

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 **120 minutos**. 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/iHziJnIN>
2. Seguir el procedimiento de Github para la creación del nuevo repositorio.
3. Clonar el repositorio a un directorio local.
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.

Enunciado

Se pide desarrollar una aplicación que emule el comportamiento de la red social Twitter. Debe permitir la creación de mensajes y la edición de estos (funcionalidad no disponible en la red original), 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
```

Añadiendo un botón que permita la edición de este.

La página contará con dos pestañas.

- Alta/Edición: Contendrá el formulario de alta de un tweet.
- 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 editar el tweet.

Al pulsar en este botón, se verá la información del tweet en el formulario usado para dar de alta un mensaje, con la información ya rellena. La fecha se actualizará con la última edición.

Desarrollo de la aplicación

1. Aplicación Java utilizando el framework Vaadin.
2. La versión de Java a utilizar es Java 8.
3. Generar un proyecto Maven con el siguiente arquetipo:
 - a. Group ID: com.vaadin
 - b. Artifact ID: vaadin-archetype-application
 - c. Version: 8.6.2
4. Con la ayuda de una librería de terceros, mostrar por pantalla las fechas anteriormente generadas con el formato solicitado.
GroupID: org.ocpsoft.prettytime
Artifact ID: prettytime
Version: 4.0.0.Final
Referencia: <https://www.ocpsoft.org/prettytime/>
5. Definir el Group ID del proyecto a "ufv.dis2010.final" y el Artifact ID como las iniciales del alumno.
6. La interfaz deberá contener al menos el formulario de alta de mensajes y el listado de estos según la especificación anterior. Se valorará positivamente un diseño de la UI adecuado.
7. Los tweets deberán estar en un fichero JSON donde se almacenarán y leerán posteriormente.
8. Para trabajar con el fichero JSON se deberá hacer uso de la siguiente librería:
GroupId: com.google.gson

ArtifactId: gson

Version: 2.6.2

9. El formulario de creación de mensajes deberá contener los siguientes campos:
- Nombre (Autor)
 - Mensaje
 - Fecha. En formato (DD-MM-YYYY. Ej: 10-10-2022).
10. La aplicación deberá desplegarse en Heroku y ser accesible públicamente.

Tips para el desarrollo

- Clase Tweet
 - Contiene el nombre, mensaje y fecha de cada mensaje.
 - Genera un método que devuelve la fecha con el formato solicitado.
- Clase Tweets
 - Contiene un ArrayList de Tweet, y se encarga de gestionarlo. Contiene al menos los métodos print(), add() y get().
- 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).
- Para la conversión de fechas en formato Date de Java a String, se puede usar el comando SimpleDateFormat, en el que se indica el formato de String que tiene la fecha de origen.
- La librería permite configurar el idioma de las fechas. Mostrarlas en español.

Entregables

- Ficheros del proyecto y la carpeta .git. Eliminar la carpeta target y la carpeta node_modules.
- Código del proyecto alojado en el repositorio de Github, incluidos los casos de prueba.
- Fichero Heroku.doc con la URL donde esté desplegada la aplicación en Heroku.
- Fichero referencias.txt en la 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.

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	Heroku
Puntuación	1,5	7	1,5