

## Desafío - Obtener información y repositorios Github de usuarios

- Para realizar este desafío debes haber estudiado previamente todo el material disponibilizado correspondiente a la unidad.
- Una vez terminado el desafío, comprime la carpeta que contiene el desarrollo de los requerimientos solicitados y sube el .zip en el LMS.
- Desarrollo desafío:
  - El desafío se debe desarrollar de manera Grupal
  - Para la realización del desafío necesitarás apoyarte del archivo Apoyo Desafío - Obtener información y repositorios Github de usuarios

### Capítulos

El desafío está basado en los siguientes capítulos de la lectura:

- Trabajar con APIs
- Manejo de errores

### Descripción

En el siguiente desafío se desea visualizar los datos y repositorios de un usuario en específico de Github. Para esto nos comunicaremos con la [API](#) pública de Github, específicamente con la dirección web de la API perteneciente al usuario: "<https://api.github.com/users/{user}>", a la cual se le debe pasar como parámetro el nombre de usuario **/user**, para traer la información del usuario indicado, como: Nombre de usuario, login, cantidad de repositorios, localidad, tipo de usuario y avatar. Así mismo, para traer los repositorios del usuario indicado se deben pasar otros dos parámetros, como el número de páginas que deseamos visualizar y la cantidad de repositorios que deseamos visualizar por página, siendo la dirección web de API para este caso: [https://api.github.com/users/{user}/repos?page={pagina}&per\\_page={cantidad\\_repos}](https://api.github.com/users/{user}/repos?page={pagina}&per_page={cantidad_repos}).

Para realizar la consulta por usuario, en la carpeta "Apoyo Desafío - Obtener información y repositorios Github de usuarios" encontrarás todos los archivos necesarios para elaborar el desafío, como el caso del archivo HTML con el formulario ya elaborado, en donde, se muestran los tres campos de entrada o parámetros que se necesitan para conectar con la API mediante la sentencia fetch:

## Busqueda de Usuarios en GitHub y sus Repositorios

Nombre de Usuario

Escriba el nombre del usuario de GitHub que desea buscar.

Número de Página

Repositorios por Páginas

Enviar

### Resultados

Imagen 1. Formulario de búsqueda.

Al obtener la información de la API mediante las dos consultadas realizadas, se deben mostrar los resultados exactamente abajo del titulo "Resultados" en una columna se debe mostrar toda la información del usuario y en otra columna la cantidad de repositorios que se deseen visualizar, cada repositorio tendrá el nombre y enlace correspondiente que al hacer click, deberá llevarnos al sitio del repositorio. Como se muestra a continuación en la siguiente imagen:

### Resultados

Datos de Usuario	Nombre de repositorios
	<a href="#">-TallerRubyHashes</a>
Nombre de usuario: DesafioLatam	<a href="#">act01-fds</a>
Nombre de login: DesafioLatam	<a href="#">ACT01_crear_temas_WP</a>
Cantidad de Repositorios: 212	<a href="#">ACT01_intro-wordpress_WP</a>
Localidad: Santiago, Chile y Monterrey, México	<a href="#">ACT01_optimizando_WP</a>
Tipo de usuario: Organization	<a href="#">ACT01_personalizando_temas_WP</a>
	<a href="#">ACT01_seo-deploy-seguridad_WP</a>
	<a href="#">act02-fds</a>
	<a href="#">ACT02_crear_temas_WP</a>
	<a href="#">ACT02_intro-wordpress_WP</a>

Imagen 2. Resultado de la búsqueda.

Es importante destacar que la API soporta listar hasta 100 repositorios por vez, por lo que hay que enviar los datos de paginación para poder visualizar todos los repositorios.

## Requerimientos

1. Crear tres funciones, una `request`, otra `getUser` y por último una función `getRepo`, todas deben implementar `async..await`. La función `request` hará las peticiones a la API y retorna el resultado, mientras que las funciones `getUser` y `getRepo` enviarán los datos a la función `request` para obtener la información del usuario y los repositorios a mostrar. Utiliza una URL base con el valor: <https://api.github.com/users>.
2. Agregar una escucha (`addEventListener`) al formulario, que permita activar una función en donde se capturen los datos ingresados por el usuario de la página (nombre de usuario, número de página, repositorio por páginas).
3. Mediante la implementación de una Promesa, realizar el llamado a las dos funciones al mismo tiempo que permiten conectarse con la API y traer la información en el caso de existir `"getUser"` y `"getRepo"`. Pasando como parámetros los valores necesarios para cada llamado de la API según la URL.
4. Mostrar los resultados obtenidos de la API en el documento HTML en la sección de "Resultados", como se muestra en la figura número dos.
5. En el caso que el mensaje retornado por la API sea "Not Found", indicar mediante una ventana emergente que el usuario no existe y no mostrar ningún tipo de información en la sección de resultado en el documento HTML.