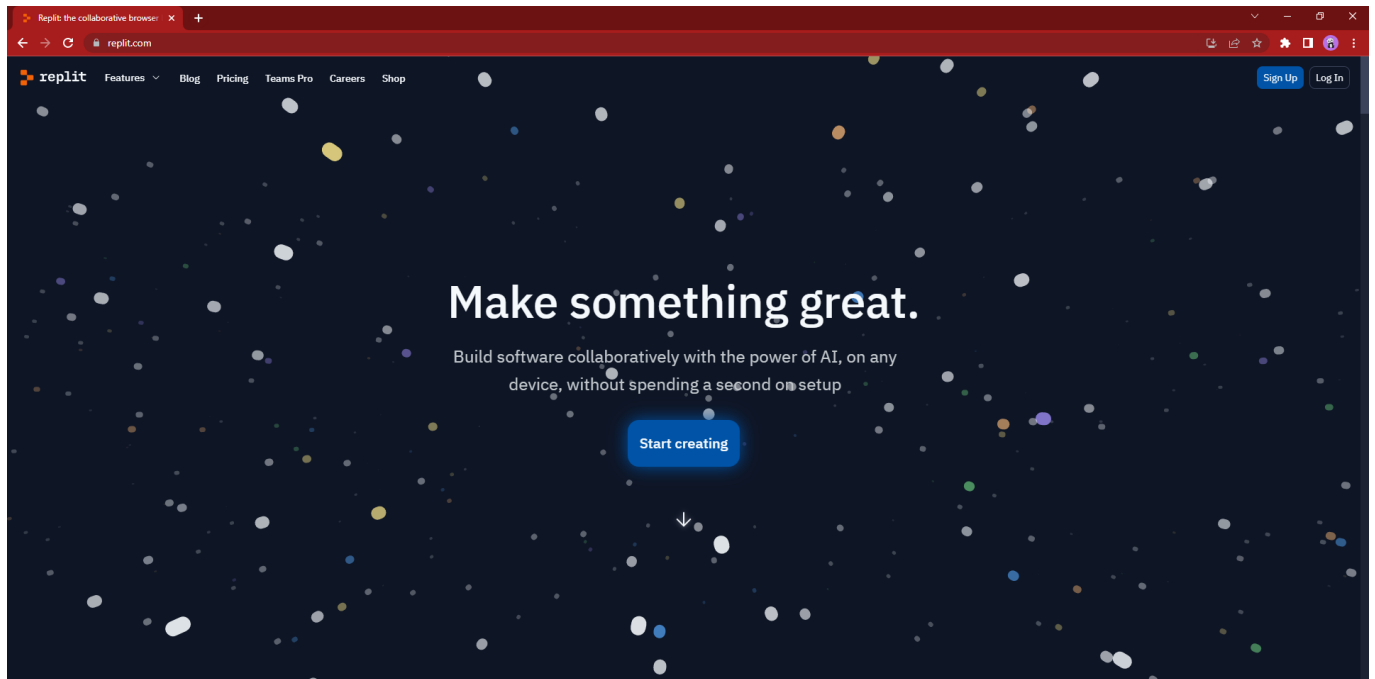


Instructivo Replit

Replit es un entorno de desarrollo integrado o *IDE* en línea.



¿Por qué Replit?

La plataforma fue la elegida para el desarrollo de los programas solicitados para el experimento por varias razones:

- No requiere instalación, ni del IDE ni de ninguna de las dependencias de los lenguajes (node, intérprete de Python, etc.)
- Facilita la recolección de las soluciones

Instructivo preparación entorno

El instructivo de preparación consta sólo de dos pasos.

1. Unirse a Equipo Tesina

El proyecto estará alojado en un equipo educativo de Replit llamado **Tesina**.

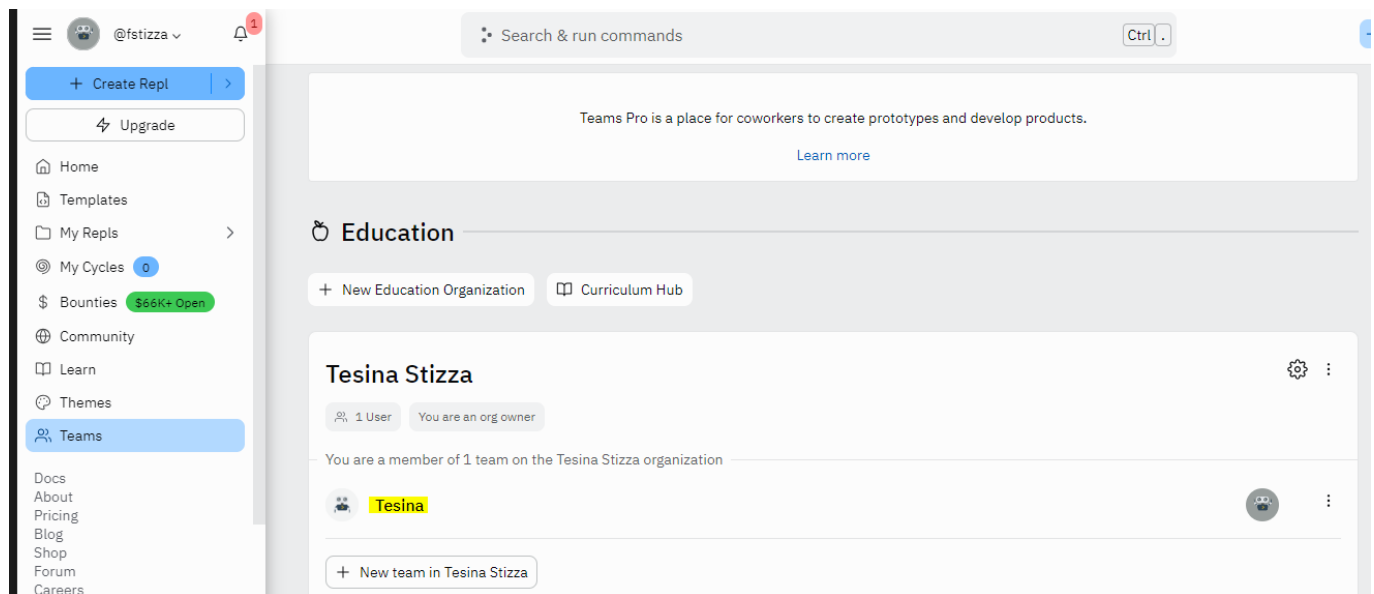
Haga click en el siguiente [enlace](#).

Si no posee cuenta de replit le solicitará crear una y verificar el correo electrónico.

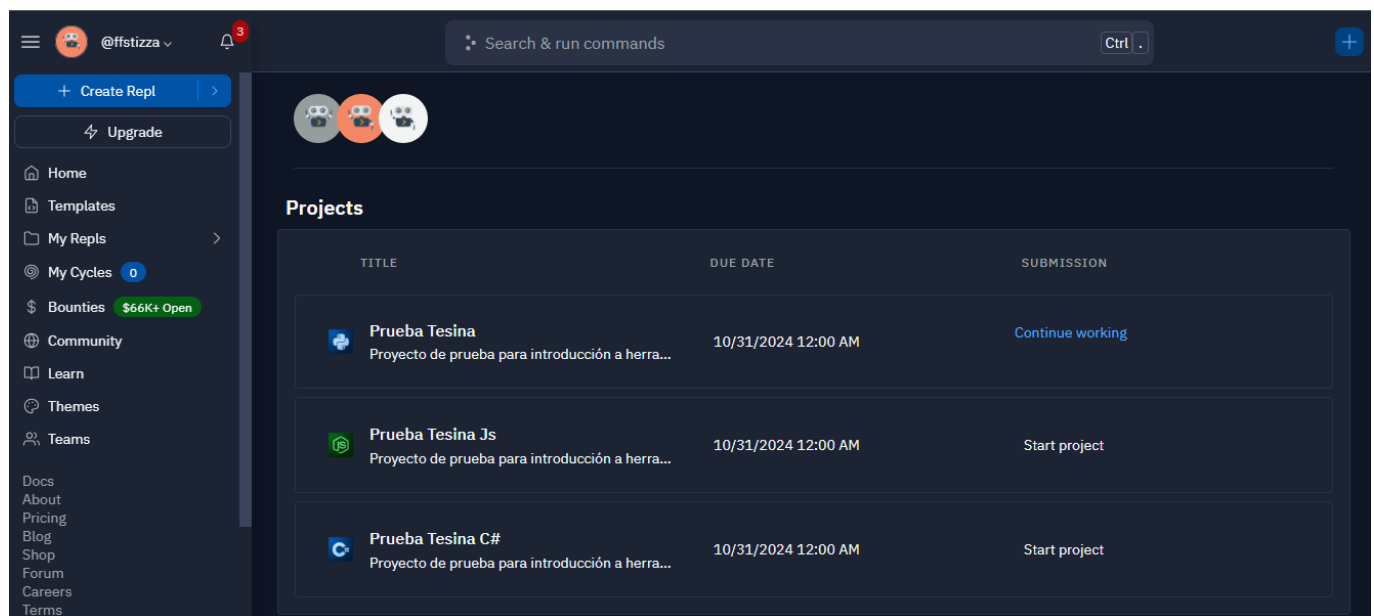
2. Menú equipo de Tesina

Una vez creada su cuenta y que haya aceptado la invitación al equipo podrá navegar al menú del equipo.

Navegue con este [link](#) o utilice el menú *Teams* en la barra lateral izquierda de la web y luego haga click en **Tesina**.



Luego encontrará la página de los proyectos del equipo.

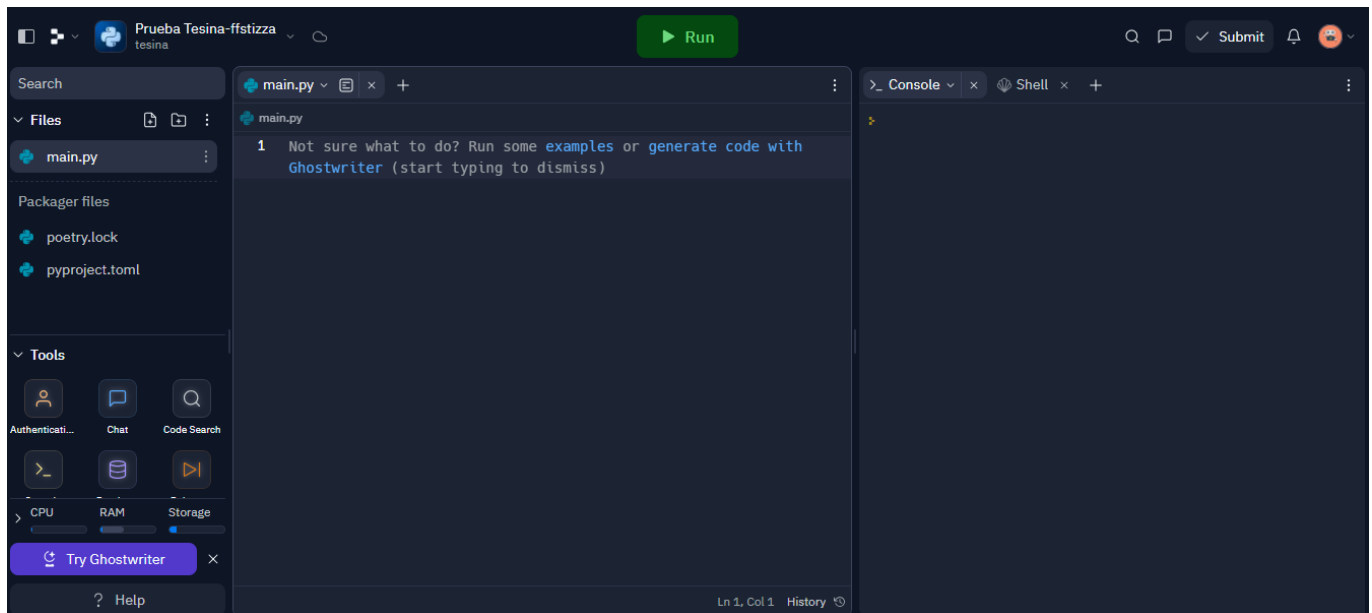


Por el momento podrá visaulizar tres proyectos de prueba.

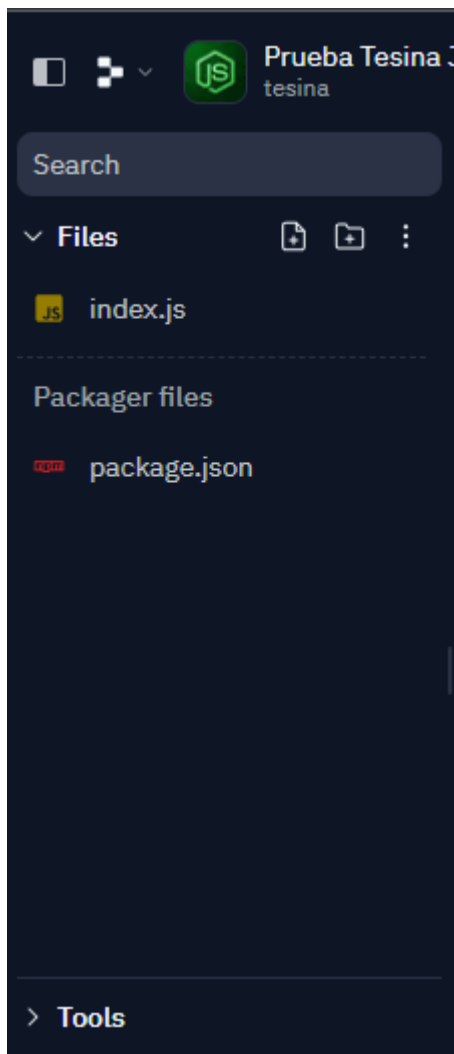
Cada proyecto corresponde a uno de los lenguajes disponibles para programar la solución del experimento.

Siéntase libre de navegar por los proyectos, escribir y ejecutar código.

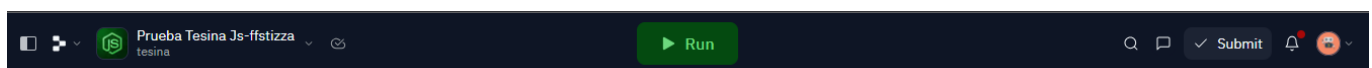
Explicación de características del entorno



El entorno de desarrollo es bastante dinámico, se pueden cerrar las ventanas, juntarlas, dejarlas flotando, etc.



En la barra lateral izquierda encontrará un gestor de archivos simple.

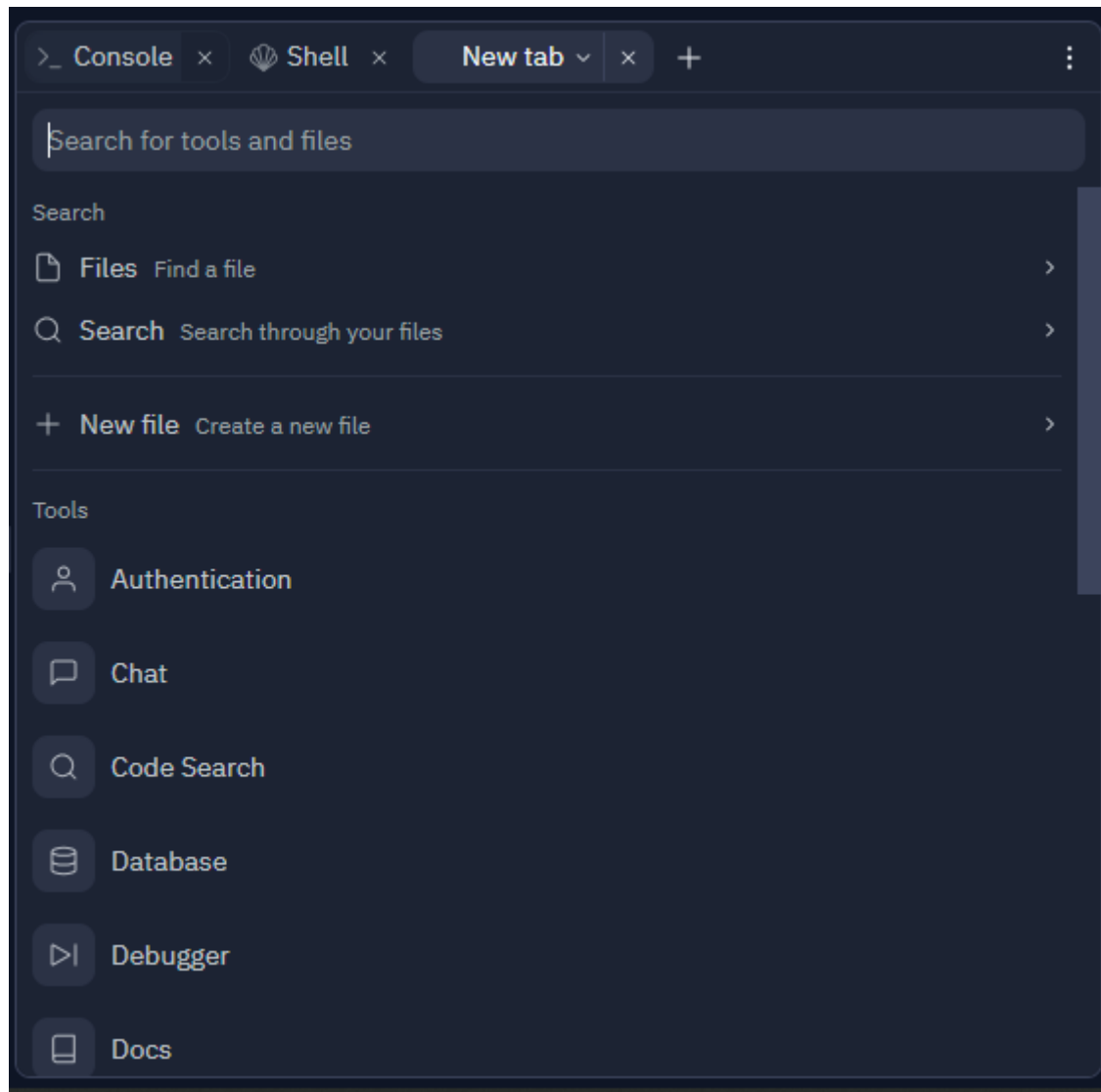


En la barra superior se puede correr el programa principal que en el caso de los proyectos en *C#* corresponde al archivo **main.cs**, en *JavaScript* al **index.js** y en *Python* a **main.py**.

En la barra superior también se encuentra el botón de **Submit** que es para enviar definitivamente la solución. Una vez que haya concluido el desarrollo es obligatorio enviar la solución.

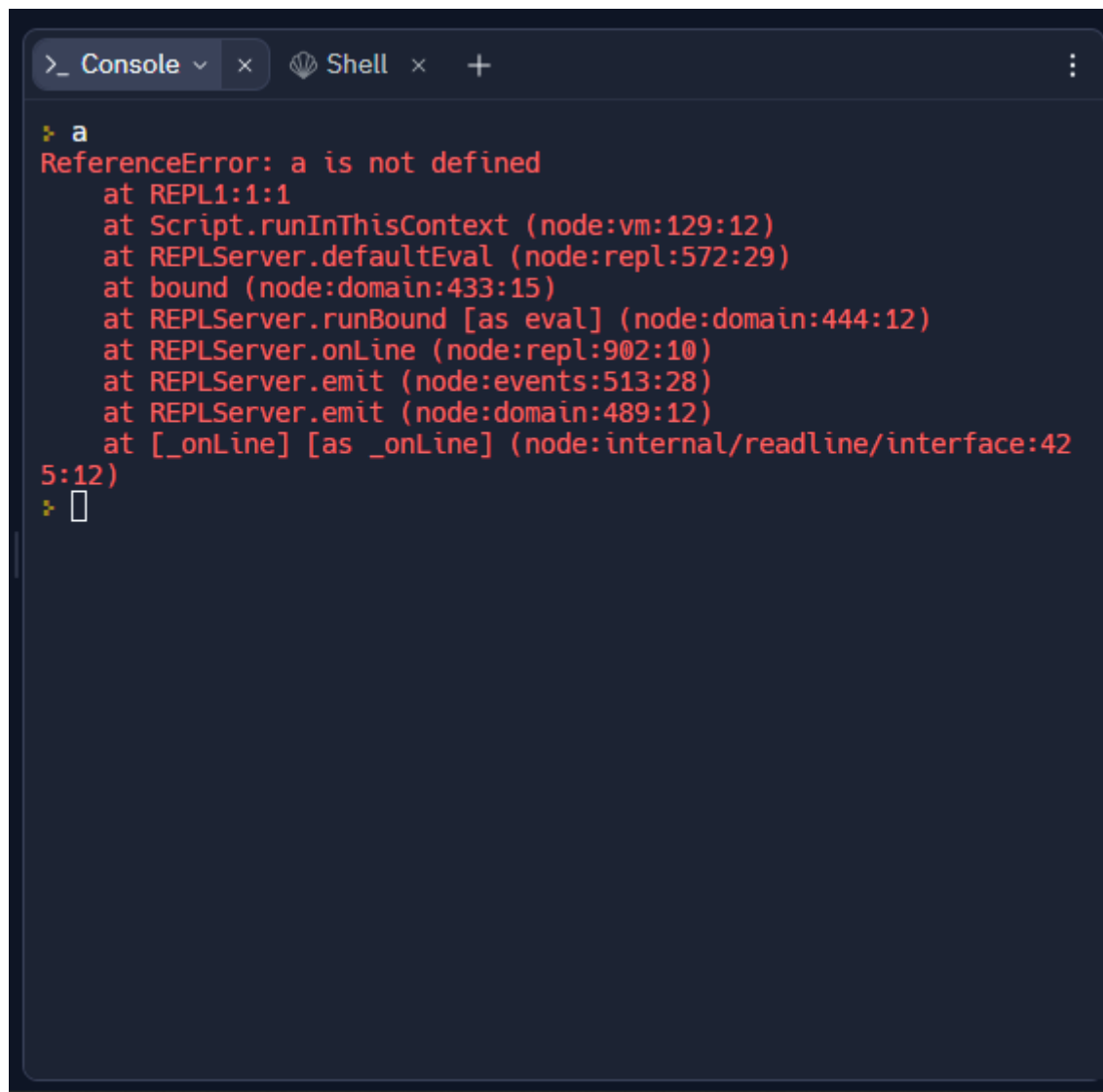
Como se mencionó anteriormente, el IDE es bastante dinámico y por error se puede haber cerrado alguna ventana que era de interés.

Utilizando el botón **+** que se encuentra en la barra superior de los archivos abiertos podrá volver a abrir la consola el depurador, entre otras cosas.



Console y Shell

Dos de las herramientas más importantes son **Console** y **Shell** que en muchos contextos puede significar lo mismo pero en *Replit* no.

A screenshot of the Replit interface showing a console window. The console has a dark background with light-colored text. At the top, there are tabs for 'Console' and 'Shell'. The 'Console' tab is active, displaying a red error message: 'ReferenceError: a is not defined'. Below the error message, there is a stack trace with several lines of code and line numbers. The error occurred at line 1, column 1 of the current script. The stack trace includes frames for 'Script.runInThisContext', 'REPLServer.defaultEval', 'bound', 'REPLServer.runBound', 'REPLServer.onLine', 'REPLServer.emit', and '[_onLine]'. The console ends with a prompt character 'a' and a cursor.

La **Console** es una ventana donde se pueden ejecutar comandos interactivos según el lenguaje en el que se esté desarrollando, por ejemplo en **Python** corre un *intérprete* interactivo, en **JavaScript** una consola de *Node*.

Además a través del **Console** se vuelca la salida estándar del programa que está corriendo.

En el caso de **C#** no se posee tal característica.

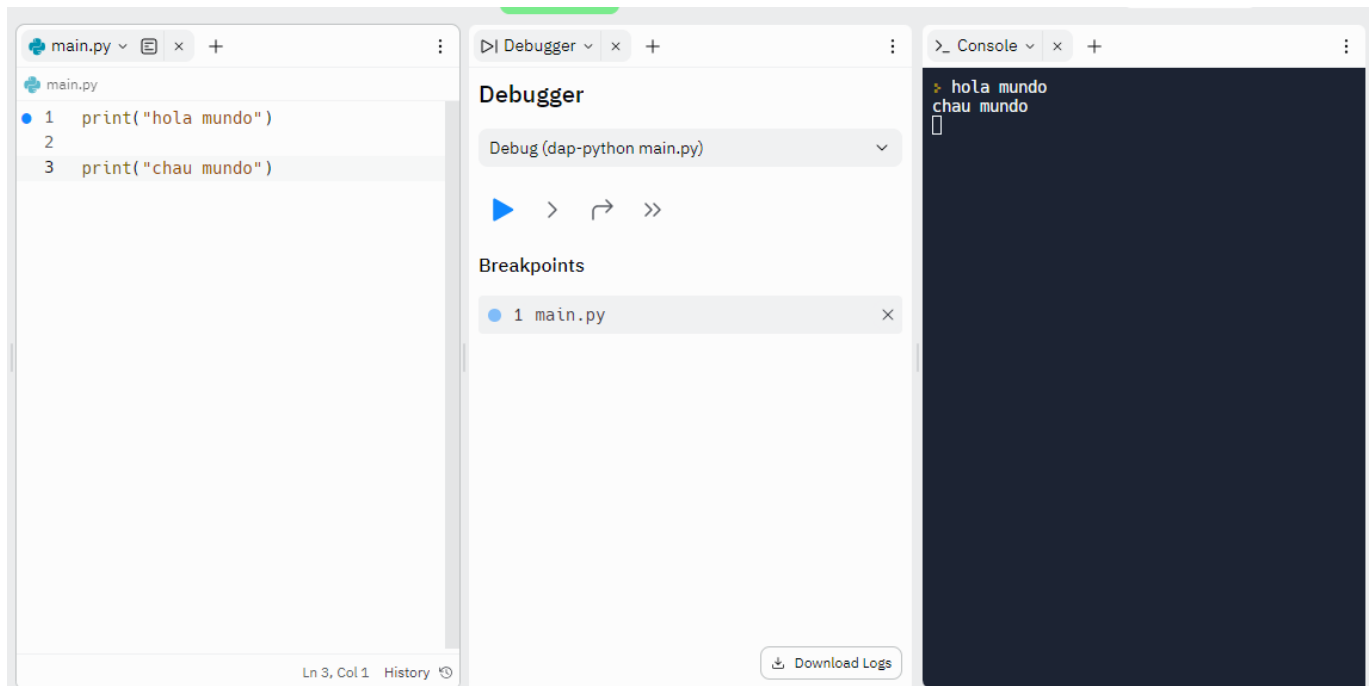
Por otro lado el **Shell** es una terminal **bash** que corre en el entorno que se está desarrollando, aquí podrá correr por ejemplo **ls** para listar los archivos, correr los scripts, etc.

Depurador

Replit tiene depuradores integrados para proyectos en **Python** y **JavaScript**.

Lamentablemente **no** soporta **C#**.

En estos proyectos se puede abrir en la caja de herramientas **Tools** que se ubica en la barra lateral izquierda o abriendo una nueva ventana con el botón **+**.



Para agregar puntos de interrupción hay que clickear a la izquierda de la línea de código donde se desea agregar.

Recordar tener abierto un **Console** para poder visualizar la salida estándar del programa.

Pruebas

Puede correr o definir pruebas en la sección de Tests ubicada en la caja de herramientas en la barra lateral izquierda o usando el +.

Ejecutar programas

Utilizando la pestaña **Shell**, se pueden ejecutar los programas de línea de comandos y evaluar el código de retorno del programa.

- **Python:** `python3 programa.py` y luego `echo $?`
- **JavaScript:** `node programa.js` y luego `echo $?`
- **C#:** `dotnet run` y luego `echo $?`

Recordar que para especificar el código de salida del programa se debe utilizar:

```
if __name__ == '__main__':  
    exit(1)
```

```
process.exit(1);
```

```
System.Environment.Exit(1);
```

Además los programas en **Python** deben tener la sintaxis `if __name__ ...` para ejecutarse como script.