sobre lo aprendido. Asegúrate de que tu plataforma de comunicación virtual (por ejemplo, Microsoft Teams) esté lista para compartir contenido y que el sonido esté habilitado.

***Si tus estudiantes participan en una sesión virtual asincrónica (que no es en vivo),*** deberás facilitar la sesión como si fuera una experiencia autoguiada en el salón de clases. Tu labor como docente es asegurarte de que los estudiantes tengan acceso a la plataforma de Minecraft: Education Edition antes de la clase. También deberás proporcionar a los estudiantes la presentación de la Hora del Código para que la utilicen como referencia durante su experiencia de juego autoguiada a su propio ritmo. La presentación les proporcionará a los estudiantes instrucciones explícitas sobre cómo participar en la Hora del Código. También debes proporcionarles a los estudiantes una copia del glosario visual, el cual se encuentra al final de esta guía para docentes. Se recomienda que les entregues estos materiales (presentación y glosario) a los estudiantes en un medio digital de tu elección (por ejemplo: canal de Microsoft Teams, libreta de OneNote, tu sistema de administración de aprendizaje, etc.).

Si planeas ofrecer esta clase por medio de aprendizaje remoto, ten en cuenta estos consejos:

1. Esta clase no está diseñada como una experiencia multijugador. Los estudiantes deberán trabajar en su propia versión del mundo de forma individual.

2. Divide a los estudiantes en parejas o en grupos pequeños mediante el uso de diferentes salas, para que puedan ayudarse mutuamente a resolver dudas mientras resuelven las actividades de programación.

3. Ya que es probable que los estudiantes tengan diferentes niveles de experiencia con la programación y con Minecraft, puede resultar útil asignar líderes de cada grupo de estudiantes para que ayuden a sus compañeros cuando sea necesario.

Si deseas más información sobre el aprendizaje remoto en Minecraft: Education Edition, visita <https://aka.ms/remote-learning-kit>