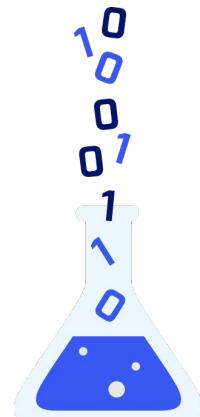


# ¿Qué son los desafíos?

- Son **ejercicios complementarios** a los realizados en clase.
- Permiten practicar y **consolidar los conocimientos** adquiridos.
- Es recomendable que los realices antes de rendir el examen de la clase.



# Ejercicio 1: Registro de información de contacto del usuario

Además de la búsqueda por nombre de producto que se realizó en el desafío del módulo 1, se agregará **una nueva funcionalidad** al sistema de LimpiezaIT.

La empresa desea mejorar el registro de todas las ventas añadiendo la **información de contacto del usuario**.

La organización advirtió que, en sus registros, **faltan los campos de teléfono y celular** del usuario.



## Objetivo:

- Investigar cómo obtener los datos de teléfono y celular del usuario, desde la API de [Randomuser.me](#), y añadirlos a los datos que ya se están obteniendo.



## Instrucciones:

1. **Acceder a la API:** ingresar a la [URL de Random User Generator](#).
2. **Investigar la documentación:** explorar la documentación para identificar dónde se encuentran los campos de teléfono (phone) y celular (cell).
3. **Modificar la lógica de obtención de datos:** ajustar el código para incluir los campos de teléfono y celular en los datos que se obtienen de la API.

## Pasos detallados:

### 1. Explorar la API:

- a. Abrir la [URL](#) en el navegador.
- b. Usar la documentación o la respuesta de ejemplo proporcionada para encontrar los campos phone y cell.

### 2. Actualizar el código:

- a. Si ya se cuenta con un método o función que obtiene los datos del usuario desde la API, modificarlo para incluir los campos phone y cell.
- b. Asegurarse de que estos nuevos datos se almacenen correctamente en el registro de ventas.



## Ejercicio 2: Implementación de seguridad básica en la API de LimpiezalT

Se agregará **seguridad básica** a la API REST que se desarrolló para LimpiezalT, en la etapa 1 del proyecto integrador.

Se implementará mediante **Spring Security** para proteger los *endpoints* de la API.



## Objetivo

- **Agregar seguridad básica:** implementar autenticación básica en la API REST.
- **Probar la seguridad:** utilizar Postman para probar los endpoints asegurados.



## Tarea

1. **Configurar el proyecto:** configurar la seguridad básica en el proyecto para proteger los endpoints de la API.
2. **Definir credenciales:** definir un usuario y una contraseña para acceder a la API.
3. **Probar con Postman:** configurar Postman para enviar las credenciales al realizar las solicitudes a los endpoints de la API.



## Pasos a seguir

1. **Agregar dependencia Spring Security:** añadir la siguiente dependencia en el archivo pom.xml del proyecto.
2. **Configurar seguridad en Spring Boot:** crear una clase de configuración para Spring Security en tu proyecto.
3. **Definir credenciales:** usuarios/claves.
4. **Probar la API con Postman**
  - a) Abrir Postman y crear una nueva colección de solicitudes.
  - b) Configurar las solicitudes para usar autenticación básica. En la sección "Authorization" de cada solicitud, seleccionar "Basic Auth" e ingresar el nombre de usuario y la contraseña definidos en el paso anterior.

## Ejemplo de prueba en Postman

1. **Abrir Postman:** iniciar Postman y abrir una nueva solicitud.
2. **Configurar URL:** ingresar la URL del endpoint de tu API, por ejemplo,  
<http://localhost:8080/api/productos>.
3. **Añadir autenticación:**
  - a. Ir a la pestaña "Authorization".
  - b. Seleccionar "Basic Auth".
  - c. Ingresar el nombre de usuario y la contraseña que se configuró (por ejemplo, usuario y contraseña).
4. **Enviar solicitud:** enviar la solicitud y verificar que se obtiene una respuesta correcta solo si se proporcionan las credenciales correctas.

A continuación encontrarás su resolución para que verifiques cómo te fue.



# ¡Terminaste el módulo!

## Todo listo para rendir el examen