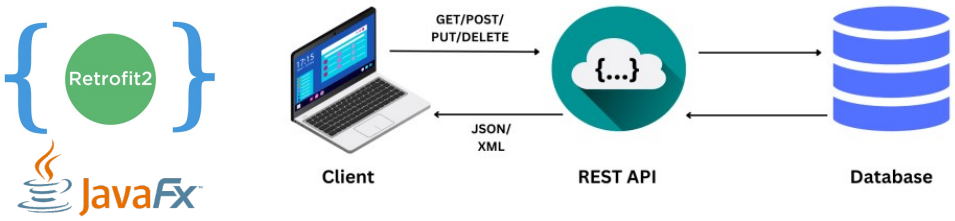
1 TEMA 4 – EXPLOTACIÓN APIS DESDE JAVAFX - RETROFIT



PARTE I : EXPLOTACIÓN APIREST MEDIANTE JAVAFX Y RETROFIT

1. Hacer una copia del proyecto entregado para DI en el primer trimestre.
2. Desplegar la API del primer trimestre en la misma máquina donde tenemos la BBDD JDBC, esto es, si tenemos un DOCKER LAMP donde accedemos a JDBC, hacemos que la API también esté alojada en dicha máquina y ataque a la misma BBDD pero vía API. **Ej:** nuestra API atacaba **PRODUCTOS** en la relación **PRODUCTOS**-VENTAS-CLIENTES.
3. Ahora deberíamos deshabilitar la conexión mediante JDBC para la tabla escogida (comentando el código). Es decir, en cada punto donde se hace CRUD, sustituir el código SQL por una llamada RETROFIT que llame a cada una de las páginas de la API. Por ejemplo, donde hacíamos "INSERT INTO…." ahora llamamos al método Insertar del servicio *MIAPIServicioInterfazInsertar* y le pasamos un objeto de la clase **PRODUCTOS** para poder insertar en la base de datos vía la API. **Ej:** nuestra API atacará **PRODUCTOS** vía APIREST y las otras dos tablas mediante JDBC.
4. Los cambios deberían ser mínimos ya que tanto para JDBC como para RETROFIT se trabaja con una clase modelo que debería ser si no la misma, muy parecida a la que ya hemos utilizado en DI.

PARTE II : EXPLOTACIÓN API PÚBLICA

5. Escoger una API pública, revisar documentación y seleccionar una tabla (STAPI y SWAPI se pueden escoger pero NO las tablas EPISODES ni PEOPLE respectivamente y **NO se puede repetir API**). También se tendrá en cuenta si escogemos una API con API Key.
6. Crea un proyecto para atacar la API (lógicamente solo en modo Consulta), y genera una interfaz gráfica donde se demuestre que se pueden consultar los distintos ítems. **Por ejemplo**: Escogemos SWAPI y la tabla People, generamos una GUI en JAVAFX donde al cambiar el personaje (utilizando un COMBO) podemos ver una serie de datos asociados (no es obligatorio cargarlos todos!). Es decir, al escoger el ID 3, estamos haciendo esto: https://swapi.py4e.com/api/people/3/, y obteniendo esto:


```
{
        "name": "R2-D2",
        "height": "96",
        "mass": "32",
        "hair_color": "n/a",
        "skin_color": "white, blue",
        "eye_color": "red",
        "birth_year": "33BBY",
        "gender": "n/a",
        "homeworld": "https://swapi.py4e.com/api/planets/8/",
        "films": [
                "https://swapi.py4e.com/api/films/1/",
                "https://swapi.py4e.com/api/films/2/",
```

1.1 NORMAS DE ENTREGA, EVALUACIÓN Y RÚBRICA DE CORRECCIÓN

Muy importante: La práctica será calificada si y solo sí se ha defendido con éxito de forma parcial o en la entrega final, en cuyo caso la nota final se calculará siguiendo la siguiente rúbrica de corrección.

APARTADOS DE LA PRÁCTICA	PESO(EN %)
Defender la **PARTE I**:, no es necesario ni entregar código ni entregar documentación. (Como mínimo se tiene que tener esta parte superada para aprobar HLC este trimestre).	50%
PARTE II: a) Devuelve correctamente los datos de la API a partir de los datos de entrada (TextField, Combo, etc). Se pueden mostrar por terminal para empezar.	10%
PARTE II: b) Crea la interfaz gráfica de la API Pública mostrando distintos valores de variado tipo (los tipos determinarán el control JAVAFX).	10%
PARTE II: c) El diseño de la interfaz sigue las normas de Usabilidad, tiene validación (en el campo de consulta), usa animaciones, CSS, etc…	20%
PARTE II: d) Utiliza una API distinta de SWAPI/STAPI	7,5%
PARTE II: e) Utiliza una API con API KEY	2,5%

IMPORTANTE: Si se entrega la **PARTE II** habrá que enviar el proyecto comprimido en .rar o en .zip.