

EXAMEN FINAL – 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 **130 minutos**. Y **10 minutos para la entrega**.*

Requisitos y pautas generales de trabajo

1. Acceder a la URL de Github Classroom.
<https://classroom.github.com/a/BZ5ynVNM>
2. Seguir el procedimiento de Github para la creación del nuevo repositorio.
3. Clonar el repositorio a un directorio local.
4. Se usará una único repositorio para ambos proyectos, el de front y el de back.
5. Trabajar en el enunciado del ejercicio en el repositorio en local.
6. **Es obligatorio realizar al menos un commit cada 15 minutos.**
7. 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.

Enunciado

Se pide desarrollar una aplicación que emule el comportamiento de la red social Twitter. Debe permitir la creación de mensajes y la eliminación de estos, así como listar los tweets que se hayan dado de alta.

El formato de salida del listado de los tweets será igual al que se indica en el siguiente ejemplo:

```
Most people do not listen with the intent to understand; they listen with the intent to reply.
-- Alfonso. moments ago
If you're not getting measurable results, there's a strong chance you're wasting your life, one hour at a time.
-- Alfonso. 2 minutes ago
There are four real ways to spend your time: thought, conversation, actions and distractions. Choose wisely!
-- Alfonso. 4 days ago
The person who says he knows what he thinks but cannot express it usually does not know what he thinks.
-- Alfonso. 9 years ago
```

Desarrollo de la aplicación

Creación de la interfaz

La página contará con dos pestañas.

- Pestaña **Alta**: Contendrá el formulario de alta de un tweet.
- Pestaña **Listado**: Mostrará todos los tweets que se hayan dado de alta en el sistema. El formato será el indicado en el ejemplo anterior y **tendrá también un botón que permita borrar el tweet**.
 - Al pulsar en este botón (el de borrado), se borrará dicho tweet.
- Los datos que se deben incluir en el formulario son:
 - Nombre de usuario.
 - Mensaje del Tweet.
 - Fecha, en formato DD-MM-YYYY. Ej: 10-10-2020.
- Estos datos servirán tanto para el alta como para la visualización de un tweet en la pestaña Listado.

Tips para el desarrollo:

1. Se realizará un proyecto único para la interfaz, usando el arquetipo Vaadin.
2. La versión de Java a utilizar es **Java 11**.
3. Utilización del framework **Vaadin 14.7.3**
4. Todas las dependencias deberán ser gestionadas por **Maven**.
5. Archetype de Vaadin para empezar la aplicación
 - a. GroupID: com.vaadin
 - b. Artifact ID: vaadin-archetype-application
 - c. Version: 14.7.3
6. Definir el Group ID del proyecto a "ufv.extraordinaria.dis" y el Artifact ID como las iniciales del alumno en minúsculas.
7. Se valorará positivamente un diseño de la UI adecuado.

API REST

Se realizará un proyecto Springboot usando la herramienta alojada en el enlace <https://start.spring.io/>. Los datos se leerán y enviarán mediante el desarrollo de una API REST que permita, mediante el método GET, obtener la lista de los tweets a mostrar en la pantalla inicial y mediante los correspondientes métodos DELETE y POST, eliminar un tweet o dar de alta uno nuevo.



Los tweets deberán estar en un fichero JSON donde se almacenarán y leerán posteriormente. Este fichero está disponible en el aula virtual.

Método 1/3: GET

Este método devolverá al front un array de objetos JSON con todos los tweets que están almacenados en el fichero JSON. Estos se mostrarán a continuación en la vista con el formato indicado anteriormente.

El formato del objeto JSON de cada tweet será el siguiente:

```
{
  "id": 1,
  "usuario": "Usuario",
  "tweet": "Mi primer tweet en la nueva app",
  "fecha": "20-02-2022"
},
```

Método 2/3: POST

Creación de un nuevo tweet. Este método recibirá un objeto JSON con los datos necesarios para dar de alta un nuevo tweet en el fichero JSON que usamos como base de datos. En este caso, el id no se enviará desde el front y se generará en la parte de la API.

```
{
  "usuario": "Usuario",
  "tweet": "Mi primer tweet en la nueva app",
  "fecha": "20-02-2022"
},
```

Método 3/3: DELETE.

Eliminación de un tweet: Este método recibirá el id del tweet en la petición DELETE y lo eliminará del fichero.

```
{
  "id": 1
},
```

Gestión del fichero JSON

Los tweets serán almacenados en un fichero JSON. Para la gestión de este fichero, se indican a continuación algunos tips para facilitar su gestión:

Tips para el desarrollo

- Clase Tweet
 - Contiene el nombre, mensaje, fecha de cada mensaje y el id del mismo.

- El id es autogenerado, es decir, los usuarios no lo tienen que gestionar desde el front a la hora de crearlo.
- Genera un método que devuelve la fecha con el formato solicitado, apoyándose en la librería **prettytime**, referenciada más adelante.
- Dependencias adicionales (GSON).
 - GroupID: com.google.code.gson
 - Artifact ID: gson
 - Version: 2.6.2
- Dependencias adicionales (Prettytime, para la gestión de las fechas por pantalla).
 - GroupID: org.ocpsoft.prettytime
 - Artifact ID: prettytime
 - Version: 4.0.0.Final
- Referencia: <https://www.ocpsoft.org/prettytime/>
- Clase Tweets
 - Contiene un ArrayList o Lista de objetos de la clase Tweet, y se encarga de gestionarlo. Contendría los métodos add(), delete() y get(). Se pueden añadir los métodos auxiliares que se consideren oportunos.
- Las fechas pueden almacenarse en un objeto "Date". new Date(). Puede pasarse como parámetro la fecha en milisegundos en caso de necesitar una diferente a la actual. new Date(System.currentTimeMillis() + 1000*60*10).

Entregables

- Ficheros de los dos proyectos y la carpeta. git. Eliminar la carpeta target y la carpeta node_modules.
- Código de los dos proyectos alojado en el repositorio de Github.
- Fichero referencias.txt en el que debes indicar la url de todas las referencias que hayas usado en el proyecto. No las que has consultado, sólo las que has usado, incluyendo el código de tus propias prácticas.

No olvidar generar una release con el contenido del proyecto y subirla al repositorio remoto clonado al comienzo del examen.

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	Vaadin/Http Request	API	Gestión del JSON
Puntuación	1,5	3	4	1,5