16-6-2020

Luisa Cogollos Perucho y Rocío Sosa Ruiz

ufv, tecnologías avanzadas de programación

TAREAS APP

PRÁCTICA 2

Contenido

[INTRODUCCIÓN 2](#_Toc42512557)

[DESARROLLO 2](#_Toc42512558)

[GIT 2](#_Toc42512559)

[DIAGRAMA DE CLASES 2](#_Toc42512560)

[DIAGRAMA DE FLUJO 2](#_Toc42512561)

[DISEÑO DEL SOFTWARE DE LA APLICACIÓN 2](#_Toc42512562)

[INTERFAZ DE USUARIO 2](#_Toc42512563)

[BASE DE DATOS H2 2](#_Toc42512564)

[REPORTE DE SONARQUBE 2](#_Toc42512565)

[CONCLUSIÓN 2](#_Toc42512566)

# INTRODUCCIÓN

En este documento se muestran las explicaciones del desarrollo de la práctica 2 de la asignatura de Tecnologías Avanzadas de Programación.

El objetivo de dicha práctica es comprender y afianzar todos los conocimientos vistos en la asignatura, que nos permitirán estudiar las distintas estrategias avanzadas de programación orientadas al uso de marcos de trabajo y profundizando en la Arquitectura Orientada a Servicios, aprovechando para resolver problemas reales de manejo de información en la vida empresarial.

La práctica consiste en el desarrollo de una aplicación en la que se ponga de manifiesto el aprendizaje de los principios de diseño de software y de la programación orientada a objetos, así como las técnicas de refactoring y la arquitectura de una aplicación web cliente-servidor.

La aplicación de tareas que consistirá en un “GTD: Getting Things Done”, desarrollará un frontend y un backend independientes, cuya comunicación se realizara por HTTP. (Desarrollo de una API. JSON.)

Requisitos de Usuario:

* Se podrán crear tareas con una descripción
* Se podrá asignar prioridad a una tarea. (Default, low, medium, high)
* Se podrá asignar una fecha límite “deadline” a las tareas.
* Las tareas se podrán marcar como completadas
* Las tareas se podrán organizar en diferentes listas.

Requisitos aplicación:

* El lenguaje de programación a utilizar será Java.
* El frontend se realizará con el framework Java Vaadin
* El backend se realizará con el framework Java Spring.
* El backend dispondrá de una base de datos MySQL para persistir la información.
* El gestor de dependencias a utilizar en ambos proyectos será Maven.

# DESARROLLO

Este apartado consistirá en la explicación de como se ha ido desarrollando todo el proceso de la creación de la aplicación web de manera que quede claro todos los pasos realizados y el funcionamiento de la aplicación obtenida. Se han usado diferentes herramientas y métodos que se procederán a explicar a continuación.

## **GIT**

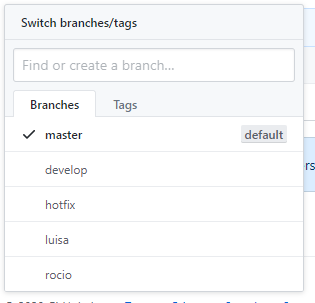
Durante todo el proceso del desarrollo de la práctica se ha hecho uso de la herramienta Git para facilitar el trabajo en equipo y la gestión de versiones del proyecto.

Se ha creado un repositorio privado (Practica-2-TAP-lr) en el que se ha dado acceso a los integrantes del grupo y al profesor mediante la herramienta Github.

<https://github.com/Rochsos/Practica-2-TAP-lr>

Para el buen uso de Git se ha usado la metodología Gitflow, que consiste en un flujo de trabajo que define un modelo estricto de ramificación proporcionando un marco sólido para gestionar proyectos grandes. Por lo que la estructura de ramificación se ha basado en el modelo de esta metodología haciendo uso de 5 ramas de trabajo que se procederán a explicar:

* **Máster:** es la rama principal y solo se subirán los cambios que estén preparados para subir a producción.
* **Develop:** se crea y se integra a partir de la rama Máster y estará el código que conformará la siguiente versión planificada del proyecto.
* **Hotfix:** se origina y se integra de Máster y se usará para corregir errores y bugs en el código en producción.
* **Luisa (Feature):** se origina e incorpora a Develop y se usa para desarrollar nuevas características de la aplicación y pertenece a la desarrolladora Luisa.
* **Rocío (Feature):** se origina e incorpora a Develop y se usa para desarrollar nuevas características de la aplicación y pertenece a la desarrolladora Rocío.



## **DIAGRAMA DE CLASES - LUISA**

## **DIAGRAMA DE FLUJO - LUISA**

## **DISEÑO DEL SOFTWARE DE LA APLICACIÓN - ROCIO Y LUISA**

En este apartado se procederá a explicar los principios o patrones usados referenciando a los conceptos adquiridos en la asignatura, de manera que expliquemos en que nos hemos apoyado o basado a la hora de desarrollar nuestro código.

Para empezar, al hablar de Programación Orientada a Objetos, es obvio referirnos a los cuatro pilares fundamentales que se cumplen y en los que nos hemos apoyado para el desarrollo correcto de nuestro código:

* Abstracción: los objetos representados en nuestro programa contienen tan solo los atributos y comportamientos principales en el contexto del problema, de manera que ignora otras variables menos importantes para no profundizar en detalles no relevantes.
* Encapsulación: nuestro programa contiene objetos encapsulados, de manera que ocultan partes de su estado y comportamiento a otros objetos para que únicamente sean accesibles desde los métodos de la propia clase. Además, las interfaces y clases abstractas usadas en nuestro programa están basadas en los conceptos ya descritos de abstracción y encapsulación.
* Herencia: este es el pilar más fuerte que asegura la reutilización de código, ya que a partir de esta característica es posible reutilizar (heredar) las características y comportamientos de una clase superior llamada clase padre, a sus clases hijas. Nuestro código está constantemente haciendo uso de este pilar de manera que se hace un código muy reutilizable por otras clases o métodos.
* Polimorfismo: son los diferentes comportamientos que tiene un método dependiendo desde la clase que sea llamado en nuestro programa.

## **INTERFAZ DE USUARIO - LUISA**

## **BASE DE DATOS H2**

## **REPORTE DE SONARQUBE**

# CONCLUSIONES

# BIBLIOGRAFÍA

<https://moodleufv.ufv.es/>

<https://www.eclipse.org/>

<https://github.com/>

<https://spring.io/>

<https://vaadin.com/>

<https://maven.apache.org/>

<https://www.h2database.com/html/main.html>

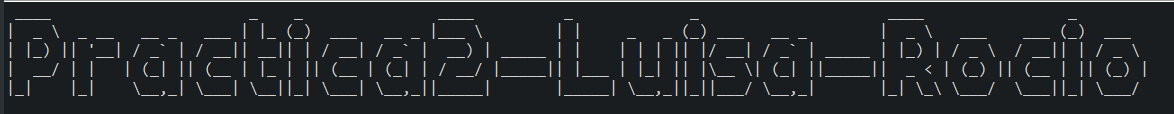
<https://hub.docker.com/>

<https://www.sonarqube.org/>

<https://app.diagrams.net/>

<https://stackoverflow.com/>

<https://mvnrepository.com/>

<http://aprendegit.com/que-es-git-flow/>