



Manual de Uso

Julio de 2025

Proyecto TED2021-131172B-I00: "TUTOR DE
PROGRAMACION INTELIGENTE MEDIANTE ROBOTICA
SIMULADA"



ÍNDICE

Introducción a ProgTutor	3
Diseño de ProgTutor y configuración inicial	3
Opciones de ProgTutor-Sim.....	5
Opciones de ProgTutor-Plugin.....	10
Fases de ejecución de ProgTutor por un alumno	12
Panel del profesor	22
Metodología de enseñanza	32

Introducción a ProgTutor

ProgTutor es una herramienta de enseñanza personalizada diseñada para el aprendizaje de los fundamentos de la programación en Python, de una forma práctica y aplicada, mediante retos que deben ser resueltos con un robot simulado. Desarrollado por la Universidade da Coruña, el sistema integra un tutor inteligente que adapta los ejercicios al ritmo de progreso de cada estudiante con un interfaz que proporciona apoyo mediante pistas y correcciones automáticas de los programas. El enfoque pedagógico combina metodologías activas, tecnología educativa y aprendizaje autónomo, con un énfasis especial en el papel central del profesorado.

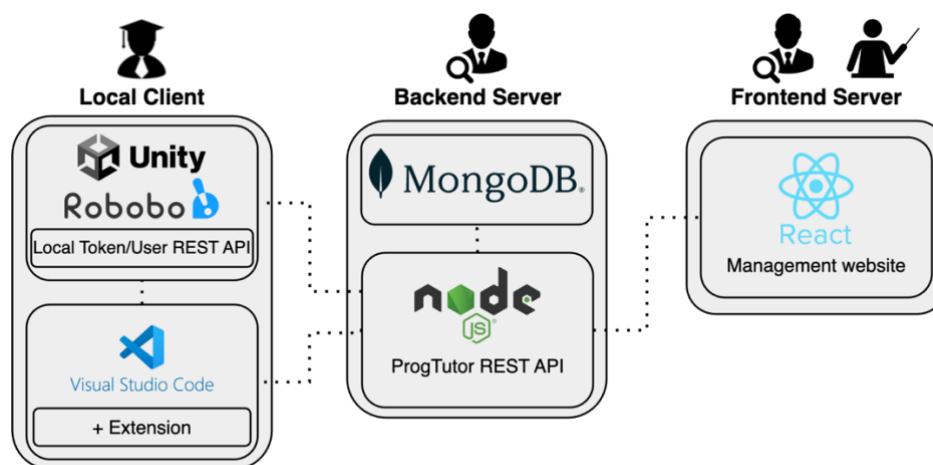
ProgTutor es totalmente funcional y está disponible para su descarga y uso en la web del proyecto: https://www.udc.es/es/lidiagroup/proyectos_id/progtutor. Se compone de 2 elementos principales:

1. Una extensión de Visual Studio Code, **ProgTutor-Plugin**, que gestiona la parte de creación del programa en Python, aportando realimentación sencilla sobre los errores, y controlando la comunicación con el segundo elemento
2. Un simulador 3D llamado **ProgTutor-Sim**, que permite plantear y resolver retos realistas de robótica con el robot educativo Robobo.

Ambos elementos están disponibles en la web para que cualquier docente interesado en impartir formación en Python de forma aplicada y personalizada lo pueda utilizar. Para ello, se ha desarrollado un **manual de instalación** y un **manual de uso (este documento)** que pueden ser utilizados de manera autónoma. Asimismo, se ha implementado un panel web (**dashboard**) que permite monitorizar el avance de los estudiantes, identificando aquellos que necesitan más apoyo y qué errores son más comunes, reforzando así su capacidad para guiar el aprendizaje.

Diseño de ProgTutor y configuración inicial

La implementación de ProgTutor se basa en la arquitectura cliente-servidor que se muestra en la figura siguiente:



Por una parte, el cliente (ordenador del estudiante) debe tener instalados los siguientes elementos (ver **manual de instalación**):

1. Python: debe estar instalado en el sistema para poder desarrollar y probar los programas
2. Visual Studio Code: necesario para el desarrollo del código
3. ProgTutor-Plugin: necesario para interactuar con el simulador y la base de datos
4. ProgTutor-Sim: necesario para la ejecución de los programas
5. Librería robobo.py: necesaria para programar el robot simulado
6. Extensión de JupyterLab para Visual Studio Code: necesaria para que los estudiantes realicen los cuadernos de forma individual

Por otro lado, ProgTutor se apoya en un servidor, en el cual se ejecuta una base de datos no SQL con MongoDB para almacenar toda la información relevante del sistema. Tanto el simulador como la extensión VSC interactúan con los datos de MongoDB a través de una interfaz segura.

Finalmente, para crear los ejercicios utilizados durante el curso y almacenarlos en la base de datos, se desarrolló un formulario web con una interfaz gráfica intuitiva en React, facilitando la navegación de todos los usuarios.

A continuación, se explica el funcionamiento básico de *ProgTutor-Plugin*, *ProgTutor-Sim* y el *panel del profesor*. Es imprescindible haber instalado previamente todos los elementos anteriores siguiendo las indicaciones del **Manual de Instalación**.

En caso de querer utilizar la herramienta en el aula, se debe contactar con francisco.bellas@udc.es para gestionar la creación de los usuarios y la configuración inicial del curso.

Opciones de ProgTutor-Sim

Al abrir el simulador ProgTutor-Sim se muestra la pantalla de login, en la que el alumno debe seleccionar el curso al que pertenece e introducir tanto su código de alumno como su contraseña asignada.



Tras acceder correctamente, se presenta el menú principal:



Se describen a continuación los distintos apartados.

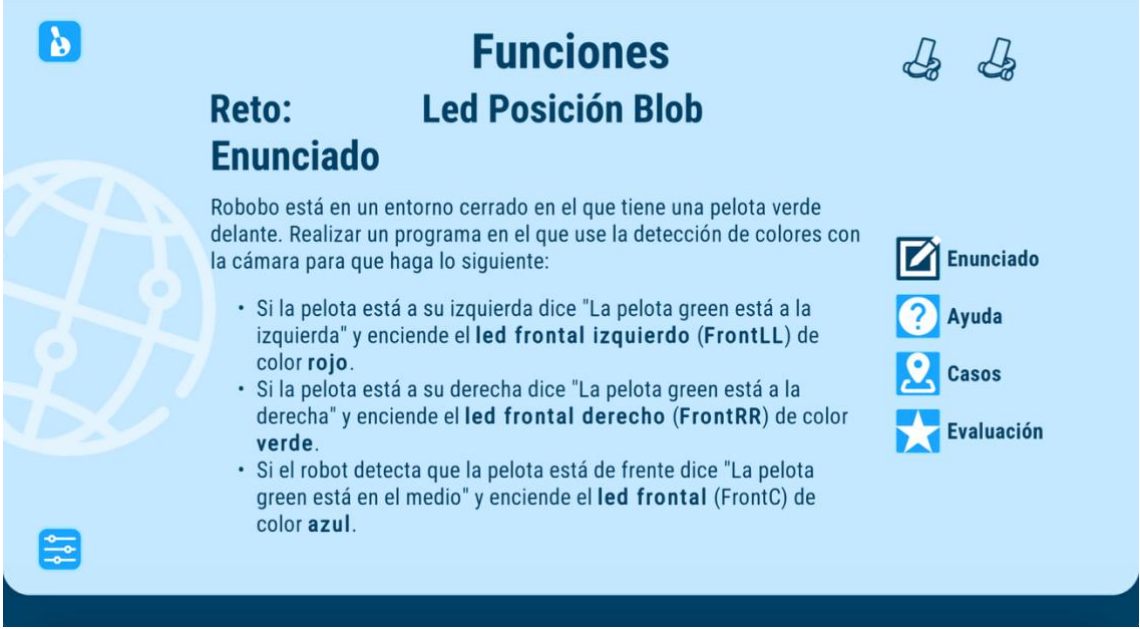
1. Ejercicios de programación

Este es el apartado principal de la herramienta ProgTutor, donde se muestran los datos de los ejercicios de robótica que el alumno debe completar. Se muestra en esta pantalla el bloque temático en el que se encuentra el alumno, así como el nombre del reto actual y su nivel de dificultad, indicado con los iconos del robot Robobo.

El resto de la información sobre el ejercicio actual se divide en las siguientes pestañas.

1.1 Enunciado

Muestra el enunciado del ejercicio, con los pasos que debe seguir el robot para completar el reto. El alumno debe programar al robot para que siga estos pasos correctamente.



The screenshot shows the 'Enunciado' tab of the ProgTutor interface. The title is 'Funciones Led Posición Blob'. The sub-title is 'Reto: Enunciado'. The main text describes the challenge: 'Robobo está en un entorno cerrado en el que tiene una pelota verde delante. Realizar un programa en el que use la detección de colores con la cámara para que haga lo siguiente:'. Below this, there are three bullet points: '• Si la pelota está a su izquierda dice "La pelota green está a la izquierda" y enciende el **led frontal izquierdo (FrontLL)** de color **rojo**.', '• Si la pelota está a su derecha dice "La pelota green está a la derecha" y enciende el **led frontal derecho (FrontRR)** de color **verde**.', and '• Si el robot detecta que la pelota está de frente dice "La pelota green está en el medio" y enciende el **led frontal (FrontC)** de color **azul**.'. On the right side, there is a vertical menu with four items: 'Enunciado' (selected), 'Ayuda', 'Casos', and 'Evaluación'. Each item has a corresponding icon: a notepad for 'Enunciado', a question mark for 'Ayuda', a location pin for 'Casos', and a star for 'Evaluación'. The background features a stylized globe icon on the left and a small robot icon in the top right corner.

1.2 Ayuda

Muestra los métodos del robot Robobo que se utilizarán en este ejercicio, con hiperenlaces a la documentación de cada método.



The screenshot shows the 'Ayuda' tab of the ProgTutor interface. The title is 'Funciones Led Posición Blob'. The sub-title is 'Reto: Ayuda'. The main text states: 'En este reto se utilizan los siguientes métodos de Robobo:'. Below this, there is a list of methods: '- readColorBlob', '- sayText', '- setLedColorTo', '- moveTiltTo', and '- movePanTo'. On the right side, there is a vertical menu with four items: 'Enunciado', 'Ayuda' (selected), 'Casos', and 'Evaluación'. Each item has a corresponding icon: a notepad for 'Enunciado', a question mark for 'Ayuda', a location pin for 'Casos', and a star for 'Evaluación'. The background features a stylized globe icon on the left and a small robot icon in the top right corner.

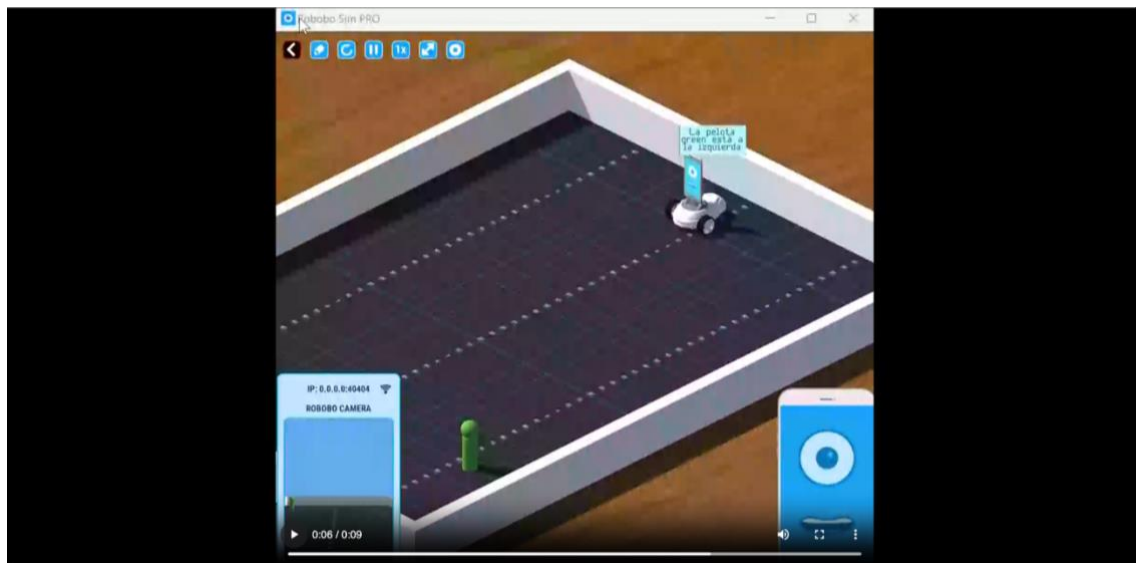
1.3 Casos

Cada ejercicio cuenta con tres casos de uso distintos, y el alumno debe resolver el reto de forma genérica para que cumpla con los requisitos de los tres casos. Cada caso de uso puede requerir una salida por pantalla distinta, o ejecutarse en un mundo distinto con una disposición de objetos distinta. Por ejemplo, si el ejercicio pide indicar dónde se encuentra una pelota, cada caso de uso puede tener la pelota colocada en un sitio distinto (izquierda, medio, derecha).

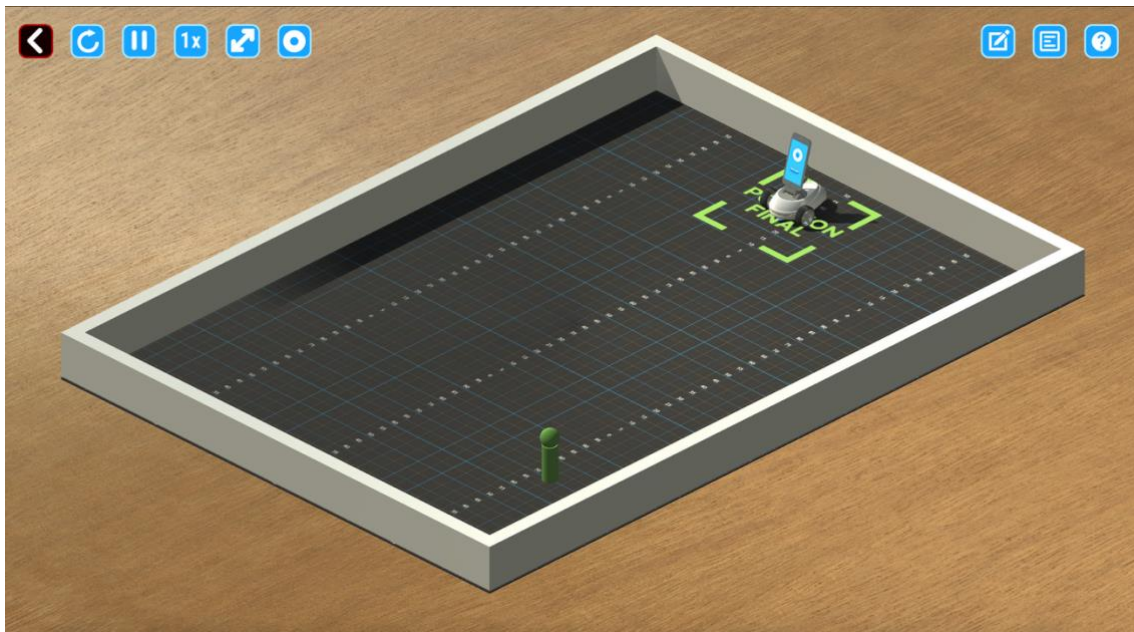


Cada caso de uso cuenta con un enlace a un vídeo de ejemplo, en el que se muestra el resultado esperado para ese caso.

Al hacer clic en el enlace, se abrirá el vídeo en el navegador.



Cada caso de uso cuenta con un botón “Prueba este caso”, que nos lleva al escenario en el que el alumno debe programar y probar su solución.



1.4 Evaluación

Una vez el alumno prueba el ejercicio en los tres casos de usos y envía su código a evaluar, se muestran en esta pantalla los resultados, indicando para cada caso de uso si lo ha resuelto correctamente o si cometió algún error. Se muestran notas de ayuda para cada tipo de error posible.



Funciones

Led Posición Blob




Reto:

Evaluación

Caso de Ejemplo	Notas
✓ Caso 1	
✓ Caso 2	
✗ Caso 3	Error en setLedColorTo: No se ha encontrado el comando (led: LED.FrontC, color: Color.BLUE).

 Enunciado

 Ayuda

 Casos

 Evaluación

2. Mundos de prueba

En este apartado se permite seleccionar escenarios de prueba, sobre los que probar las funciones de Robobo en diferentes entornos. Este modo no cuenta con ejercicios asignados.



3. Opciones

Menú de con las posibles opciones del simulador y de su robot simulado. Se recomienda comprobar la instalación de Python, e indicar una ruta personalizada en caso de que falle la ruta por defecto. Se muestran a continuación las opciones recomendadas.

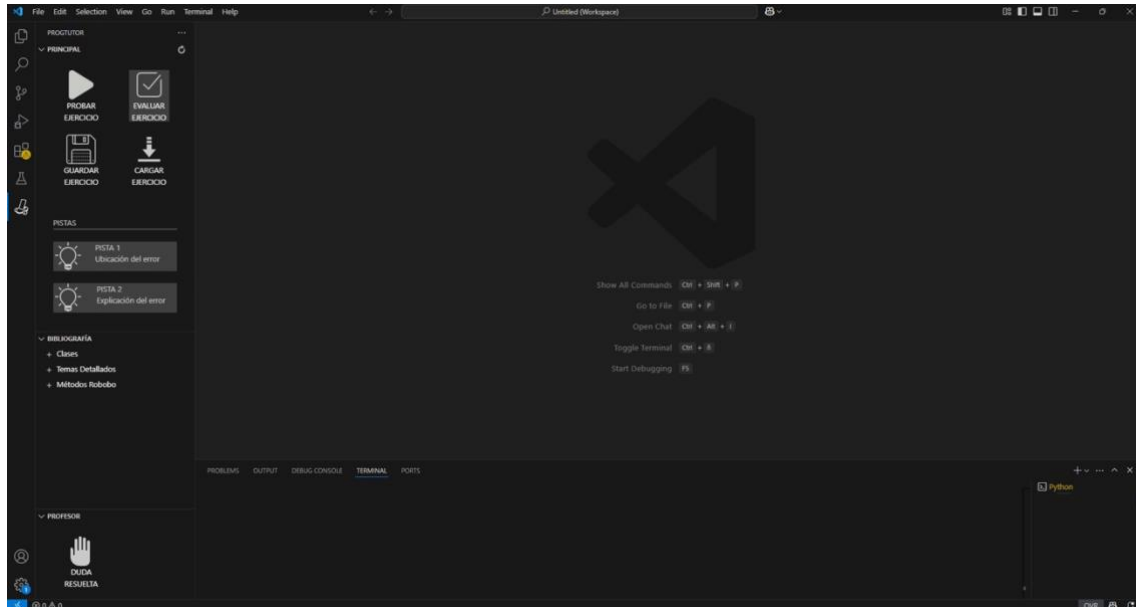


4. Salir

Permite volver a la pantalla de login.

Opciones de ProgTutor-Plugin

Para acceder al plugin, se debe hacer clic sobre el icono de Robobo ubicado en la barra lateral izquierda de Visual Studio Code. Al realizar esta acción, se desplegará el menú principal del plugin, como se muestra en la imagen correspondiente. Este menú está compuesto por tres secciones principales, las cuales se describen a continuación.



1. Principal

En esta sección se encuentran los botones con las acciones principales para el estudiante:

- **Probar Ejercicio:** al hacer clic, ejecuta el código que está en la ventana de edición dentro del simulador de ProgTutor.
- **Evaluar Ejercicio:** inicialmente está desactivado y solo se activa cuando el código ha sido ejecutado al menos una vez sin errores. Si después de esa ejecución se modifica el código, el botón se desactiva nuevamente hasta que se vuelva a ejecutar sin errores. Este botón envía el reto para su evaluación en ProgTutor y debe usarse solo después de haber probado el ejercicio con todos los casos de uso.
- **Guardar Ejercicio:** guarda en la base de datos el código que está activo en el editor en ese momento.
- **Cargar Ejercicio:** carga en el editor el último código guardado en la base de datos para ese ejercicio.

2. Pistas

Para cada ejercicio existen dos pistas disponibles. Inicialmente, estas pistas están desactivadas, ya que se entregan automáticamente a los estudiantes cuando cometen errores durante la resolución. Sin embargo, a medida que los estudiantes avanzan en el curso, las pistas dejan de mostrarse automáticamente y solo estarán disponibles bajo demanda, a través de estos botones.

3. Bibliografía

En esta sección se encuentra la bibliografía recomendada para todo el curso, dividida en tres grupos principales:

- **Clases:** incluye todos los cuadernos de Jupyter utilizados durante el curso para las sesiones teóricas.
- **Temas Detallados:** contiene enlaces a un libro de Python donde se explica con detalle y ejemplos cada uno de los temas tratados en el curso.
- **Métodos de Robobo:** compuesta por cuadernos de Jupyter que ofrecen explicaciones detalladas y ejemplos de todos los métodos disponibles en la librería de Robobo.

4. Profesor

Cuenta con el botón de ayuda del profesor, que debe ser utilizado exclusivamente por el profesor. Este botón se presiona cada vez que el profesor asiste al estudiante en un reto. Al hacer clic, se registra en la base de datos que el profesor brindó ayuda al estudiante en ese reto específico. Es importante que el profesor presione el botón cada vez que ofrezca asistencia, incluso si es más de una vez dentro del mismo reto.

Fases de ejecución de ProgTutor por un alumno

A continuación, se detallan los pasos a seguir por el alumno para utilizar ProgTutor.

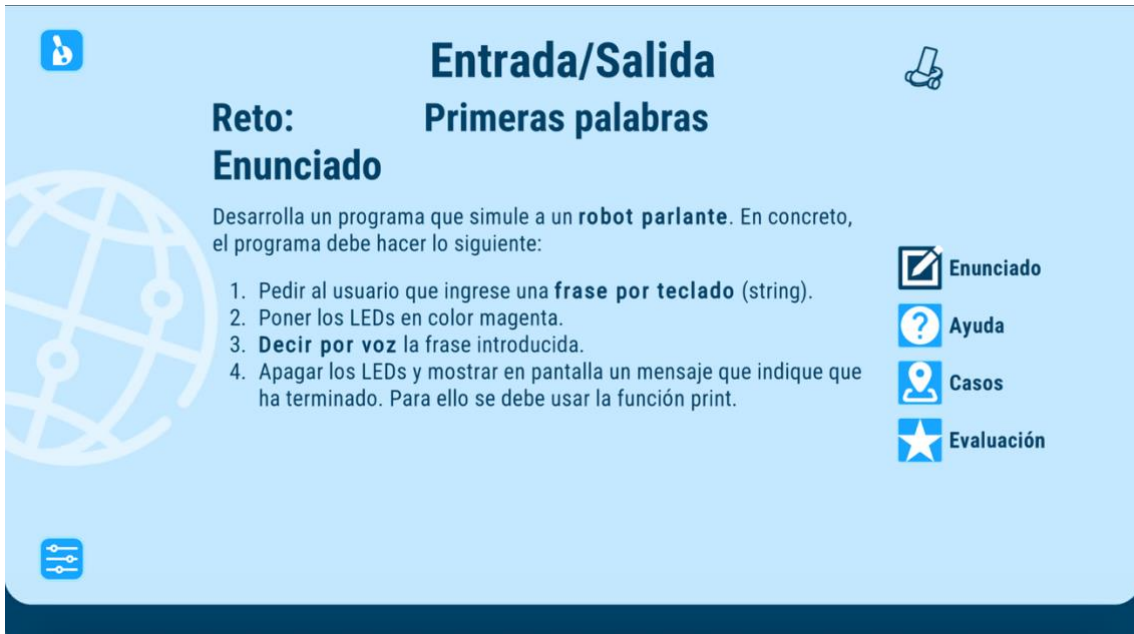
1. Abrir ProgTutor-Sim, seleccionar el curso al que pertenece el alumno, y autenticarse en la pantalla de login.



2. Entrar en el apartado “Ejercicios de Programación”.



3. Leer el enunciado del ejercicio asignado.



Entrada/Salida
Primeras palabras

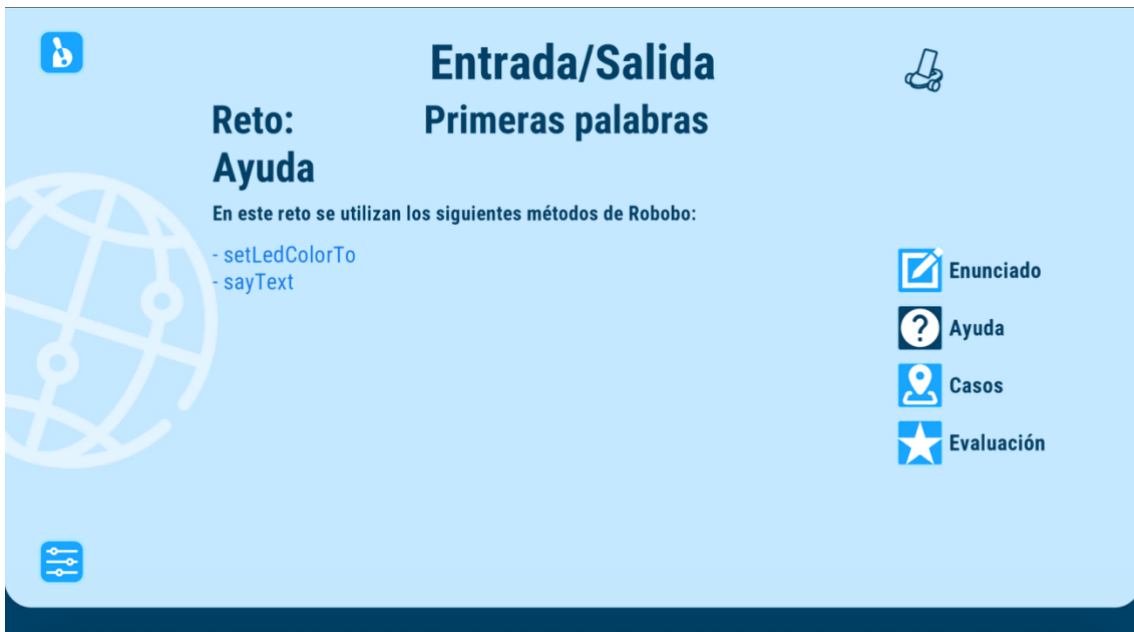
Reto:
Enunciado

Desarrolla un programa que simule a un **robot parlante**. En concreto, el programa debe hacer lo siguiente:

1. Pedir al usuario que ingrese una **frase por teclado** (string).
2. Poner los LEDs en color magenta.
3. **Decir por voz** la frase introducida.
4. Apagar los LEDs y mostrar en pantalla un mensaje que indique que ha terminado. Para ello se debe usar la función print.

Enunciado
Ayuda
Casos
Evaluación

4. Leer la ayuda del ejercicio.



Entrada/Salida
Primeras palabras

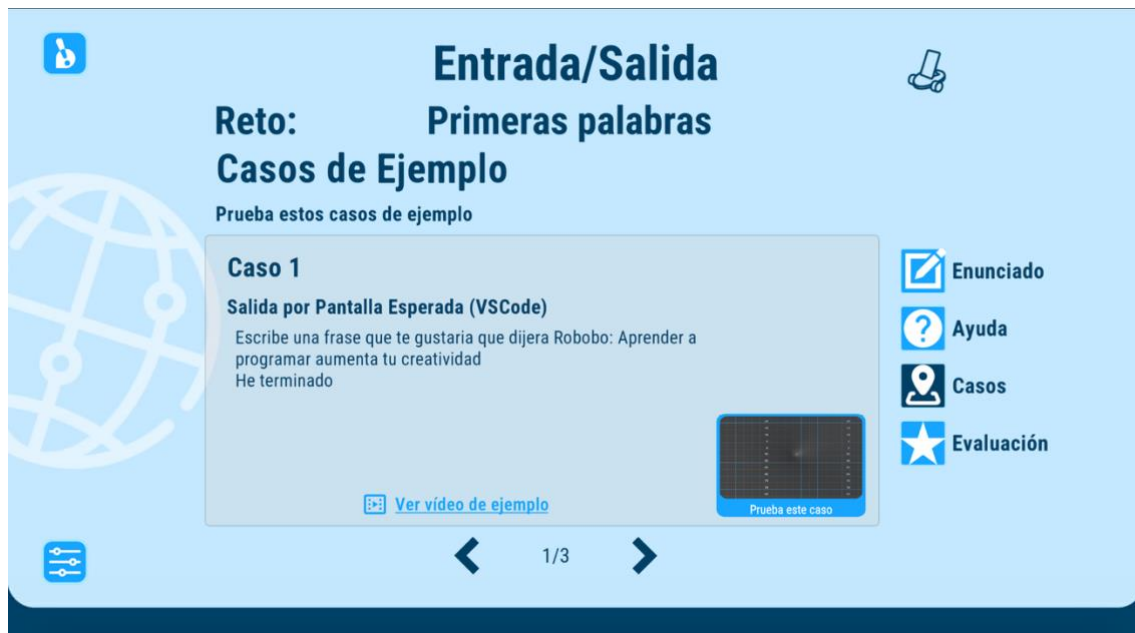
Reto:
Ayuda

En este reto se utilizan los siguientes métodos de Robobo:

- setLedColorTo
- sayText

Enunciado
Ayuda
Casos
Evaluación

5. Leer los casos de uso. Revisar las diferencias entre ellos y ver los vídeos de ejemplo.



Entrada/Salida

Reto: Primeras palabras

Casos de Ejemplo

Prueba estos casos de ejemplo

Caso 1

Salida por Pantalla Esperada (VSCode)

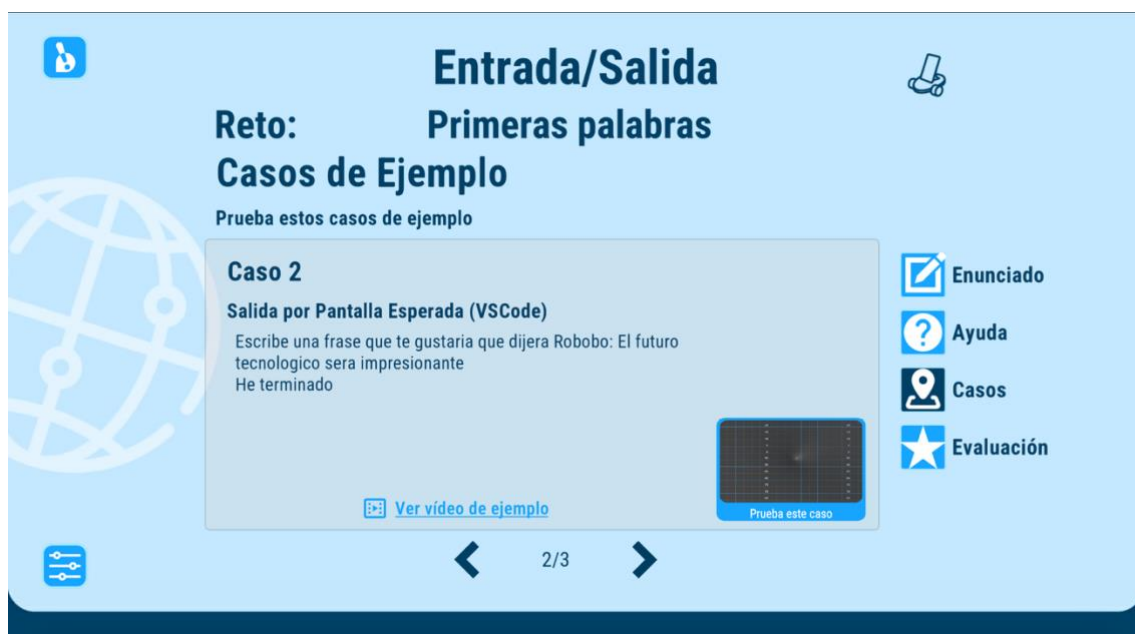
Escribe una frase que te gustaría que dijera Robobo: Aprender a programar aumenta tu creatividad
He terminado

[Ver vídeo de ejemplo](#)

Prueba este caso

Enunciado
Ayuda
Casos
Evaluación

1/3



Entrada/Salida

Reto: Primeras palabras

Casos de Ejemplo

Prueba estos casos de ejemplo

Caso 2

Salida por Pantalla Esperada (VSCode)

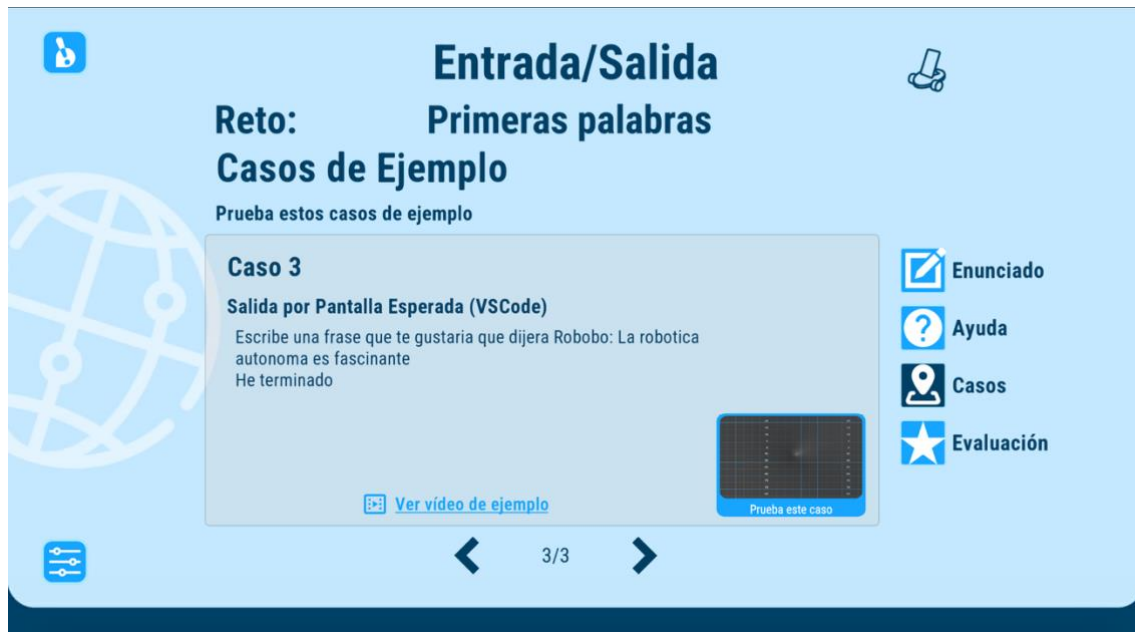
Escribe una frase que te gustaría que dijera Robobo: El futuro tecnologico sera impresionante
He terminado

[Ver vídeo de ejemplo](#)

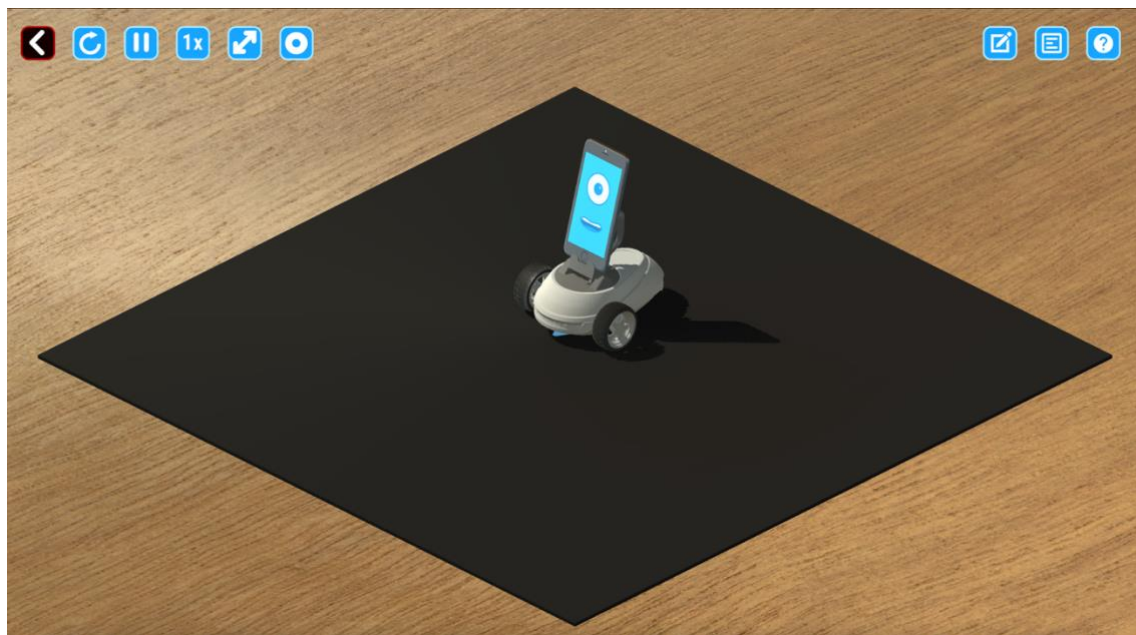
Prueba este caso

Enunciado
Ayuda
Casos
Evaluación

2/3

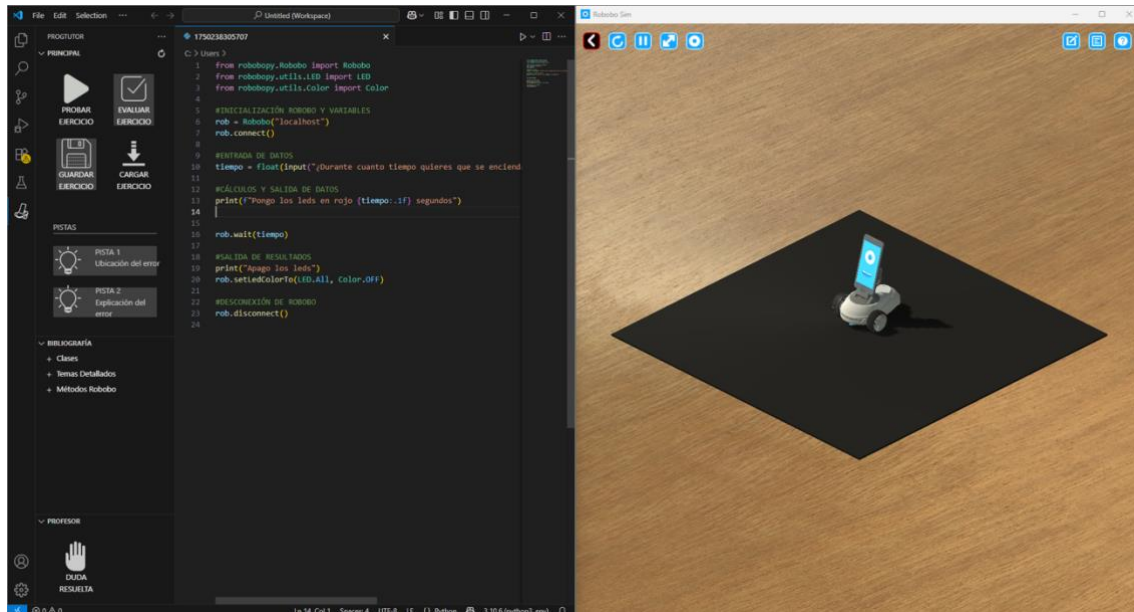


6. Acceder al escenario del primer caso de uso, utilizando el botón “Prueba este caso”.



7. Abrir Visual Studio Code, y hacer clic sobre el icono de Robobo ubicado en la barra lateral izquierda para abrir ProgTutor-Plugin.

Para facilitar el trabajo con la herramienta, se recomienda utilizar ambas partes (el plugin y el simulador) en pantalla dividida, tal como se muestra en la siguiente figura. Esto permite visualizar en el simulador, en tiempo real, la ejecución del código que se desarrolla desde Visual Studio Code.

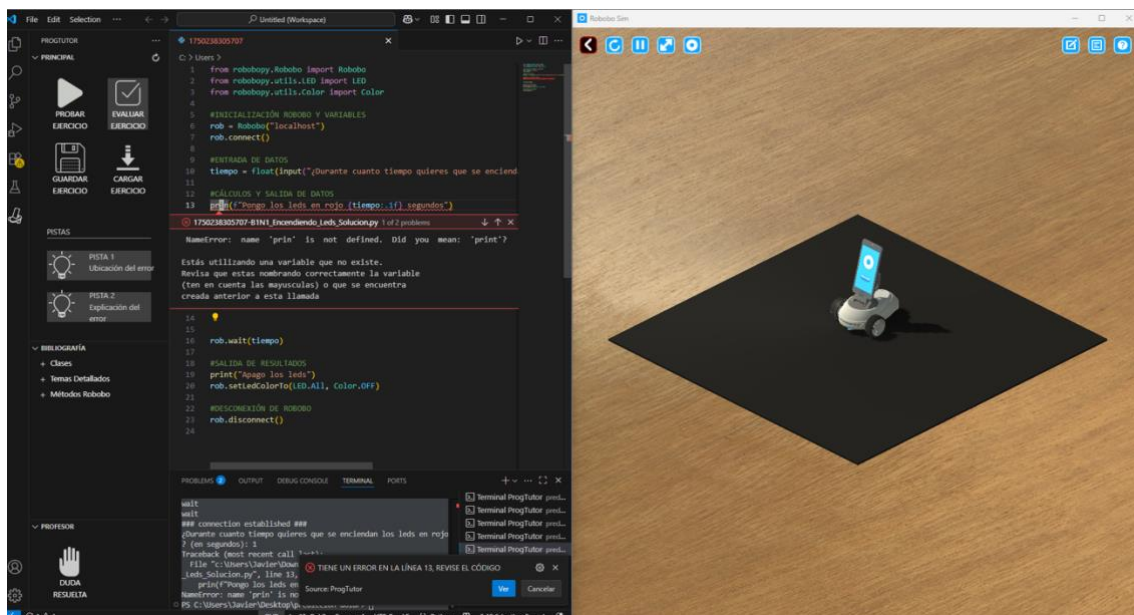


8. Crear un nuevo fichero de Python (con la extensión .py) y empezar a programar la solución al ejercicio.

Para programar los distintos retos, es fundamental tener siempre abierto el simulador en la ventana correspondiente al caso de uso.

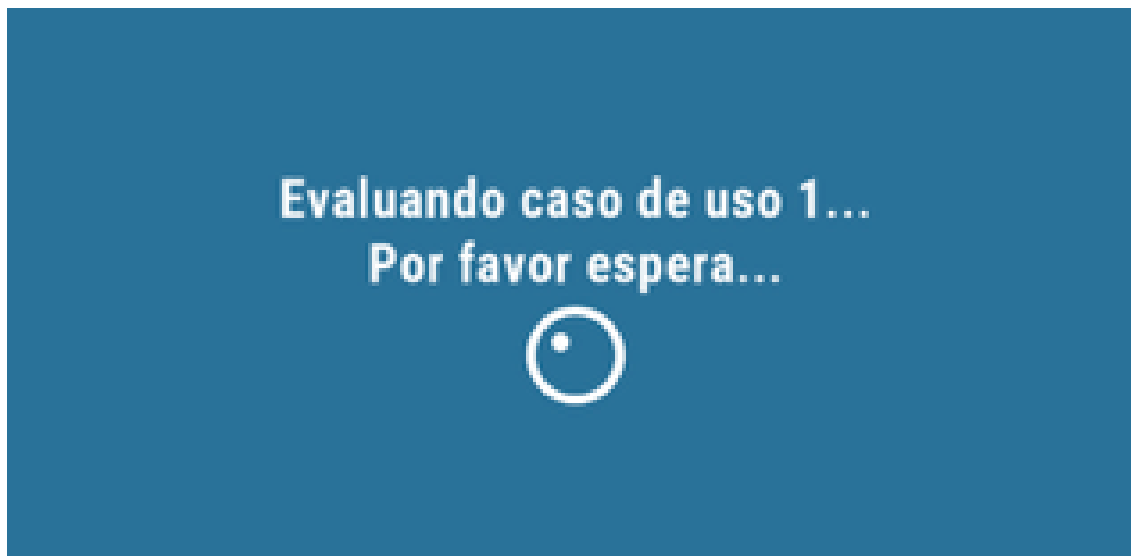
9. Clicar en el botón “Probar Ejercicio” una vez se ha programado la solución, para ejecutar el código en el simulador. Estas ejecuciones funcionan como pruebas preliminares, útiles para verificar que el código se comporta correctamente.

Si el código contiene un error de Python, se mostrará una ventana con información sobre el tipo de error y la línea exacta donde ocurre, como se ilustra en la figura. Es importante tener en cuenta que, durante los primeros temas, estas pistas de error se mostrarán automáticamente. Sin embargo, a medida que se avance en los retos, dejarán de aparecer de forma automática y estarán disponibles solo cuando se soliciten.



10. Entrar en el escenario del siguiente caso de uso cuando el programa funcione para el primer caso de uso. Repetir hasta haber probado el programa exitosamente en todos los casos.
11. Clicar el botón “Evaluar Ejercicio” cuando hayamos probado el programa exitosamente en todos los casos de uso.

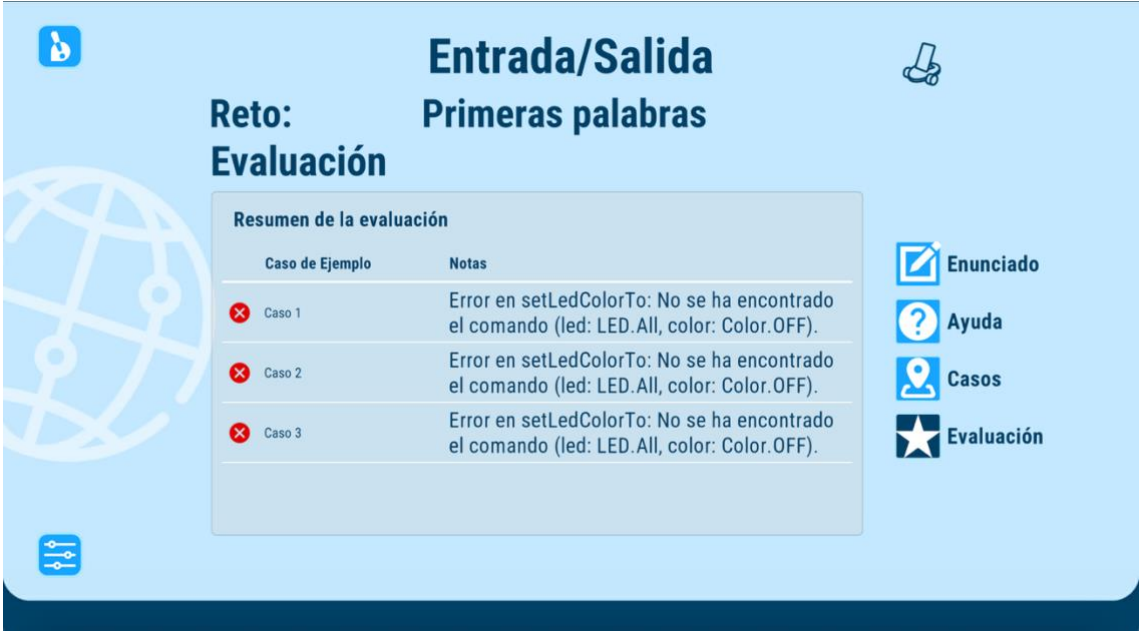
El botón se habilitará automáticamente cuando no tengamos errores en el código. Recuerde que la evaluación constituye el paso final en la resolución del ejercicio, por lo tanto, solo debe llevarse a cabo cuando se tenga plena certeza de que la solución es correcta.



Si nuestro código tiene algún error, se nos mostrará un aviso indicando que la evaluación ha fallado.



Cuando falla la evaluación, se nos indica qué casos de uso han fallado, y se nos muestran notas de ayuda sobre el comando que ha fallado en cada caso.



The screenshot shows the 'Entrada/Salida' (Input/Output) section of the ProgTutor interface. The main heading is 'Reto: Evaluación' (Challenge: Evaluation). Below it, a table titled 'Resumen de la evaluación' (Evaluation Summary) lists three cases, all of which failed. The error message for each case is 'Error en setLedColorTo: No se ha encontrado el comando (led: LED.All, color: Color.OFF)'. On the right side, there are four icons with labels: 'Enunciado' (Statement), 'Ayuda' (Help), 'Casos' (Cases), and 'Evaluación' (Evaluation).

Caso de Ejemplo	Notas
✗ Caso 1	Error en setLedColorTo: No se ha encontrado el comando (led: LED.All, color: Color.OFF).
✗ Caso 2	Error en setLedColorTo: No se ha encontrado el comando (led: LED.All, color: Color.OFF).
✗ Caso 3	Error en setLedColorTo: No se ha encontrado el comando (led: LED.All, color: Color.OFF).

Si nuestro código es correcto, se nos mostrará un aviso indicando que la evaluación ha terminado correctamente.



The screenshot shows the 'Resultado de la Evaluación' (Evaluation Result) dialog box. It contains the text 'Todos los casos de uso fueron evaluados correctamente.' (All use cases were evaluated correctly.) and an 'Aceptar' (Accept) button.

Resultado de la Evaluación

Todos los casos de uso fueron evaluados correctamente.

Aceptar

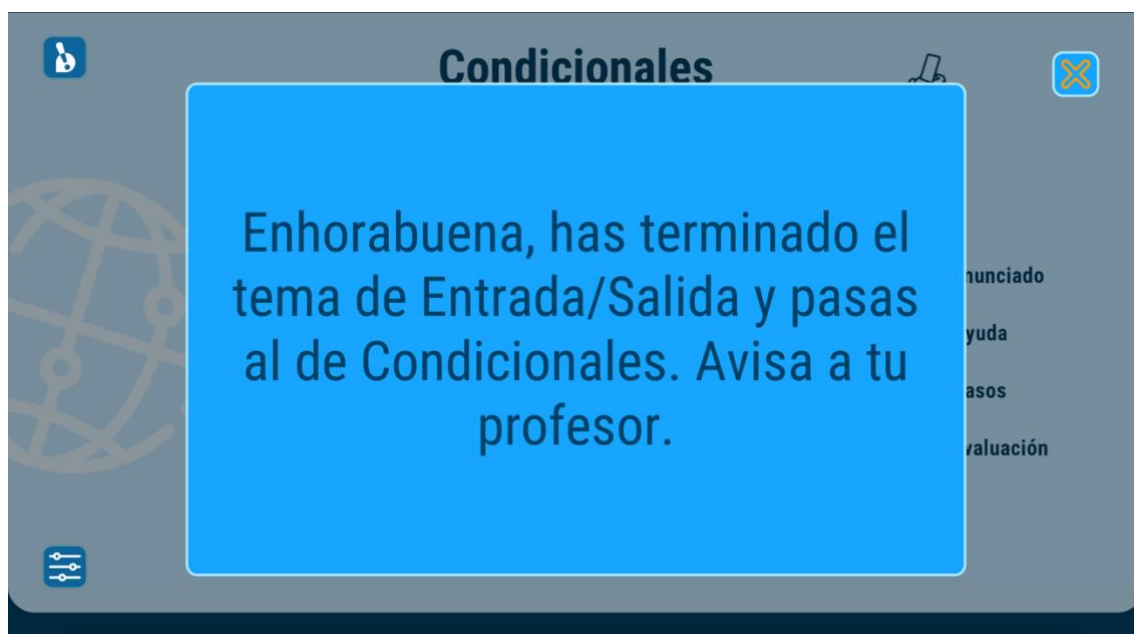


Una vez hayamos logrado la evaluación correcta del ejercicio, se nos asignará un nuevo ejercicio automáticamente, para el que debemos repetir los mismos pasos.

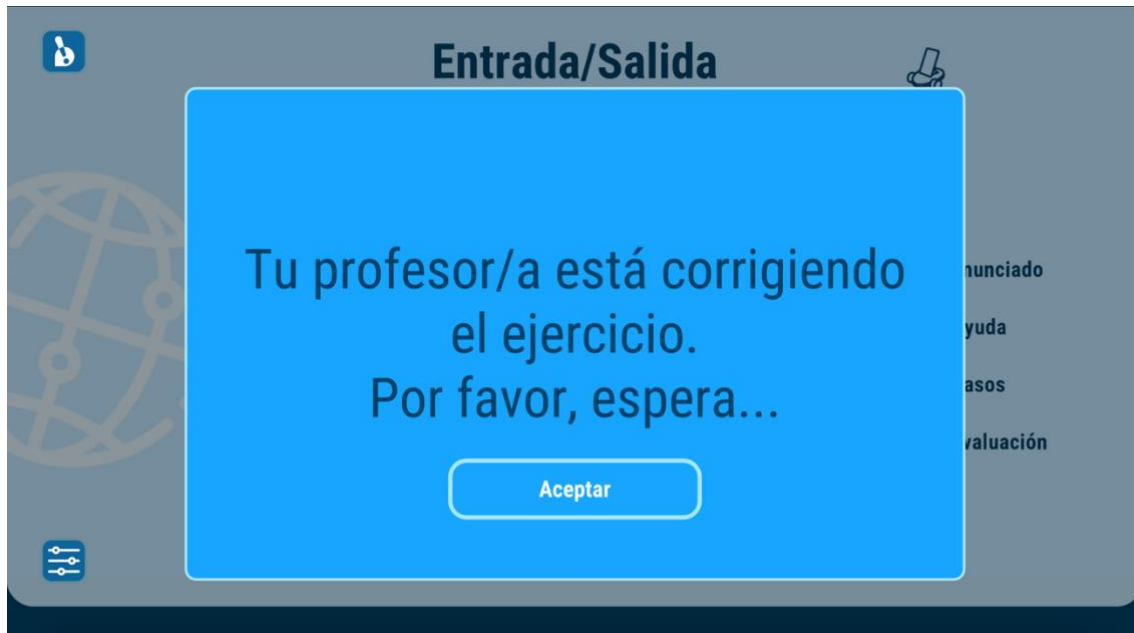
Paneles de aviso

Tras la evaluación del ejercicio, pueden aparecer los siguientes paneles de aviso.

1. **Siguiente tema:** aparece al terminar todos los ejercicios asignados de un mismo tema. Es recomendable avisar al/la profesor/a para que explique la teoría del siguiente tema.



2. **Profesor/a corrigiendo:** si el código del alumno pasa la evaluación, pero dista mucho del código de la solución propuesta, se avisará al profesor para que valore la nota del ejercicio.



3. **Repetir nivel:** aparece al detectar que se debe realizar otro ejercicio del mismo nivel de dificultad. Se consideran varios criterios que determinan esta decisión, y se muestra en la gráfica cuánto ha influido cada criterio en la probabilidad de repetir nivel:
- **Nota:** si el código del alumno pasa la evaluación, pero dista mucho del código de la solución propuesta, su nota será menor y aumentará la probabilidad de repetir nivel de dificultad.
 - **Tiempo usado:** si el alumno tarda demasiado en resolver el ejercicio, aumentará la probabilidad de repetir nivel de dificultad.
 - **Ayudas:** si el profesor ha ayudado mucho al alumno, aumentará la probabilidad de repetir nivel de dificultad.
 - **Evaluaciones fallidas:** si el alumno ha evaluado incorrectamente muchas veces el ejercicio, aumentará la probabilidad de repetir nivel de dificultad.

Debido a los datos de tu evaluación, debes hacer un ejercicio del mismo nivel:

Nota:	25%
Tiempo usado:	0%
Ayudas:	0%
Evaluaciones fallidas:	0%

?

Debido a los datos de tu evaluación, debes hacer un ejercicio del mismo nivel:

Nota: nota del código del último ejercicio.

Tiempo usado: tiempo usado en el último ejercicio.

Ayudas: número de dudas resueltas por el/la profesor/a.

Evaluaciones fallidas: número de evaluaciones fallidas en el último ejercicio.

?

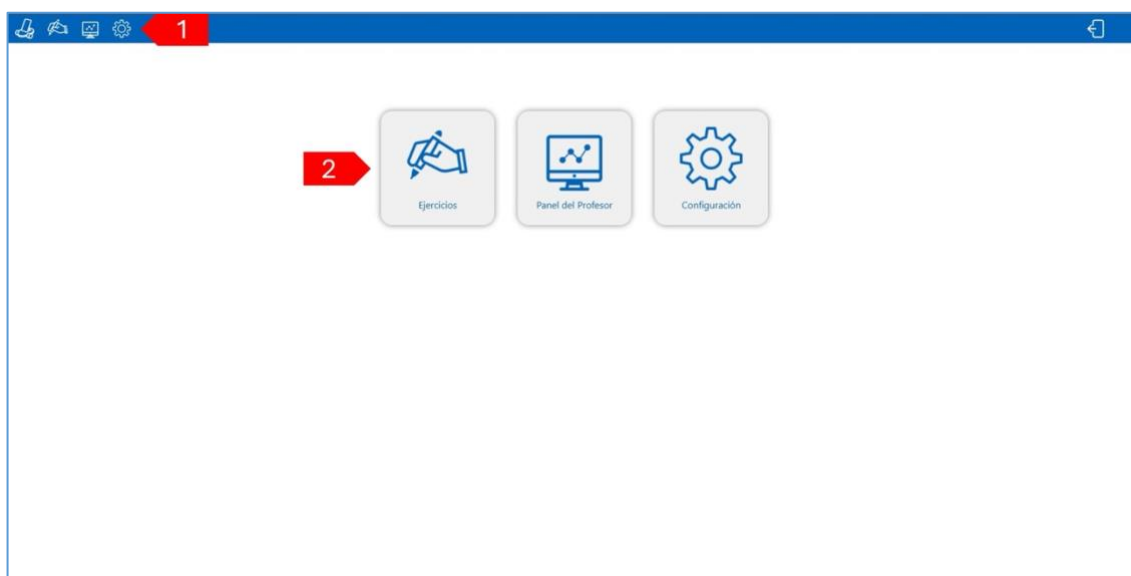
Panel del profesor

Acceso al panel del profesor

El acceso al panel del profesor se realiza a través del enlace [ProgTutor Web](#). Al ingresar, se mostrará una pantalla de inicio de sesión, donde deberá introducir su nombre de usuario y contraseña asignados previamente por el equipo del **Proyecto ProgTutor**.

Menú Principal

A continuación, se describe la pantalla principal a la que se accede al iniciar sesión en la plataforma, junto con las funcionalidades disponibles numeradas en la siguiente imagen.



1. Barra Superior

Contiene un conjunto de botones que permanecen visibles y accesibles desde cualquier sección del sitio web. Los botones son los mismos que aparecen en el punto 2, solo que estos se mantienen visibles para permitir la navegación entre páginas

2. Botonera central

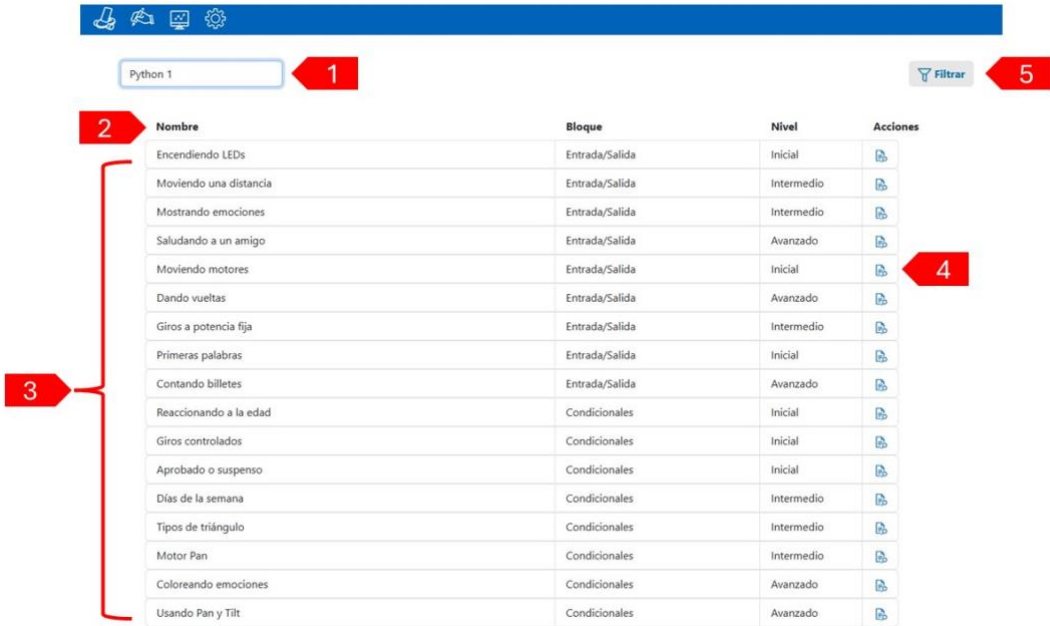
Presenta botones que permiten navegar hacia las distintas funcionalidades y ventanas del sistema. Compuesta por tres botones principales, cada uno con funciones específicas:

- **Botón "Ejercicios"**: permite acceder al listado completo de ejercicios disponibles. En esta sección se pueden visualizar todos los datos relevantes asociados a cada ejercicio.
- **Botón "Panel del Profesor"**: dirige al panel general del profesor, donde se muestra información detallada y actualizada sobre los alumnos presentes en clase.

- **Botón "Configuración"**: da acceso al área de configuración del sistema. Desde aquí es posible gestionar los retos asignados a cada alumno, así como agregar o eliminar retos de manera individual.

Navegación desde el Botón "Ejercicios"

A continuación, se describe la pantalla principal a la que se accede mediante el botón **"Ejercicios"**, junto con las funcionalidades disponibles numeradas en la imagen correspondiente.



Nombre	Bloque	Nivel	Acciones
Encendiendo LEDs	Entrada/Salida	Inicial	
Moviendo una distancia	Entrada/Salida	Intermedio	
Mostrando emociones	Entrada/Salida	Intermedio	
Saludando a un amigo	Entrada/Salida	Avanzado	
Moviendo motores	Entrada/Salida	Inicial	
Dando vueltas	Entrada/Salida	Avanzado	
Giros a potencia fija	Entrada/Salida	Intermedio	
Primeras palabras	Entrada/Salida	Inicial	
Contando billetes	Entrada/Salida	Avanzado	
Reaccionando a la edad	Condicionales	Inicial	
Giros controlados	Condicionales	Inicial	
Aprobado o suspenso	Condicionales	Inicial	
Días de la semana	Condicionales	Intermedio	
Tipos de triángulo	Condicionales	Intermedio	
Motor Pan	Condicionales	Intermedio	
Coloreando emociones	Condicionales	Avanzado	
Usando Pan y Tilt	Condicionales	Avanzado	

1. Selector de Curso

Este campo muestra todos los cursos disponibles y permite seleccionar el curso correspondiente. Al cambiar la selección, se actualiza automáticamente el listado de ejercicios mostrado.

2. Encabezado de la Tabla

Muestra los nombres de las columnas principales: Nombre (título del ejercicio), Bloque (categoría de los ejercicios según el tema tratado), Nivel (grado de dificultad de los ejercicios categorizados en Inicial, Intermedio y Avanzado), Acciones (permite acceder a mi pantalla donde aparecen los detalles del ejercicio).

Al clicar en cada encabezado el listado de ejercicios se organiza en función de dicho encabezado

3. Listado de Ejercicios

Se presenta una tabla con todos los ejercicios disponibles. Cada fila contiene un ejercicio individual con su información correspondiente

4. Botón de Acción (Visualizar)

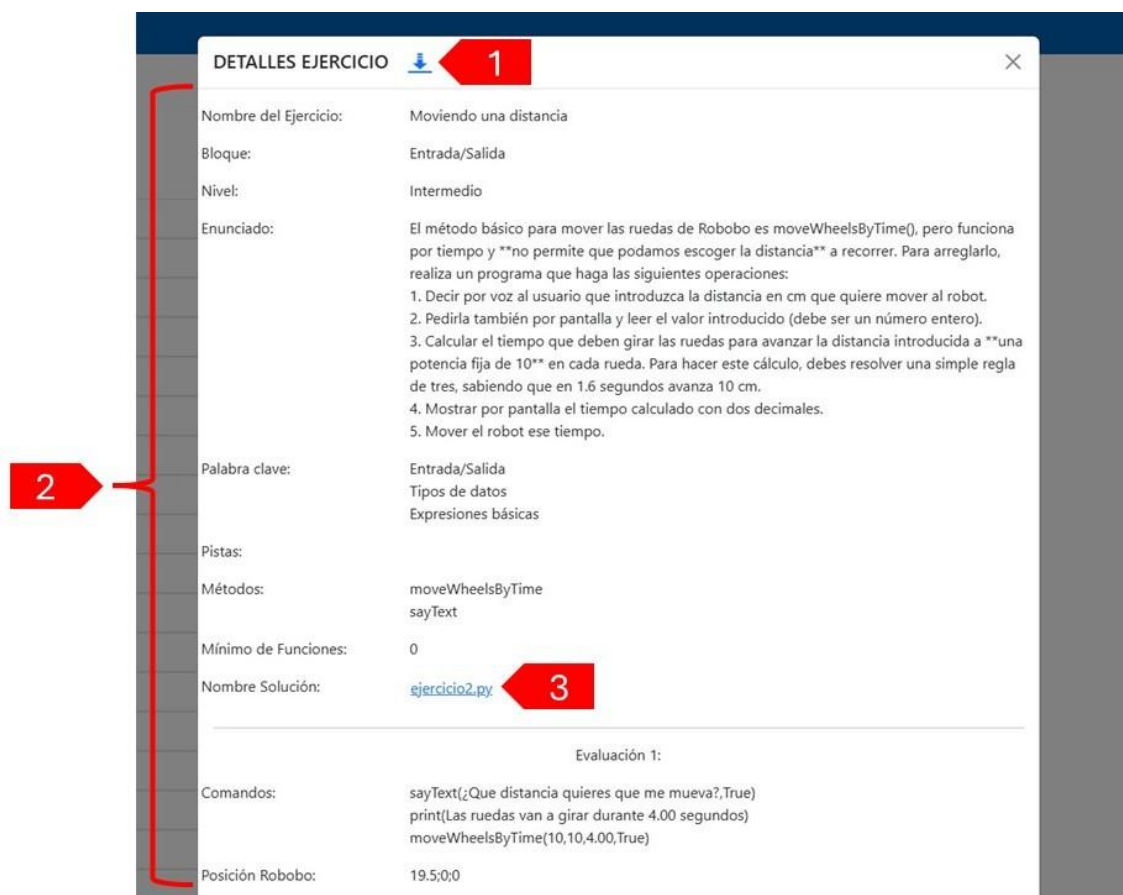
Este botón permite abrir el detalle de un ejercicio específico. Al hacer clic, se accede a una nueva pantalla con el enunciado completo y otros datos de interés

5. Filtro de Búsqueda

Este botón aplica los filtros seleccionados actualizando la tabla de ejercicios mostrados.

Navegación desde el Botón de acción "Visualizar"

A continuación, se describe la pantalla a la que se accede mediante el botón **"Visualizar" (4)**, junto con las funcionalidades disponibles numeradas en la siguiente imagen.



DETALLES EJERCICIO

Nombre del Ejercicio: Moviendo una distancia

Bloque: Entrada/Salida

Nivel: Intermedio

Enunciado: El método básico para mover las ruedas de Robobo es `moveWheelsByTime()`, pero funciona por tiempo y **no permite que podamos escoger la distancia** a recorrer. Para arreglarlo, realiza un programa que haga las siguientes operaciones:

1. Decir por voz al usuario que introduzca la distancia en cm que quiere mover al robot.
2. Pedirla también por pantalla y leer el valor introducido (debe ser un número entero).
3. Calcular el tiempo que deben girar las ruedas para avanzar la distancia introducida a **una potencia fija de 10** en cada rueda. Para hacer este cálculo, debes resolver una simple regla de tres, sabiendo que en 1.6 segundos avanza 10 cm.
4. Mostrar por pantalla el tiempo calculado con dos decimales.
5. Mover el robot ese tiempo.

Palabra clave: Entrada/Salida
Tipos de datos
Expresiones básicas

Pistas:

Métodos: `moveWheelsByTime`
`sayText`

Mínimo de Funciones: 0

Nombre Solución: [ejercicio2.py](#)

Evaluación 1:

Comandos: `sayText(¿Que distancia quieres que me mueva?,True)`
`print(Las ruedas van a girar durante 4.00 segundos)`
`moveWheelsByTime(10,10,4.00,True)`

Posición Robobo: 19.5;0;0

1. Descarga

Al clicar sobre el botón se descargará en su ordenador un archivo pdf que contendrá toda la información del ejercicio en cuestión, incluyendo los casos de uso y la solución en python del mismo en formato de texto plano.

2. Detalles del ejercicio

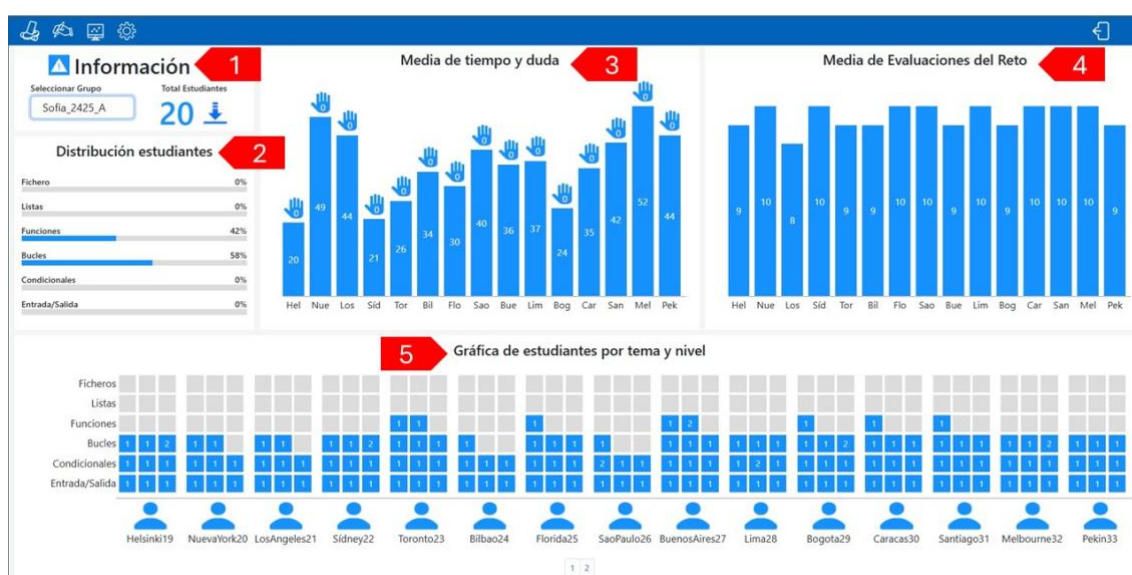
Muestra todos los detalles del ejercicio como nombre, tema de python al que pertenece, enunciado, casos de uso, etc. Además, muestra todos los casos de uso con los valores de interés.

3. Solución del ejercicio

Al clicar sobre las letras en azul se descarga en su ordenador el archivo python (.py) de la solución del ejercicio en cuestión. Este mismo método es aplicable para descargar el mundo y el video de cada caso de uso que aparecerán al hacer scroll sobre esta misma ventana.

Navegación desde “Menú Principal” a través del botón “Panel del Profesor”

A continuación, se describe la pantalla principal a la que se accede mediante el botón **“Panel del Profesor”** desde el **“Menú Principal”**, junto con las funcionalidades disponibles numeradas en la imagen correspondiente.



Esta pantalla muestra la información más relevante de todos los estudiantes de un grupo seleccionado. La pantalla está dividida en cinco secciones según el tipo de información que muestra, a continuación, se describe cada una:

1. Información

Esta sección contiene varios módulos. El primer paso es seleccionar el grupo que se desea visualizar. Para hacerlo, el profesor debe hacer clic en el botón **“Seleccionar Grupo”**, lo cual desplegará una lista con todos los grupos asignados a ese profesor. Desde ahí, podrá elegir el grupo que quiera consultar.

Una vez seleccionado el grupo, el sistema cargará automáticamente los datos de todos los estudiantes que lo conforman. En el campo **“Total Estudiantes”** se mostrará el número total de estudiantes del grupo. Junto a este campo se encuentra un botón de **“Descarga”**,

que se habilita tras la selección del grupo. Al hacer clic en este botón, se descargará un archivo en formato Excel con toda la información disponible de los estudiantes.

Además, en la parte superior derecha de esta sección, al lado del título **“Información”**, se encuentra el botón **“Avisos”**. Este botón parpadeará en color rojo si existe alguna incidencia en el aula que requiere la atención del profesor. Al hacer clic en él, se abrirá una nueva ventana donde se podrá revisar la información detallada de dichas incidencias (esta funcionalidad se explicará más adelante en el manual).

2. Distribución estudiantes

Esta sección muestra el porcentaje de estudiantes del grupo que actualmente están trabajando en ejercicios de un tema específico. Gracias a esta visualización, el profesor puede llevar un control detallado del avance del grupo dentro del curso

3. Media de tiempo y duda

Esta sección presenta una gráfica de barras que muestra el tiempo medio que tarda cada estudiante en resolver los retos del curso, calculado desde el momento en que se les asigna hasta que los completan. Es importante tener en cuenta que este valor corresponde a una media general por estudiante, no por reto específico. En la parte superior de cada barra, se muestra un ícono de mano, dentro del cual aparece el promedio de dudas resueltas por el profesor para ese estudiante.

4. Media de evaluaciones del Reto

Esta sección muestra la media de la nota propuesta por el modelo automatizado de calificación de ejercicios para cada estudiante. Es importante destacar que este valor tiene únicamente un propósito informativo, ya que ofrece una estimación de la calidad general de las soluciones entregadas por cada estudiante. En ningún caso debe utilizarse como calificación oficial, ni influir directamente en la evaluación académica del alumno.

5. Gráfica de estudiantes por tema y nivel

Esta sección presenta una gráfica de progreso individual para cada estudiante, que permite visualizar su avance en los distintos temas del curso.

Cada gráfica está organizada en 6 filas, correspondientes a los temas de programación del curso, y 3 columnas, que representan los niveles de dificultad dentro de cada tema.

- La barra de progreso se activa y se muestra en color azul cuando el estudiante comienza a resolver retos en un tema y nivel específicos.
- Dentro de cada barra, se indica el número de retos resueltos por el estudiante en esa categoría.
- A medida que el estudiante progresa, se irá actualizando la información visible en la gráfica.

Debajo de cada gráfica, se encuentra un botón con el nombre del estudiante. Al hacer clic en este botón, se abrirá una nueva ventana con información más detallada del desempeño de ese estudiante. Esta funcionalidad será explicada más adelante en el manual.

Navegación desde “Panel del Profesor” a través del botón “Panel del Estudiante”

A continuación, se describe la pantalla principal del Panel del Estudiante, a la cual se accede haciendo clic en el botón “Panel del Estudiante” desde el “Panel del Profesor”. Esta pantalla muestra toda la información disponible relacionada con un estudiante específico. Las funcionalidades y secciones están numeradas en la siguiente imagen para facilitar su identificación y comprensión.



1. Información y Control

Esta sección agrupa varias funcionalidades clave relacionadas con la gestión individual del estudiante:

- **Nombre del Alumno:** muestra el código único asignado al estudiante seleccionado. Es importante tener en cuenta que nunca se almacena el nombre real del estudiante, por lo que el profesor debe conocer a qué estudiante corresponde cada código.
- **Asignación de Nivel:** esta opción permite al profesor asignar manualmente un ejercicio diferente al estudiante, ya sea del mismo nivel o de un nivel superior al que le fue asignado automáticamente. Se recomienda utilizar esta función solo en casos excepcionales, ya que el sistema está diseñado para asignar retos que se ajustan al nivel de habilidad del estudiante de forma automática.
- **Avisos:** muestra un historial completo de todos los avisos registrados para ese estudiante. Estos avisos pueden incluir incidencias, intervenciones o anotaciones relevantes.

- **Ayuda:** este botón debe pulsarse cada vez que el profesor brinde asistencia directa al estudiante durante una clase, ya sea para resolver un ejercicio o aclarar una duda de programación. Al hacer clic, el sistema registrará automáticamente la ayuda en el reto que el estudiante esté resolviendo en ese momento.
- **Atrás:** permite regresar a la ventana anterior, es decir, al Panel del Profesor.

2. Listado de ejercicios realizados por el estudiante

Esta sección muestra el listado completo de los ejercicios que el estudiante ha realizado, clasificados y ordenados por tema y nivel de dificultad. Cada fila representa un ejercicio y contiene un cuadro de color al inicio que indica su estado:

- **Azul oscuro:** El reto fue resuelto correctamente por el estudiante.
- **Azul claro:** El estudiante no logró resolver el reto, y el profesor tuvo que asignarle uno diferente.

Además, cada fila incluye dos botones: **“Evaluación”** y **“Ver”**.

- **Evaluación:** Este botón muestra en texto la nota propuesta por el sistema automatizado para ese ejercicio. Las posibles clasificaciones son: Deficiente, Suspenso, Aprobado, Notable, Sobresaliente. Al hacer clic sobre este botón, se accede a una ventana con la explicación detallada de por qué el sistema asignó esa calificación. Esta ventana será explicada más adelante en el manual.
- **Ver:** Al pulsar este botón, se abre una pantalla dividida que permite comparar:
 - En un lado, el código fuente escrito por el estudiante.
 - En el otro lado, la solución propuesta por el profesor para ese mismo ejercicio.

Navegación desde “Panel del Profesor” a través del botón “Avisos”

A esta pantalla se accede haciendo clic en el botón “Avisos” desde el “Panel del Profesor”. En ella se muestra toda la información relacionada con los avisos generados por el sistema, los cuales están diseñados para alertar al profesor sobre situaciones que requieren su atención directa. La siguiente imagen muestra la apariencia de esta pantalla.

Un aviso puede generarse en los siguientes casos:

- Cuando un estudiante obtiene una nota baja en un ejercicio, según la evaluación propuesta por el sistema automatizado.
- Cuando el tiempo que un estudiante ha tardado en resolver un ejercicio supera el tiempo medio del grupo para ese mismo reto.



Fecha	Alumno	Nombre del reto	Bloque	Tipo de Alarma	Acción	Reconocer
18/06/2025	Toronto23	Moviendo una distancia	Entrada/Salida	Tiempo		Reconocer
19/06/2025	Toronto23	Contando billetes	Entrada/Salida	Tiempo		Reconocer
19/06/2025	Toronto23	Giros controlados	Condicionales	Tiempo		Reconocer
19/06/2025	Toronto23	Días de la semana	Condicionales	Tiempo		Reconocer
25/06/2025	Toronto23	Usando Pan y Tilt	Condicionales	Tiempo		Reconocer
25/06/2025	Toronto23	Giro con detección de obstáculos	Bucles	Tiempo		Reconocer

1. Botones de selección

Este panel permite filtrar los avisos que se desean visualizar según su estado. El profesor puede elegir entre las siguientes opciones:

- **Alarmas activas:** muestra únicamente los avisos que aún no han sido atendidos.
- **Alarmas reconocidas:** muestra únicamente los avisos que ya han sido gestionados.
- **Todas las alarmas:** muestra todos los avisos, sin importar su estado.

2. Listado de Avisos

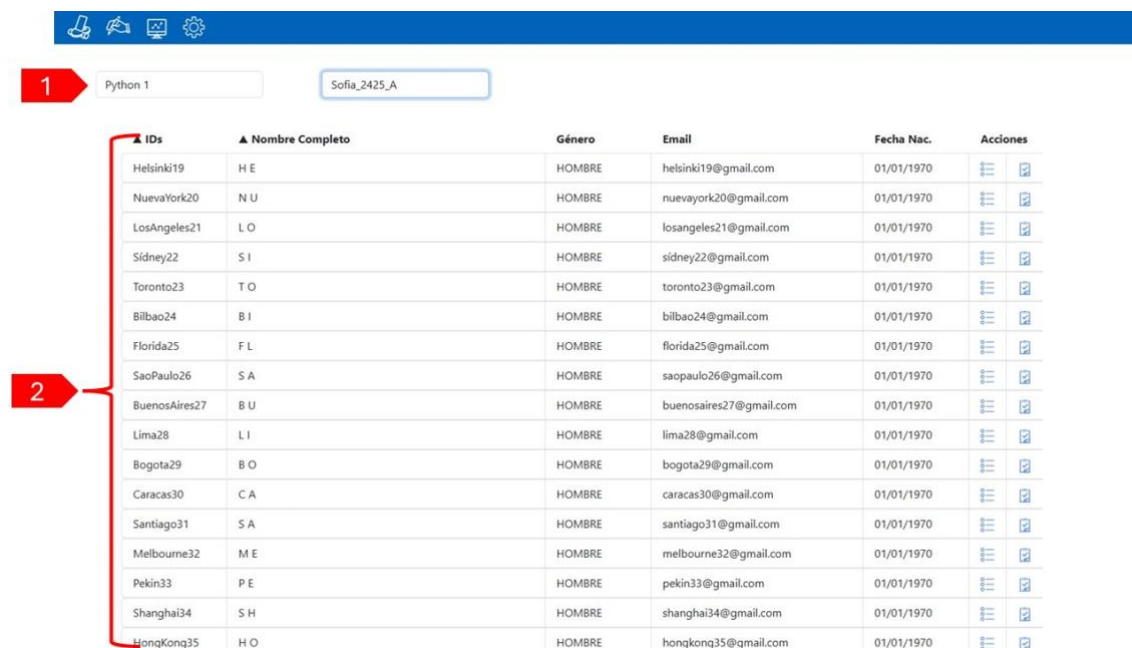
En esta sección se presenta un listado con todos los avisos generados por el sistema. Cada aviso incluye la siguiente información:

- **Fecha:** indica el día en que se generó el aviso.
- **Alumno:** identificador único del estudiante asociado al aviso.
- **Nombre del reto:** especifica el ejercicio en el que se presentó la situación.
- **Bloque:** muestra el tema de programación al que pertenece el reto.
- **Tipo de alarma:** señala la causa del aviso, como bajo rendimiento o tiempo excesivo de resolución.
- **Acción:** campo editable donde el profesor puede escribir o seleccionar la acción que desea tomar frente al aviso.
- **Botón "Reconocer":** al hacer clic en este botón, el aviso cambiará de estado, pasando de "Activo" a "Reconocido", lo que indica que ya fue revisado por el profesor.

Navegación desde "Menú Principal" a través del botón "Configuración"

Esta pantalla se accede al hacer clic en el botón "Configuración", ubicado en el Menú Principal. Desde aquí, el profesor puede visualizar los datos de registro de todos los

estudiantes pertenecientes a un grupo específico, así como realizar acciones personalizadas sobre cada uno de ellos. A continuación, se describen las funcionalidades disponibles en cada sección de esta ventana, con el objetivo de facilitar su uso y gestión.



IDs	Nombre Completo	Género	Email	Fecha Nac.	Acciones
Helsinki19	H E	HOMBRE	helsinki19@gmail.com	01/01/1970	[icon] [icon]
NuevaYork20	N U	HOMBRE	nuevayork20@gmail.com	01/01/1970	[icon] [icon]
LosAngeles21	L O	HOMBRE	losangeles21@gmail.com	01/01/1970	[icon] [icon]
Sidney22	S I	HOMBRE	sidney22@gmail.com	01/01/1970	[icon] [icon]
Toronto23	T O	HOMBRE	toronto23@gmail.com	01/01/1970	[icon] [icon]
Bilbao24	B I	HOMBRE	bilbao24@gmail.com	01/01/1970	[icon] [icon]
Florida25	F L	HOMBRE	florida25@gmail.com	01/01/1970	[icon] [icon]
SaoPaulo26	S A	HOMBRE	saopaulo26@gmail.com	01/01/1970	[icon] [icon]
BuenosAires27	B U	HOMBRE	buenosaires27@gmail.com	01/01/1970	[icon] [icon]
Lima28	L I	HOMBRE	lima28@gmail.com	01/01/1970	[icon] [icon]
Bogota29	B O	HOMBRE	bogota29@gmail.com	01/01/1970	[icon] [icon]
Caracas30	C A	HOMBRE	caracas30@gmail.com	01/01/1970	[icon] [icon]
Santiago31	S A	HOMBRE	santiago31@gmail.com	01/01/1970	[icon] [icon]
Melbourne32	M E	HOMBRE	melbourne32@gmail.com	01/01/1970	[icon] [icon]
Pekin33	P E	HOMBRE	pekin33@gmail.com	01/01/1970	[icon] [icon]
Shanghai34	S H	HOMBRE	shanghai34@gmail.com	01/01/1970	[icon] [icon]
HongKong35	H O	HOMBRE	hongkong35@gmail.com	01/01/1970	[icon] [icon]

1. Selección de Curso y Grupo

En la parte superior de la pantalla se encuentran dos botones desplegables que permiten al profesor seleccionar el curso correspondiente y el grupo dentro de ese curso. Una vez seleccionados, se mostrará automáticamente el listado de estudiantes registrados en el grupo elegido, junto con sus datos asociados.

2. Listado de estudiantes

Una vez seleccionado el curso y grupo, se despliega el listado completo de estudiantes registrados en ese grupo. Para cada estudiante se muestran sus datos de registro básicos (como el código asignado, nivel, y otra información relevante) y dos botones de acción específicos: Asignar Reto y Visualizar Retos. Las funcionalidades de ambos botones, junto con las pantallas asociadas, se explican en las siguientes secciones del manual.

Navegación desde “Configuración” a través del botón “Asignar Retos”

Al hacer clic en el botón “Asignar Reto” junto al nombre de un estudiante, se abre una ventana que muestra un listado con todos los retos disponibles para asignar. Este listado omite automáticamente aquellos retos que el estudiante ya haya tenido asignados anteriormente. Para realizar la asignación, el profesor solo necesita seleccionar el reto deseado marcando la casilla correspondiente y luego presionar el botón “Aceptar”, ubicado al final de la ventana. Una vez hecho esto, el reto quedará asignado al estudiante de forma inmediata y podrá comenzar a trabajar en él desde su entorno habitual.

ASIGNAR NUEVO RETO

Seleccione el nuevo reto para: **H E** con ID: **Helsinki19**

☐	Encendiendo LEDs	Entrada/Salida	Inicial
☐	Moviendo una distancia	Entrada/Salida	Intermedio
☐	Mostrando emociones	Entrada/Salida	Intermedio
☐	Saludando a un amigo	Entrada/Salida	Avanzado
☐	Moviendo motores	Entrada/Salida	Inicial
☐	Dando vueltas	Entrada/Salida	Avanzado
☐	Giros a potencia fija	Entrada/Salida	Intermedio
☐	Primeras palabras	Entrada/Salida	Inicial
☐	Contando billetes	Entrada/Salida	Avanzado
☐	Reaccionando a la edad	Condicionales	Inicial
☐	Giros controlados	Condicionales	Inicial
☐	Aprobado o suspenso	Condicionales	Inicial
☐	Días de la semana	Condicionales	Intermedio
☐	Tipos de triángulo	Condicionales	Intermedio
☐	Motor Pan	Condicionales	Intermedio
☐	Coloreando emociones	Condicionales	Avanzado
☐	Usando Pan y Tilt	Condicionales	Avanzado
☐	Moviendo a Robobo	Condicionales	Avanzado
☐	Ferrol municipal	Burlesco	Inicial

Navegación desde “Configuración” a través del botón “Visualizar Retos”

Al hacer clic en el botón “Visualizar Retos”, se abre una ventana emergente donde se pueden consultar todos los retos que han sido asignados al estudiante. En esta ventana también se muestra el estado actual de cada reto, lo que permite al profesor verificar si el estudiante ya lo resolvió, si está en curso o si aún no ha sido iniciado.

Lista de Retos Asignados

Retos del alumno: **H E** con ID: **Helsinki19**

Encendiendo LEDs	Entrada/Salida	Inicial	Finalizado y Resuelto
Mostrando emociones	Entrada/Salida	Intermedio	Finalizado y Resuelto
Saludando a un amigo	Entrada/Salida	Avanzado	Finalizado y Resuelto
Reaccionando a la edad	Condicionales	Inicial	Finalizado y Resuelto
Motor Pan	Condicionales	Intermedio	Finalizado y Resuelto
Coloreando emociones	Condicionales	Avanzado	Finalizado y Resuelto
Frenado progresivo	Bucles	Inicial	Finalizado y Resuelto
Armonía impar	Bucles	Intermedio	Finalizado y Resuelto
Alcanzar color	Bucles	Avanzado	Finalizado sin Resolver
Aparcamientos disponibles	Bucles	Avanzado	En proceso...

Aceptar

Metodología de enseñanza

A continuación, se propone una metodología para introducir ProgTutor en la enseñanza reglada, que combina sesiones guiadas por el profesor con otras en las que los alumnos usan la herramienta. Ha sido probada y validada durante dos cursos académicos con 2 grupos de estudiantes de 1º de bachillerato en Galicia.

Antes de comenzar, la disposición física del aula debe facilitar tanto el uso individual de ProgTutor como las explicaciones del docente durante las clases magistrales. Por tanto, debe haber un ordenador por estudiante, que puede ser de sobremesa o portátil. Sin embargo, en ambos casos, es imprescindible el uso de un ratón externo, ya que los *touchpads* no son adecuados para manejar el simulador. También es necesario un proyector o pantalla grande para las presentaciones y demostraciones del profesorado, así como una conexión a internet estable para acceder a los recursos en línea.

En cuanto a las especificaciones técnicas, es crucial que todos los ordenadores cumplan con los requisitos mínimos para ejecutar el software correctamente. Se necesita un monitor con resolución adecuada para visualizar claramente tanto el simulador como el editor de código, y el software necesario debe estar previamente instalado. Los requisitos aparecen detallados en la siguiente tabla:

REQUISITOS DEL ORDENADOR.

Hardware	Sistema Operativo	Software Aplicación
4-8 GB RAM	Ms Windows 7	Python 3
Intel i3 9300 o superior	o superior	Visual Studio Code
(y equivalentes de AMD)	MacOS 11 o superior	Extensión Jupyter para VSC
0.5-1 GB de espacio en disco	Debian 10	Librería RoboBoPy
Resolución: 1920x1080	(y distribuciones	Plugin ProgTutor_vsc
Ratón	equivalentes)	ProgTutorSim

*Requiere conexión a Internet.

**No se requiere una tarjeta gráfica dedicada, pero se recomienda.

En su defecto, se recomienda una CPU con chip gráfico integrado.

En el desarrollo de ProgTutor, varios **profesores de programación y robótica de la Universidad de A Coruña**, con muchos años de experiencia en la enseñanza de estas materias, han diseñado ejercicios que abordan **cuatro temas fundamentales de programación**: entrada/salida, condicionales, bucles y funciones. Cada uno de estos temas ha sido **integrado con conceptos de robótica** para mejorar la comprensión práctica y su aplicación, tal como se detalla en la siguiente tabla:

CONTENIDOS DE PROGRAMACIÓN POR NIVEL DE DIFICULTAD

Bloque	Nivel	Contenidos de programación	Sensores y actuadores
Entrada/Salida	1	Entrada y salida simples sin conversión de tipo, usando una expresión básica (fórmula) para calcular valores. Enfoque en variables simples.	Actuadores simples para mover el robot (mover ruedas por tiempo) y expresarse (decir texto, reproducir sonidos o activar LEDs).
	2	Entrada con conversión de tipo, manejo de expresiones matemáticas más complejas y salida de un tipo de dato diferente. Múltiples tipos de variables.	
	3	Incluye todo el contenido del nivel 2, pero con un algoritmo o expresión ligeramente más complejo.	
Condicionales	1	Condicionales simples, usando estructuras if-else o if-elif-else. Condiciones simples o enlazadas usando un único operador and/or.	Actuadores simples para mover la cara del robot y establecer diferentes emociones. También incluye sensores para detectar la posición de la cara.
	2	Múltiples sentencias elif. Condiciones complejas, usando múltiples operadores and/or.	
	3	Estructuras anidadas if-elif-else. Mayor complejidad algorítmica.	
Bucles	1	Bucles para validar datos de entrada e iterar como parte de la implementación del algoritmo.	Se añaden actuadores para mover el robot (mover ruedas sin control por tiempo). Se incrementa el número de sensores para detectar más información sobre el robot (posición de las ruedas, orientación del robot, identificación de Blobs).
	2	Bucle para validar datos de entrada, combinado con una sentencia condicional dentro del bucle para manejar el algoritmo.	
	3	Bucle para validar datos de entrada, combinado con bucles anidados o ejercicios de mayor complejidad algorítmica.	
Funciones	1	Adaptación de ejercicios de bucle del nivel 2. El enunciado especifica las funciones a implementar, incluyendo parámetros de entrada y salida.	Se consideran sensores más complejos para detectar QR y objetos.
	2	El enunciado indica el número de funciones a realizar, pero sin indicar sus parámetros.	
	3	El enunciado no indica el número de funciones. Aun así, se insta al alumnado a estructurar adecuadamente su programa y la evaluación verificará que haya un número mínimo de funciones.	

Para adaptarse a las diferentes necesidades de aprendizaje, **cada tema de programación se divide en tres niveles de dificultad:**

- Nivel 1: Principiante
- Nivel 2: Intermedio
- Nivel 3: Avanzado

Esta estructura escalonada garantiza que los estudiantes puedan **desarrollar progresivamente sus competencias**, comenzando por los conceptos básicos y avanzando hacia escenarios de robótica más complejos (ver Tabla anterior). Es importante destacar que, aunque cada tema introduce nuevos conceptos, la programación es una **disciplina incremental**, por lo que los conceptos aprendidos en temas anteriores se **integran en los desafíos de los nuevos temas**, reforzando el conocimiento previo a medida que los estudiantes se enfrentan a tareas más avanzadas.

Del mismo modo que ocurre con los conceptos de programación, las **funciones del robot se introducen de forma progresiva**: desde métodos simples como encender un LED o mostrar una emoción, hasta sensores más complejos capaces de detectar un código QR (véase la última columna de la Tabla anterior). Sin embargo, es importante recordar que el objetivo principal de ProgTutor **no es enseñar robótica, sino programación informática**.

La metodología de enseñanza propone una **estructura combinada de sesiones autodirigidas (ProgTutor) y otras llevadas por el docente**. El curso está diseñado en sesiones de 50 minutos, de acuerdo con la duración habitual de las clases en la educación secundaria en Galicia, aunque este formato puede adaptarse a otros contextos educativos. También es flexible el número de sesiones por tema, que puede variar según el criterio docente. Se propone un rango total de entre **14 y 20 sesiones**, en función del ritmo del alumnado.

La **planificación de sesiones** propuesta aparece en la figura siguiente:



Las **dos primeras sesiones** (parte superior de la figura) son de carácter introductorio y sirven para presentar a los estudiantes tanto los conceptos básicos del lenguaje Python como las herramientas que utilizarán:

Sesión 1 – Familiarización con las herramientas:

- Configuración y uso básico de Visual Studio Code (VSC).
- Uso de Jupyter Notebooks como entorno de aprendizaje interactivo.
- Repaso de fundamentos básicos de Python mediante un cuaderno introductorio (que también permite comprobar la configuración de VSC).
- Explicación de la documentación disponible de Python.

Sesión 2 – Introducción a RoboboSim:

- Presentación del simulador y de los mundos de simulación disponibles.

- Programación del robot Robobo con la librería robobo.py, guiando al alumnado paso a paso.
- Explicación de la documentación de robobo.py.

A continuación, se comienza un ciclo de aprendizaje que se repite para cada tema de Python. Las sesiones siguen un patrón cíclico con **cinco objetivos principales**:

1. Explicación teórica del tema.
2. Ejercicios de Python sin simulador.
3. Ejercicios de Python con simulador.
4. Sesiones con el ITS *ProgTutor*.
5. Examen.

Cada ciclo comienza con la explicación de un tema (según los establecidos en la tabla anterior) mediante un cuaderno de *Jupyter*. Esta parte puede abarcar una o varias sesiones, dependiendo de la complejidad. Por ejemplo, esta podría ser una distribución de sesiones por tema:

Tema	Teoría	Prácticas sin simulador	Sesiones ITS (mín. - máx.)
Entrada/Salida (E/S)	1	1	3 – 6
Condicionales	1	2	3 – 6
Bucles	2	3	6 – 9
Funciones	1	3	6 – 9

Después de la explicación teórica, se realizan **sesiones prácticas sin simulador**, centradas en los fundamentos del tema, utilizando programas simples con salida en consola. El profesorado debe guiar estas actividades, facilitando al alumnado una solución en Python como plantilla de referencia.

En la siguiente sesión, el docente resuelve **ejercicios con el simulador RoboboSim**, explicando los nuevos métodos de robobo.py relacionados con el tema. Esta sesión no es una clase magistral, sino práctica guiada.

En las siguientes sesiones, el alumnado trabaja con los ejercicios propuestos **por ProgTutor ITS, de manera autónoma**, mientras el docente actúa como facilitador. La primera vez que se usa el ITS, se debe explicar la interfaz al alumnado.

Todos los estudiantes comienzan en el mismo nivel inicial del ITS (primer tema y nivel 1 de dificultad). El ITS permite avanzar al ritmo de cada estudiante, lo que puede dificultar la coordinación grupal. Para evitar una gran disparidad de progreso, se proponen los siguientes **criterios para avanzar de tema**:

- Cuando **2/3 del grupo ha completado los 3 niveles** y se han superado las sesiones mínimas indicadas, se activa la fase final del ciclo: un **examen**.
- Los estudiantes que finalicen antes pueden estudiar de forma autónoma el siguiente *notebook* y resolver ejercicios simples en consola.
- Si se alcanzan las sesiones máximas sin que 2/3 del alumnado haya avanzado, se recomienda dedicar 1 o 2 sesiones extra para ofrecer más apoyo.

Una vez cumplidas las condiciones, se realiza un examen individual sobre el tema. **Incluso quienes no hayan terminado los niveles del ITS deben hacer el examen**, para evitar una dispersión excesiva. Se recomienda que el examen consista en ejercicios de programación sin el simulador, reforzando que el objetivo es **aprender a programar en Python en general, no solo con desafíos robóticos**.

Después del examen, el ciclo se repite con el siguiente tema.