

## 0.1. Manual de instalación del proyecto

En primer lugar, debemos de asegurarnos de tener instalado NodeJS<sup>1</sup> en la máquina en la que vaya a ser ejecutado el proyecto, ya sea un ordenador personal o en un servidor con exposición directa a internet.

Una vez la instalación del entorno de ejecución esté lista, procedemos a descargar el código fuente del proyecto. En este caso, este se encuentra guardado en un repositorio *git* privado, por lo que se necesitará permisos de colaborador para poder acceder a él.

```
$ git clone https://github.com/DavidGomez-coder/TFG.git
```

A continuación, debemos de abrir una consola de comandos y situarnos en el directorio raíz del proyecto, para instalar así todas las dependencias necesarias.

```
$ npm install
```

Cuando este proceso finalice iniciamos el proceso de la aplicación ejecutando el siguiente comando:

```
$ npm start
```

, tras el cuál se abrirá en una nueva pestaña del navegador web por defecto en la dirección <http://localhost:3000/TFG>; o si estamos utilizando un entorno de servidor, este se podrá acceder sustituyendo *localhost* por la dirección IP de la máquina en cuestión.

## 0.2. Manual de usuario de la aplicación

Lo primero que vemos al iniciar la aplicación, es la siguiente interfaz presentada en la figura 1, en la que podemos distinguir claramente una división en dos de la misma, teniendo en la parte superior un panel de navegación que nos permite cambiar entre las dos simulaciones y una parte central con el contenido principal.

Para poder diferenciar entre ambos circuitos y saber qué está ocurriendo, se añade una imagen de los mismos sobre el que aparecerán electrones en movimiento, ilustrando así el proceso que está ocurriendo durante la simulación (figura 2). Además, para los estados de dispersión de energía, a la fuente se le da cierta transparencia, para emular de alguna manera

---

<sup>1</sup><https://nodejs.org/en/download/>

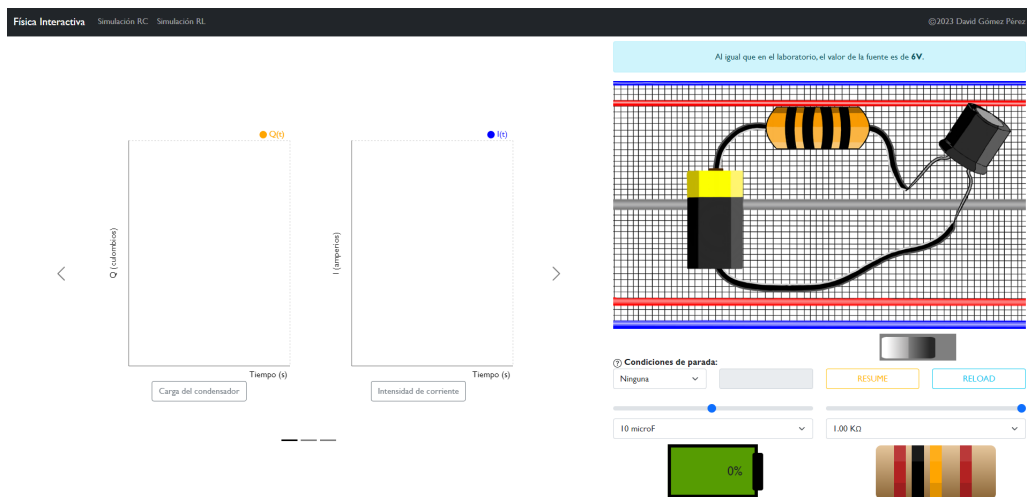


Figura 1: Interfaz de usuario de la aplicación (I).

que esta ha sido quitada del circuito.

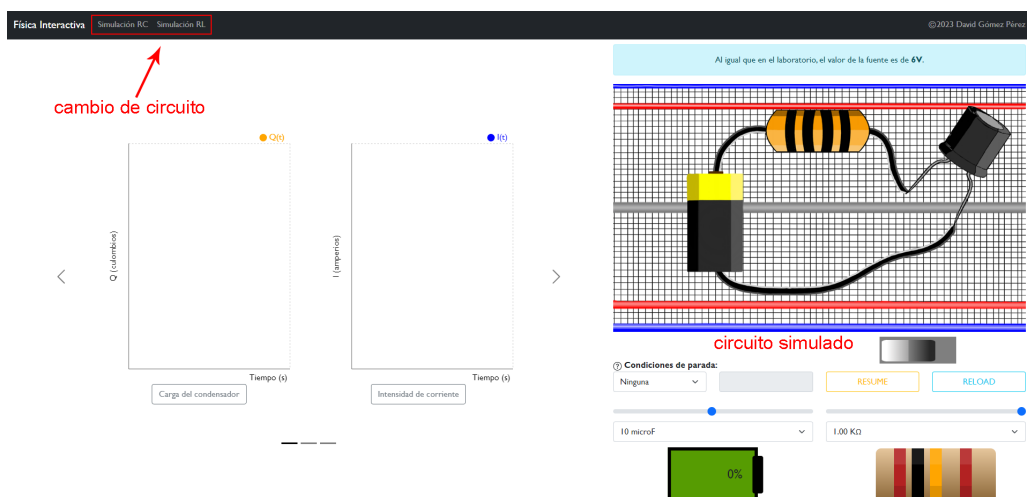


Figura 2: Interfaz de usuario de la aplicación (II).

Junto a la representación del circuito, tenemos la sección dónde se van a mostrar los resultados de cada una de las magnitudes físicas, por las que si pasamos el cursor podremos ver con más en detalle los valores que estas toman (figura 3) .

En el centro de la pantalla, tenemos los controladores de la simulación. Por un lado en la parte superior se encuentra el controlador de las condiciones de parada, las cuáles se encuentran en un *dropdown* (lista desplegable) junto a un *input* que toma el valor de dicha condición.

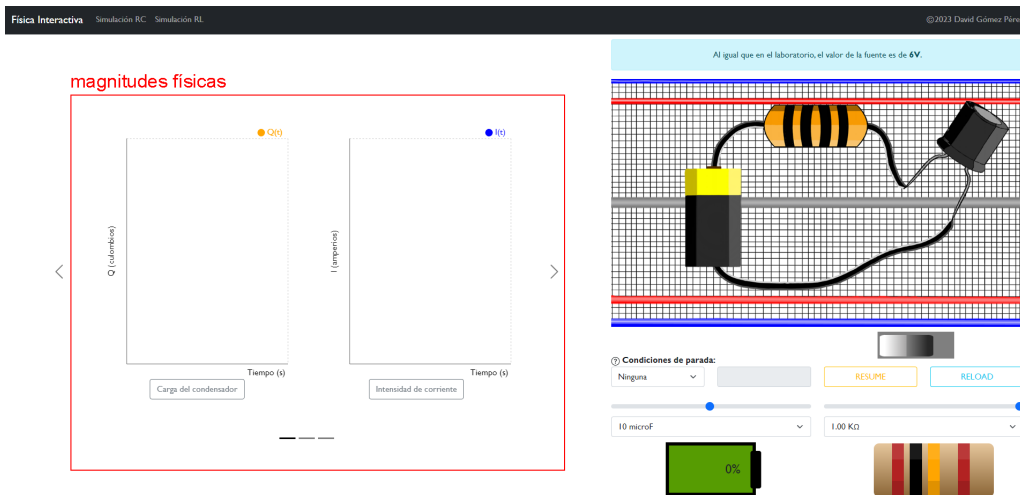


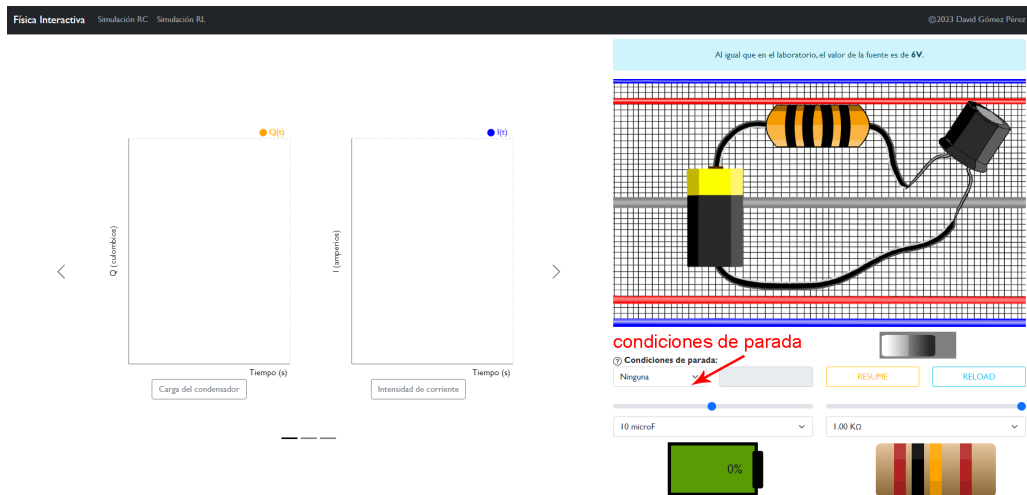
Figura 3: Interfaz de usuario de la aplicación (III).

Justo debajo, se dispone de dos botones cuyo objetivo es controlar el estado de la simulación. El botón de la derecha se encarga de reiniciar la simulación (*reload*) y el de la izquierda parar o reanudar esta simulación, dependiendo del estado actual de la misma (figura 4a y 4b).

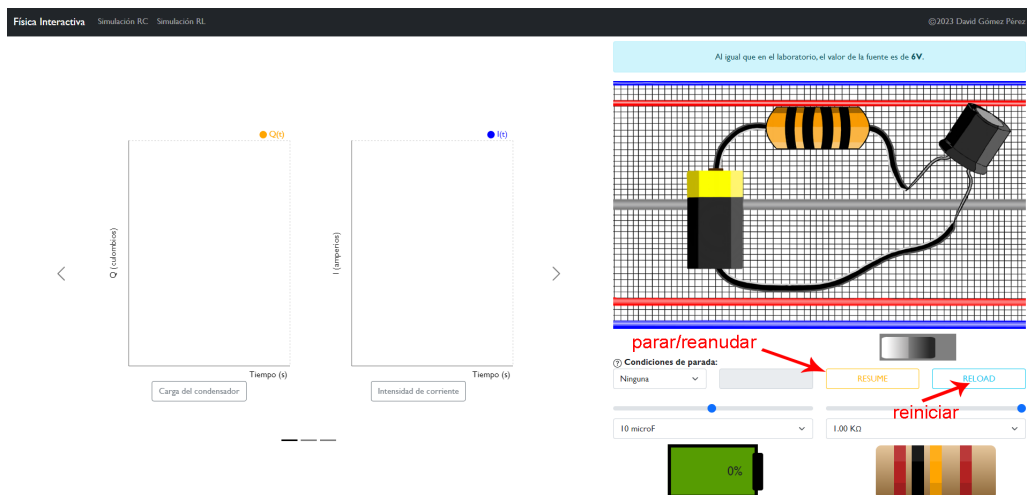
En la parte inferior tenemos los controladores que permiten cambiar los componentes del circuito. Estos se dividen en dos partes. La primera se trata de un *slider* para proporcionar el valor numérico de los mismos mientras que la segunda, situada justo debajo, es un *dropdown* utilizado para cambiar el multiplicador (por ejemplo, para utilizar una resistencia en megaohmios).

Por último nos quedan los apartados teóricos con información acerca del circuito. Estos se encuentran ocultos, y para activarlos, debemos de pulsar sobre la magnitud física de la que se quiera conocer más. La información disponible es la siguiente:

- Para el caso del circuito RC, tenemos: *carga del condensador, intensidad de corriente, diferencia de potencial en la resistencia, diferencia de potencial en el condensador, energía almacenada, capacidad del condensador, ley de Ohm y fuerza electromotriz (F.E.M)*. Estas tres últimas se muestran haciendo click en los elementos situados bajo los controladores de los componentes.



(a)



(b)

Figura 4: Interfaz de usuario de la aplicación (IV).

- En el caso del circuito RL, tenemos: *intensidad de corriente, diferencia de potencial en el inductor, diferencia de potencial en la resistencia, energía almacenada, flujo magnético, fenómeno de autoinducción, ley de Ohm y fuerza electromotriz (F.E.M).* Al igual que en el caso anterior, estas se muestran tras pulsar sobre cada uno de los elementos situados debajo de los controladores del circuito.

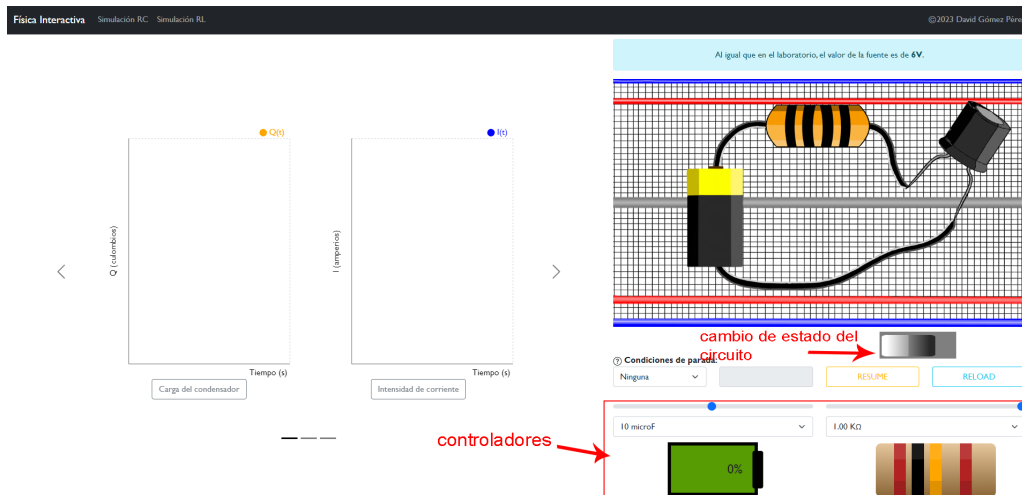


Figura 5: Interfaz de usuario de la aplicación (V).

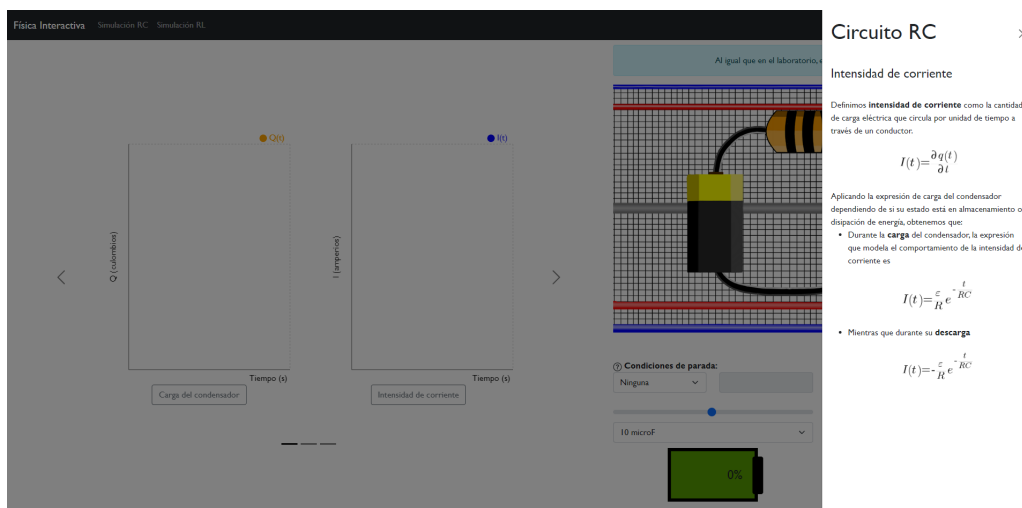


Figura 6: Interfaz de usuario de la aplicación (VI).

### 0.3. Manual de usuario del script *python* utilizado para las pruebas

Para utilizar el *script* debemos de abrir el directorio *simulation-tests* situado en la carpeta raíz del proyecto. El primer paso a realizar es instalar las dependencias guardadas en el fichero *requirements.txt*:

```
$ pip install -r requirements.txt
```

A continuación, podemos utilizar el script de la siguiente manera:

```
$ python3 main.py [arguments]
```

, dónde los argumentos son los que siguen:

COMANDO	ABREV.	DESCRIPCIÓN
-time	-t	Establece la duración de la simulación.
-simulationType	-st	Tipo de simulación. Puede tomar los valores “RC” o “RL”. Su definición es obligatoria.
-incrementValue	-inc	Escala de tiempo para el cálculo de los resultados. Por defecto su valor es de 0.001 segundos.
-capacitor	-c	Valor en Faradios de la capacidad del condensador. Solo es utilizado cuando el tipo del circuito a simular es RC. Por defecto su valor es de 0.005F.
-inductor	-i	Valor en Henrios de la inductancia de la bobina. Solo es utilizado cuando el tipo del circuito a simular es RL. Por defecto su valor es de 10H.
-resistor	-r	Valor en Ohmios de la resistencia. Por defecto su valor es de $3\Omega$ .
-voltage	-v	Valor en Voltios de la fuente de alimentación. Por defecto su valor es de 5V.
-conditionVal	-condV	Establece la condición de parada de la simulación. Si el circuito a simular es del tipo RC, este valor hará referencia a la carga del condensador. En caso del circuito RL a la intensidad de corriente. *
-conditionPer	-condP	Establece la condición de parada de la simulación. Si el circuito a simular es del tipo RC, este valor hará referencia al porcentaje de carga del condensador (0-100). En caso del circuito RL, a la intensidad de corriente del mismo. *

Tabla 1: Argumentos válidos del *script* de python.

\*En caso de que el valor de la carga o intensidad supere su valor máximo permitido (o en caso de indicar el porcentaje, este sea menor que cero o mayor que cien), se mostrará un mensaje de error indicando la carga o intensidad de corriente máximas permitidas para el

circuito a simular.