Introducción

El siguiente documento expone el enunciado y pautas a seguir para la realización de una práctica que repasa los conocimientos teóricos necesarios impartidos en el curso **Plan de formación Apificación y Arquitecturas Backend JAVA** para comenzar a trabajar en proyectos bajo el marco tecnológico de una **Metodología API-First de Axpe Consulting**.

Alcance

El ejercicio aborda la realización de una aplicación dentro del marco tecnológico de **Metodología API**, para el diseño, definición, analisis e implementación de especificación API con las siguientes características:

- Metodología Api First.
- Diseño bajo el estandar API Rest.
- Recursos de APIs CRUD.
- Utilización de herramientas básicas de desarrollo: validación de datos, validación de la especificación en tiempo de diseño, desarrollo e ejecución.
- Provisión de algún tipo de seguridad básica.
- Diseño de plan de pruebas.
- Generación de entorno de pruebas.
- Generación de colección de pruebas.

Todo ello, siguiente en todo momento la guía de **Diseño API** marcada por la compañía.

Práctica

La práctica se divide en varios apartados, cada uno de los cuales deberán de ser realizados y probados por el alumno antes de pasar al siguiente apartado (Salvo en aquellos casos donde se diga lo contrario explícitamente o se propongan ejercicios opcionales de mejora).

Resolución de dudas

Se proporcionaran los siguiente canales para la realización y resolución de dudas: (**TODO: Incluir referencia**)

Configuración del entorno de trabajo

Para poder realizar esta práctica se debera tener configurado el entorno de trabajo, para ello indicamos las herramientas mínimas necesarias para poder llegar a alcanzar con exito la práctica:

- Insomnia
- Editor de swagger
- Stoplight
- Postman
- Spectral
- Prims

- GetSandbox
- Apimatic
- Visual Studio Code
- 42Crunch

Enunciado de la práctica

Se ha solicitado la creación de un producto para la gestión de los empleados de una compañía. Esto significa que partiremos de un modelo y conjunto de datos existentes en el sistema final. Definiremos una especificación API, enfocado en una **estrategía API-First**, que apantalle los servicios del sistema que se desea consumir. Para ello indicaremos una serie de pautas a seguir mínimas para completar con éxito la práctica final.

Primer escenario

Para este primer caso se necesita analizar e identificar los recursos que debe exponer nuestra API. Para ello deberán extraer las siguientes funcionalidades que se requieren del siguiente enunciado:

La empresa Axpe Consulting necesita "Apificar" el producto que se encarga de gestionar los empleados de la compañia. Para ello ha decido subcontratar a una empresa externa que se encargue de actualizar dicha parte. Para una primera versión de entrega la empresa Axpe ha decido que debe de incorporar las siguientes funcionalidades de negocio:

- La posibilidad de recuperar todos los empleados que hay en la actualidad en la compañía.
- Incorporación de un nuevo empleado.
- Recuperación de los datos un empleado.
- Modificación total o parciales de los datos de un empleado.
- Baja de un empleado en los sistemas.

Para ello, se analizará que es un empleado para la organización, que requisitos funcionales quieren exponer, teniendo con objetivo crear la funcionalidad de negocio correspondiente. Para ello identificaremos los recursos y campos que se necesitan, y así generar una interfaz API que se adecue a la necesidades del negocio de la organización.

* Se valorara positivamente que se añada la maxima funcionalidad para completar el caso de uso (codigo HTTP, filtros, paginación, cabeceras, HATEOAS, etc).

Segundo escenario

Despues de una segunda iteración con el equipo de negocio de la compañia, se ha detectado que se requiere una nueva funcionalidad a añadir. Según lo explicado en la reunión, se debe analizar e implementar esta nueva funcionalidad de negocio que sea capaz de validar el correo electrónico de un empleado. Para ello analizaremos que es lo que requieren y definiremos el nuevo recurso y a posterior desarrollaremos su implementación.

Por último, nos dijeron que esta validación del correo será contra un sistema fuera de la compañía y nos dejaron las siguiente documentación https://rapidapi.com/adminMelissa/api/global-email-v4/] (https://rapidapi.com/adminMelissa/api/global-email-v4.

* Se valorara positivamente que se añada la maxima funcionalidad para completar el caso de uso (codigo HTTP, filtros, paginación, cabeceras, HATEOS, etc).

Api First: ¿Qué debemos hacer?

Un **Diseño API** según la métodología de desarrollo marcada por la compañía a la hora de abordar un desarrollo de APIs enfocado a una estategía Api-First.

Como ya hemos visto en la guía y/o sesiones formativas previas, **La Metodología** es la base de cualquier estrategia API First, que contiene los procedimientos a seguir para asegurar el cumplimiento de la misma. La definición de la metodología consta de una serie de procedimientos necesarios segun los criterios definido en la **Guia de Diseño Axpe**, entre otros, apoyadose en el estándar Open Api, que se base principalmente: Qué operaciones se van a exponer, esquema de datos a utilizar en las peticiones y respuestas, casos de error, seguridad del api, etc.

El *primer* paso, analizar e identicar los recursos (endpoints) que deben exponer la funcionalidad exigida por el área de negocio.

Por ejemplo, devolver una lista de usuarios:

> GET /users

Segundo paso, identifcar los parámetros de entrada (headers/request) y salida (headers/response), por cada operación identificada en el primer paso.

```
{
  "data": [
    {
      "id": "1",
      "username": "Josemi",
      "firstName": "Jose Miguel",
      "lastName": "Ruiz",
      "lastName2": "Ruiz",
      "identification": {
        "value": "70888999A",
        "type": "NIF"
      },
      "email": "john@email.com",
      "phone": "string",
      "userStatus": "PENDING"
    }
  ],
  "pagination": {
    "offset": 10,
    "limit": 10,
    "pageNumber": 0,
    "totalPages": 0,
    "totalElements": 120,
    "links": {
      "self": "https://api.axpe.com/users/v1/users?limit=10&offset=10",
      "first": "https://api.axpe.com/users/v1/users?limit=10&offset=0",
```

```
      "last": "https://api.axpe.com/users/v1/users?limit=10&offset=120",
      "prev": "https://api.axpe.com/users/v1/users?limit=10&offset=10",
      "next": "https://api.axpe.com/users/v1/users?limit=10&offset=10"
    }
  }
}
```

* Se valora positivamente que se delimiten los campos (atributos) según su funcionalidad.
Ejemplos (formato yaml):

```
phone:
   type: string
   pattern: '\+[0-9]{1,3}-[0-9()+\-]{1,30}'
```

```
date:
   type: string
   format: date
```

```
 userStatus:
   type: string
   description: User Status
   default: PENDING
   enum:
   - ACTIVE
   - PENDING
   - REMOVED
```

Herramientas de validacion

Además de realizar un diseño API, se requiere que se utilice alguna herramienta de validación como Spectral. Para ello deberemos utilizar el editor de Stoplight + spectral o ejecutar mediante comando.

Requisitos:

- Validar el diseño implementado aplicando las reglas predefinidas por Spectral.
- Sobreescribir una regla predefinida, por ejemplo: info-contact que se encarga de validar que se esta definiendo el propietario de la especificación.
- Crear una nueva regla a elección.

Entorno de pruebas

Generar un entorno de pruebas sandbox (Prims) y adaptarlo a las pruebas de cada fase (mock/ejecución real).

Pruebas de rendimientos. Ejemplo cargar 1000 usuarios en la base. ¿Cómo lo haría?

Colección de pruebas

Seguridad

El Servicio será usado desde la intranet de la compañía. Se requiere de incluir algún mínimo de seguridad en el Servicio, para lo cual se propone utilizar el protocolo de Autenticación Básica, aunque el responsable del área está abierto a otros tipos de seguridad más avanzados, siempre que esté dentro de los plazos de entrega.

42Crunch Opcionalmente, a mayores se podrá realizar una validación a nivel de seguridad. PAra ello se podra utilizar la herramienta de 42Crunch online o descargar la extensión disponible para visual studio.

Entregables

- Definición de las API (yaml) utilizando cualquier editor de diseño API (Stopligh, Editor swagger o Insonnia).
- Reglas de validación en spectral.
- Coleccion de pruebas en postman.