Introducción

Al final del curso se realizará un proyecto de PROGRAMACIÓN DE APLICACIONES TELEMÁTICAS elegido por cada equipo de alumnos, lo que permitirá comprender y asimilar los conceptos estudiados. Esta actividad es de suma importancia para la obtención de los objetivos del curso, ya que con ella se quiere conseguir que:

- El alumno sienta la necesidad de utilizar de manera sistemática algún proceso de desarrollo predefinido al enfrentarse a un problema de tal envergadura que no es posible abordarlo de otra manera
- Al tener que aplicar todo lo que se les va explicando en teoría, vayan estudiando la teoría e interiorizándola
- Adquieran experiencia en un trabajo en equipo (que requiere una gran colaboración grupal), analizando explícitamente a final de curso las ventajas que este método de trabajo tiene y sus desventajas, que serán los problemas que necesariamente hayan ido surgiendo en el desarrollo de la actividad
- Sean capaces de abordar su proyecto de fin de carrera con mayor confianza dado que ya se han enfrentado a un proyecto de cierta envergadura
- Experimenten directamente su futura realidad profesional

Características generales

- **Equipos de 4 personas**. Los equipos para la realización del proyecto se organizan según deciden los alumnos.
- Elección del Proyecto por parte del equipo. Para que los grupos estén lo más motivados posibles, cada grupo elegirá su caso de uso. Aunque inicialmente resulta más costoso para los alumnos tener que pensar una aplicación, ya que debe tratarse de una aplicación de cierta envergadura, a la larga, por ser aplicaciones de su interés, se involucran más. En ocasiones estos proyectos han sido el embrión de sus posteriores proyectos fin de grado e incluso de alguna iniciativa de emprendimiento.
- Para intentar simular como se diseña y se produce software en el mundo real se deberá utilizar un flujo de trabajo basado en integración y despliegue continuos. Existirá un entorno de despliegue en Render.com asociado a una rama principal protegida, sobre la que se crearán otras ramas de trabajo con el código de cada funcionalidad a desarrollar. Las ramas de trabajo se probarán en el entorno local, y se mezclarán en la principal solo si pasan todos los tests configurados en la integración continua. Una vez mezclado, se disparará el despliegue en el servidor que da el servicio a los usuarios finales.

Planificación

El proyecto constará de:

- Frontend en HTML + CSS + Javascript
- Backend en Sprint Boot + Base de Datos + Pruebas

Fecha de entrega: **22/05/2024**

Este proyecto se tiene que realizar mayoritariamente fuera de clase con soporte de los profesores para la resolución de dudas y preguntas que pueda surgir.

Herramientas

Para poder tener una visión completa de todas las etapas de la Ingeniería del Software, en la realización de este proyecto no sólo se tendrá que desarrollar su código, sino que además se deberán codificar pruebas, utilizar un repositorio para integrar el software, configurar un workflow de CI/CD, documentar el diseño y la planificación, y completar al menos 2 iteraciones (ciclos o sprints de trabajo de 2 semanas para construir una parte de la funcionalidad y que se pueda ver). Para ello, y con el objetivo de unificar las herramientas a utilizar, a continuación, se detallan las que se utilizarán:

- Herramientas de desarrollo:
  - Visual Studio Code y IntelliJ IDEA
  - HTML, JS, CSS y Java 17+
  - Spring Boot y dependencias necesarias
  - Base de datos H2
  - Repositorio de código en GitHub
  - Git y GitHub Desktop (o cualquier otra herramienta de gestión de ramas)
- Herramientas de integración y despliegue:
  - Maven
  - Workflow CI mediante GitHub Actions > **Java CI with Maven**
  - Docker
  - Workflow CD mediante Render > **Web Service** > **From a GitHub repository**
- Herramientas de documentación y planificación:
  - GitHub Wiki
  - GitHub Projects > **Iterative development**

Entrega

Sólo se va a evaluar el contenido del repositorio, su wiki y el proyecto de GitHub asociado. También se comprobará el despliegue en Render. Para ello, se deberá incluir el usuario de GitHub del profesor como colaborador del repositorio y proyecto de GitHub. En la entrega de Moodle basta con indicar la URL del repositorio en el que el alumno ha colaborado. Los **repositorios** deberán ser **privados**.

Criterios de evaluación

- Planificación **(20p)**:
  - Coherente, clara y realista
  - Definición del **roadmap**: iteraciones (ciclo o sprint)
  - División de tareas y su priorización: **backlog**
  - Objetivos en cada iteración: tareas a completar
  - Asignación de tareas entre los miembros del equipo y su progreso
- Documentación **(20p)**:
  - Calidad, no cantidad
  - Presentación para stakeholders: casos de uso de la aplicación
  - Diseño técnico: API REST, modelo BD, dependencias utilizadas, workflow CI/CD
  - Manual de usuario final: secciones, esquema de navegación
  - Resultado de cada iteración: demo (vídeo o pantallazo de lo conseguido)
- Desarrollo **(30p)**:
  - Estilo del código: conciso, simple, con una correcta indentación, sin que necesite comentarios para ser entendible
  - La implementación utiliza los conceptos dados en clase y en las prácticas
  - Organización de los directorios y ficheros
  - Correcto uso del repositorio y las ramas
  - Correcto funcionamiento de la aplicación en local
  - Grado dificultad de las funcionalidades desarrolladas
  - Experiencia de usuario, coherencia y sentido de utilidad
- Tests **(20p)**:
  - Cobertura, variedad y sentido de los casos de prueba implementados
  - Configuración de la integración continua
- Despliegue **(10p)**:
  - Configuración del despliegue continuo
  - Correcto funcionamiento de la aplicación desplegada

Referencias

- Ramas GIT
- GitHub Desktop
- CI
- CD
- CI/CD
- Java CI with Maven (ver fichero proporcionado **maven.yml**)
- Docker
- Dockerfile (ver fichero proporcionado **Dockerfile**)
- Despliegue (ver fichero proporcionado **imagen Render.com**)

Ficheros

- **maven.yml**

```
name: Java CI with Maven
on:
  pull_request:
    branches: ["main"]
jobs:
  build:
    runs-on: ubuntu-latest
    steps:
    - uses: actions/checkout@v3
    - name: Set up JDK 17
      uses: actions/setup-java@v3
      with:
        java-version: '17'
        distribution: 'temurin'
        cache: maven
    - name: Build with Maven
      run: mvn test
```

- **Dockerfile**

```
FROM maven:3-eclipse-temurin-17-alpine AS build
COPY src /tmp/src
COPY pom.xml /tmp
RUN mvn -f /tmp/pom.xml clean package

FROM eclipse-temurin:17-jdk-alpine
COPY --from=build /tmp/target/*.jar app.jar
EXPOSE 8080
ENTRYPOINT ["java","-jar","app.jar"]
```

- **`Imagen Render.com`**