Universidad Francisco de Vitoria UFV Madrid

NOMBRE: _____

GRUPO: _____ FECHA: _____

CALIFICACIÓN:

EXAMEN FINAL CONVOCATORIA ORDINARIA – PRÁCTICA (60%)
DESARROLLO E INTEGRACIÓN DEL SOFTWARE

Instrucciones
El razonamiento hasta las soluciones tiene la misma importancia que la propia solución. Se valorará positivamente un razonamiento adecuado. Lee detenidamente todo el ejercicio antes de empezar y con ello elegir el orden en que debes contestar cada apartado.
La duración de esta parte de la prueba es de 120 minutos para el desarrollo. Y 10 minutos para la entrega.

Requisitos y pautas generales de trabajo

1. Acceder a la URL de Github Classroom.
    1. https://classroom.github.com/a/vmnAm477
2. Seguir el procedimiento de Github para la creación del nuevo repositorio.
3. Clonar el repositorio a un directorio local. El repositorio dispone de ficheros necesarios para la realización del ejercicio.
4. Trabajar en el enunciado del ejercicio en el repositorio en local.
5. **Es obligatorio realizar al menos un commit cada 15 minutos.**
6. Una vez dado por concluido el ejercicio, se deberá:
    1. **Realizar un commit final y generar una release con el nombre v1.0**.
    2. Comprimir la carpeta de trabajo y subirla a la tarea habilitada en el aula virtual para tal fin. Para evitar problemas de tamaño de carpeta, eliminar las carpetas node_modules y target.

Universidad Francisco de Vitoria UFV Madrid

NOMBRE: _____

GRUPO: _____ FECHA: _____

CALIFICACIÓN:

Enunciado

1/3 Desarrollo de la aplicación

Se solicita crear una pequeña aplicación con el framework Spring la cuál consiste en una API REST que consuma datos de una API ya existente y los vuelque en un fichero local en formato JSON

No es necesario desarrollar un frontal, ya que los datos se enviarán usando un cliente tipo *Postman* o equivalente.

1. La API REST se deberá desarrollar utilizando el framework Spring y SpringBoot.
2. Generar un proyecto Maven usando Spring Initializer, usando Java 11 como lenguaje de programación y el plugin Spring Web.
3. Definir el Group ID del proyecto a "ufv.dis.final2022" y el Artifact ID con las iniciales del nombre del alumno.
4. Para trabajar con el fichero JSON se debe hacer uso de la siguiente librería:
   groupId: com.google.code.gson
   artifactId: gson
   version: 2.6.2

La API de origen de los datos será la siguiente: https://swapi.dev/.

Desde un cliente tipo Postman o Thunder Client (Visual Studio Code) se realizarán peticiones a la aplicación desarrollada. La API deberá contener un método POST y un método GET con las siguientes características:

Método 1/2: POST

- Se le solicitará la descarga de la información de la API de Swap. Esta información podrá ser de tipo "people" o "starship".
- Se facilitan en el repositorio las clases donde almacenar la información recibida.
- Deberá recoger la información de Swapi, y almacenarla en el listado correspondiente de la aplicación a desarrollar.
- Además, deberá devolver la información solicitada, junto con un estado de OK

Un ejemplo del body de la solicitud es el siguiente:

```
{
"entity": "people",
"id": 3
}
```

Que deberá realizar la llamada a Swapi con el formato
https://swapi.dev/api/people?id=3

Método 2/2: GET

Devolverá la lista de todas las peticiones realizadas con anterioridad, y que deberán estar almacenadas en un fichero JSON en disco.

Podrá devolver un elemento concreto del JSON (por orden incremental), o la lista completa de elementos.

2/3 Test Driven Design (JUnit)

Deberán generarse al menos las siguientes pruebas unitarias con la librería indicada:

- El elemento solicitado se ha incluido correctamente en el listado correspondiente.
- El objeto almacenado en la lista tiene tantas propiedades como el objeto de origen.

Dependencia requerida

groupId: junit
artifactId: junit
version: 4.1

3/3 Despliegue en Heroku

La aplicación deberá ser desplegada en Heroku.

Debéis crear un fichero heroku.doc donde debes incluir todas las evidencias del proceso de despliegue y las explicaciones o aclaraciones que consideres necesarias sobre esta prueba. Texto, pantallazos, etc.… **La URL de Heroku ha de incluirse en este documento**. Este fichero debes subirlo a la plataforma junto con el resto de entregables.

Importante: Si no aparece en el documento mencionado la URL de la aplicación desplegada, este apartado se considerará suspenso.

Universidad Francisco de Vitoria UFV Madrid

NOMBRE: _____

GRUPO: _____ FECHA: _____

CALIFICACIÓN:

Entregables

1. Ficheros del proyecto y la carpeta. git.
    1. Eliminar la carpeta target y la carpeta node_modules.
    2. Incluir un fichero txt con la url del repositorio de Github
2. Código del proyecto alojado en el repositorio de Github, incluidos los casos de prueba. (debe coincidir con la última versión de Github)
3. Fichero Heroku.doc con la URL donde esté desplegada la aplicación en Heroku.
4. Un fichero "referencias.txt" que contenga el origen/enlaces de todas las referencias utilizadas durante el desarrollo del examen.

No olvidar generar una release con el contenido del proyecto.
Comprime todos los estos ficheros en un archivo zip: *TuApellidoNombre.zip* que es el que deberás subir finalmente a la plataforma.

Calificación del ejercicio

Apartado	Gestión de Git	Framework y lenguajes	Testing	Heroku
Puntuación	1,5	6	1,5	1