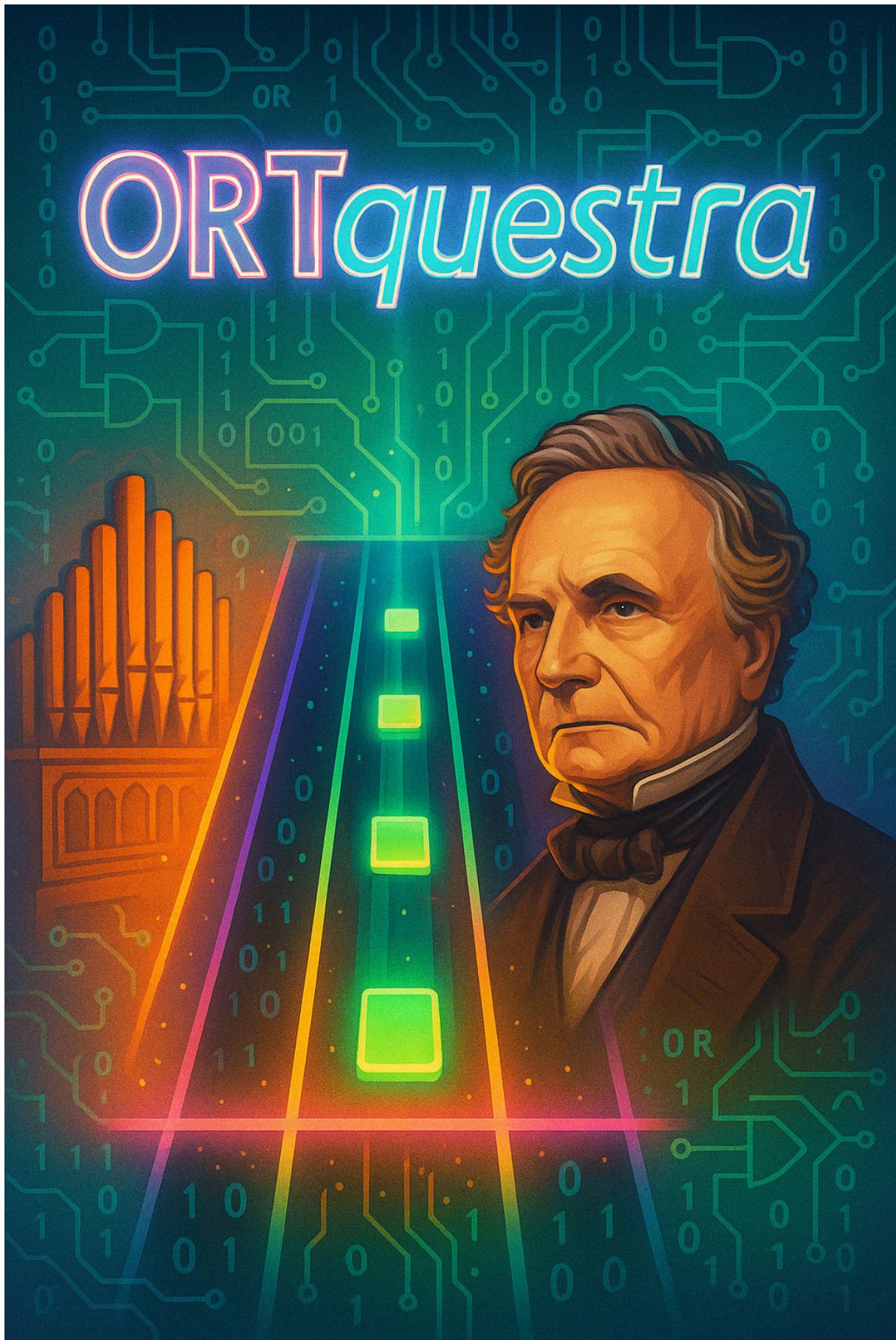
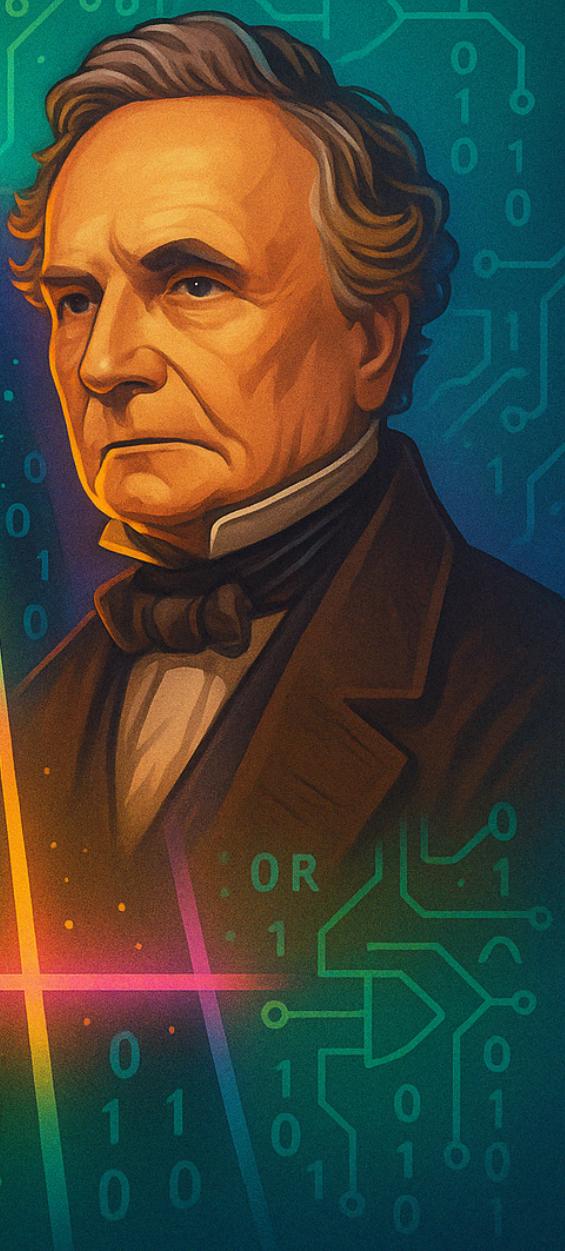


ORTquestra



Contexto

Charles Babbage, padre de la **arquitectura** computacional moderna, diseñaba la **máquina analítica**, un proyecto que exigía prolongadas jornadas de absoluto silencio.

Sin embargo, en su taller, Babbage era constantemente interrumpido por el estruendoso sonido de los **organilleros** que recorrían las calles. Estos músicos callejeros utilizaban una manivela para hacer sonar un instrumento que leía melodías desde **cintas perforadas**, generando música automática y repetitiva.

Es en ese momento que tú, viajero del tiempo, llegás desde el futuro con tu laptop del siglo XXI, decidido a aprender del maestro. Impresionado por tu curiosidad, Babbage accede a enseñarte los principios de la arquitectura digital... pero con una condición: deberás diseñar una máquina, en **Logisim**, capaz de **grabar y reproducir melodías**, imitando el funcionamiento de los organilleros, como si leyera desde hojas perforadas.

Es entonces como tú, su aprendiz moderno y junto a Ada Lovelace, se les pide realizar el diseño de esta máquina en Logisim para ganarle a los organilleros y que Babbage pueda trabajar.

Quizás, con suerte, algún día los planos lleguen a un museo británico y **demuestren que nuestro diseño no era solo teoría... sino que realmente funcionaba**.

[Charles Babbage and the Vengeance of Organ-Grinders](#)

Concepto

Se deberá realizar un circuito que permita simular tres funcionalidad de estado del sistema

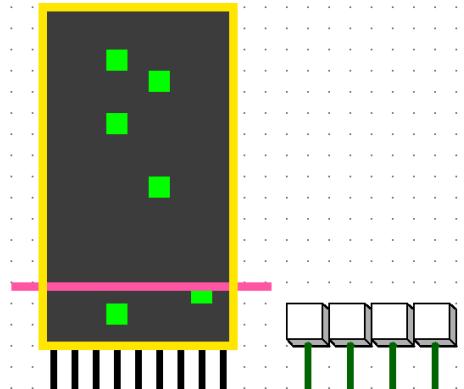
- Juego Aleatorio
- Grabación
- Jugar Grabación

A continuación se definen componentes necesarios y cada funcionalidad requerida.

Componentes Necesarios

Nuestra implementación contará con:

- Pantalla Matriz LED
- 4 Pulsadores para el juego
- Contador binario puntaje (LED rojos)
- Temporizador de 2 minutos (display de 8 segmentos)



Juego

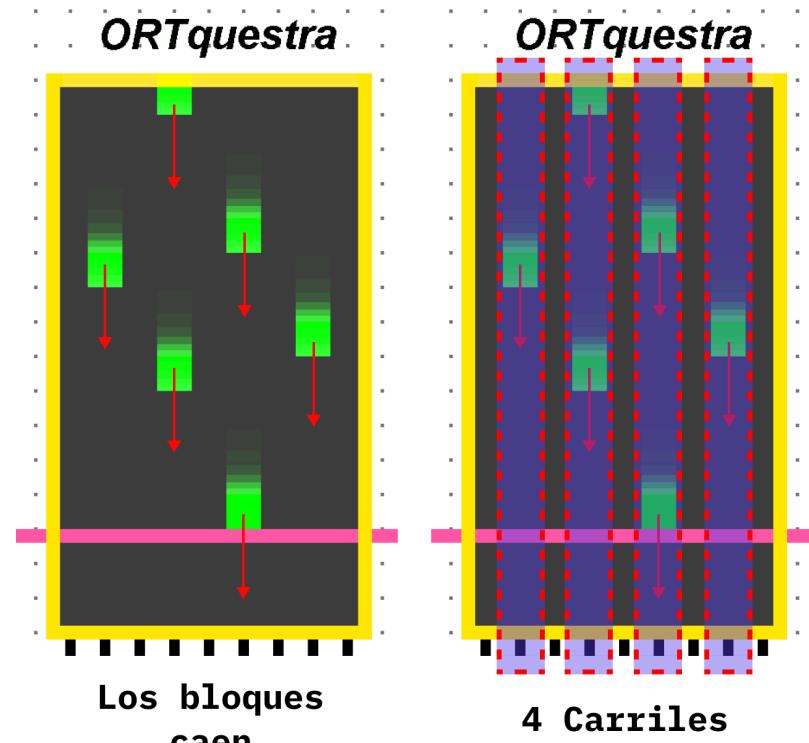
Se comienza con una pantalla en blanco, en la cual instantes después comienzan a caer bloques de arriba hacia abajo. Se cuenta con 4 columnas para esto. En cada “frame” (Instancia en la que se redibuja o cambia al siguiente estado la pantalla), deben caer todos los bloques un lugar.

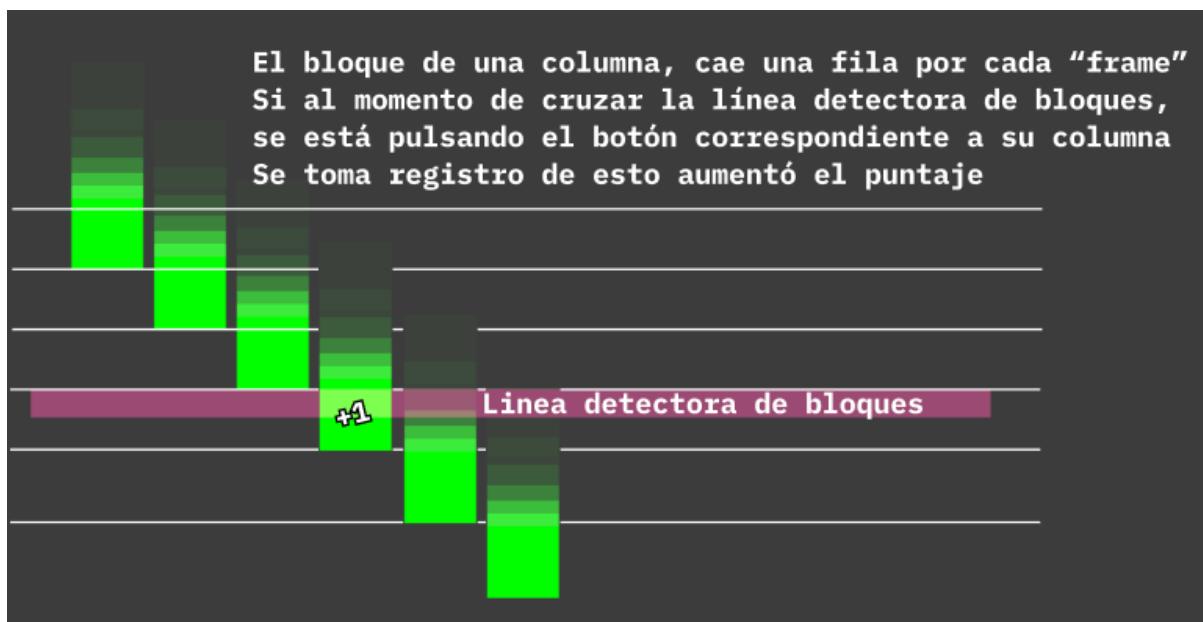
Nótese que por cada fila presentada, no habrá dos bloques presentes, solo puede haber uno o la ausencia de ellos. Se definirá una fila de las ubicadas en el inferior de la pantalla como “línea de acción”, donde cada vez que un bloque que cae de arriba hacia abajo cruce esta fila, será entonces una señal para que el usuario presione uno de los cuatro botones a su disposición. Un botón por columna.

Una interacción similar a esto se puede encontrar aquí

[Link a GIF posteado en Aulas](#)

1. Pantalla en blanco.
2. Cada frame, los bloques caen un paso.
3. La “línea de acción” (fila inferior): al cruzarla, el usuario pulsa el botón de esa columna.

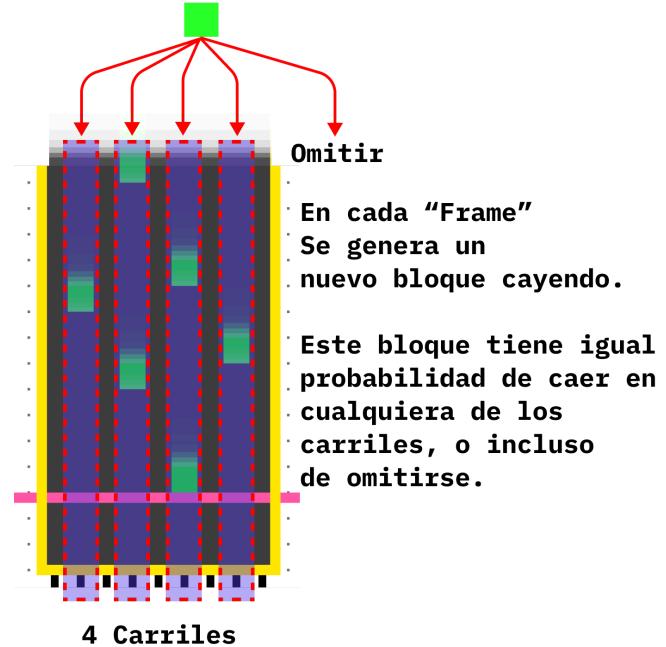




Por “aleatorio” se entiende lo siguiente.

Aleatorio

Los bloques que caen y se desplazan por la pantalla lo hacen sin ningún tipo de orden en particular sobre los cuatro carriles. Un bloque puede caer en el carril 1,2,3,4 o incluso omitir esa fila (un silencio). Se sobre entiende que no pueden caer dos bloques en la misma fila; esta limitación se incluye ya que quien esté jugando solo puede presionar un botón a la vez dentro del simulador.



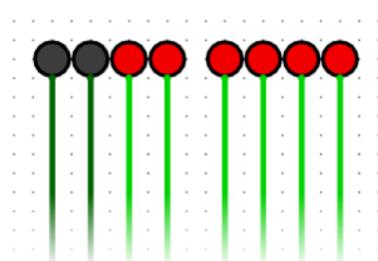
Juego Aleatorio

La matriz LED se encargará de mostrar los bloques cayendo de arriba a abajo.

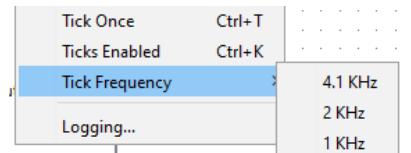
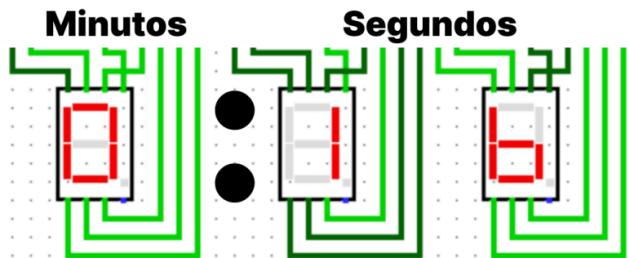
Los pulsadores serán utilizados para registrar cuando un jugador los presiona. Se tomará como “punto” cuando un bloque cruce la línea horizontal mientras que al mismo tiempo se esté pulsando el botón que corresponda a la columna. Como se muestra en la foto. Los estudiantes quedan libres de hacer modificaciones sobre la ubicación de la fila horizontal que será la línea de detección que se utiliza para contabilizar y detectar la caída de bloques más su pulsación de botón, como también así el valor del puntaje.

El sistema registra el puntaje y lo muestra en binario mediante LEDs rojos. No hay una condición de pérdida de puntos, solo se suma si las condiciones se cumplen. El puntaje se debe mantener hasta comenzar un nuevo juego.

63 Puntos!



En este modo de juego, se deben acumular cuantos más puntos se pueda en un límite de tiempo de 2 minutos.



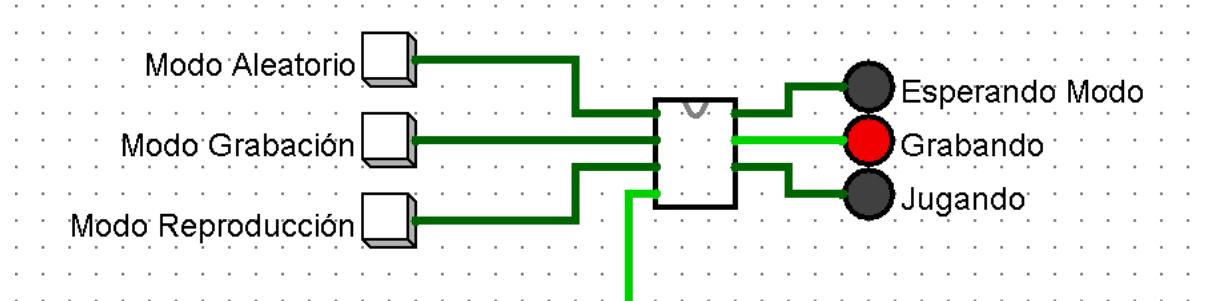
⚠ La precisión del temporizador al tiempo real puede tolerar ligeros desfases, siempre que el límite de 2 minutos se mantenga aproximado.

Menú (Máquina de estados)

El órgano que nos pide Babbage, debe incluir algún tipo de menú para manejar cuando se está realizando cada funcionalidad y poder escoger cual queremos realizar en cada momento dado. Ada Lovelace astutamente sugiere hacer uso de una máquina de estados para ello.

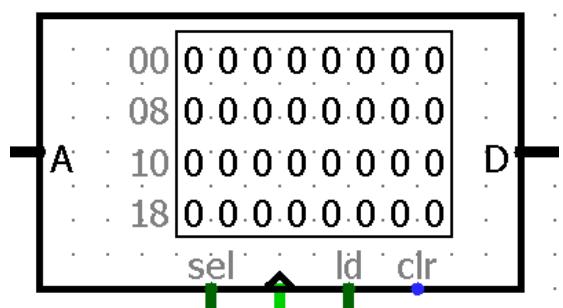
Durante la grabación, modo aleatorio o reproducción, el sistema ignora las entradas del menú. Solo volverá al menú cuando:

- En modo aleatorio o reproducción hayan transcurrido 2 minutos.
- En modo grabación se hayan completado los 32 frames.



Grabadora (RAM)

Nuestro órgano digital deberá contar con la opción de poder realizar grabaciones. Es decir, una vez que se entra en este modo, el órgano registra toda entrada realizada en un tiempo predeterminado de 32 "frames" (Siendo 1 frame = 1 botón). Se deben usar los mismos pulsadores que se utilizan para jugar. El registro de estos eventos debe hacerse en una memoria RAM.



Queda a disposición del estudiante registrar en memoria estos eventos como cada grupo escoja hacerlo, su representación debe estar especificada en la documentación. Mientras se esté en el modo de grabación, se debe mostrar en el reloj, el tiempo restante de grabación.

Juego Modo Reproducción

En términos de juego, es igual que el modo aleatorio, pero se utilizara como pista de bloques lo que se haya grabado en el modo grabación. Se debe mantener el límite de dos minutos, haciendo loop sobre lo grabado hasta completar el tiempo.

Documentación

Además del archivo .circ de logisim (Especificar en el caso de ser Evolution), se debe adjuntar una documentación del sistema. Este en parte sirve para explicar el proceso de creación del mismo, detallando los problemas que se encontraron, que opciones tuvieron en cuenta, como lo solucionaron. Como así también todo cálculo realizado que acompañe el trabajo, como máquinas de estado u otros.

Se pide que cada chip en el sistema cuenta con etiquetas en sus entradas y salidas, y que cada chip del circuito se detalle dentro de la documentación.

Alternativa

Para aquellos que estén interesados en hacer una implementación física de un tema relacionado. Pueden plantear al docente un proyecto, o pueden solicitar del docente un desafío a realizar. Si el docente está de acuerdo que el proyecto es adecuado en complejidad, interés y relevancia para la materia, entonces lo puede aprobar y el equipo puede hacer un proyecto basado en tecnologías como Arduino, Raspberry pi, con uso de motores, sensores, etc. La evaluación consistirá en una prueba presencial del proyecto con una demostración de su funcionamiento, y un documento con la información del proyecto, según las pautas del docente, entregado por gestión en la fecha estipulada.