

Grado en Ingeniería Informática del Software

Ingeniería del Proceso Software (IPS)

Descripción de las prácticas de laboratorio

Curso 2022-2023

1 Introducción

El objetivo de las prácticas de laboratorio es realizar en equipo un proyecto utilizando Scrum como metodología. En estas prácticas se realizará una evaluación continua.

Cada PL se dividirá en equipos, de 4 alumnos. Los alumnos formarán sus propios equipos, aunque el profesor podrá realizar cambios para que se cumplan las cifras anteriores, o cuando haya bajas o nuevas incorporaciones.

2 Artefactos y herramientas utilizadas en cada proyecto

- Tablero Kanban: Para gestionar las historias de usuario. Es una instalación de Redmine a la que se accede con un front-end denominado **RedKanban** (<https://in2test.lsi.uniovi.es/redkanban/>). En esta dirección se encontrará una breve guía rápida de su uso seleccionando el enlace **[Guía Rápida]** en la parte superior derecha.
- Repostorio Git creado por los alumnos en **GitHub** (<https://github.com/>) con el nombre IPS2022-PLxy y que compartirá con el profesor (de uso obligado a partir del segundo sprint).
- Espacio de trabajo en Microsoft Teams (un canal privado creado por el profesor para cada equipo de proyecto):
 - En la carpeta de **Archivos de Teams**, los alumnos mantendrán un archivo Word para incluir la documentación del mismo, denominado **XX2022-PLxy-Documentacion.docx** (XX es el acrónimo de la asignatura, x es el grupo de PL). En el Campus virtual estará disponible el archivo **IPS PL ContenidoDocumentación.pdf** que describe el contenido que debe tener el documento.
 - El profesor creará un **block de notas** OneNote con el nombre **XX2022-PLxy-Notas** que será utilizado para que los alumnos incluyan sus prototipos de pantallas (p.e. una foto) y que los pueda revisar el profesor. Se puede usar desde web o mejor abriéndolo en la aplicación (instalar la app de OneNote desde la tienda de Windows 10)
- Un proyecto de ejemplo para tomar como plantilla siguiendo una arquitectura basada en MVC disponible en <https://github.com/javiertuya/samples-test-dev>.
- Eclipse con Window Builder. El gestor de base de datos será elegido por los alumnos.

3 Proceso

Se utilizará Scrum, con diversas adaptaciones para las prácticas, con los siguientes roles:

- Profesor: Product Owner (PO, cliente) y Scrum Master.
- Estudiantes: Scrum Developer Team.

Las diferentes reuniones típicas de Scrum se personalizan de la siguiente forma:

- **Sprint planning:** Realizada en el primer día de prácticas de cada sprint, aunque se podrá prolongar durante los próximos días hasta que el profesor valide el sprint backlog.
- **Stand-up meeting:** Se realizará al principio de cada día de prácticas (excepto el de fin/inicio de cada sprint), con un máximo de 10 minutos.
- **Sprint review:** Un día de prácticas, en el que acaba un sprint y comienza el siguiente, el profesor realizará la revisión del trabajo realizado por los alumnos, dándoles una valoración alfabética.
- **Retrospective:** Se podrá realizar informalmente tras la revisión del sprint (además de existir sesiones para ello en las prácticas de aula).

4 Organización de los Sprints

Se realizarán tres sprints de tres semanas cada uno más los días de presentación inicial del proyecto y el de presentación por parte de los equipos:

Presentación de la práctica por parte del profesor: El primero o primeros días se tratará la descripción funcional de los objetivos y visión global de la aplicación a realizar. Hasta el día de inicio del primer sprint los alumnos establecerán la arquitectura y entorno tecnológico de la aplicación, creando un proyecto base para el posterior desarrollo y familiarizándose con la arquitectura. Este proyecto base tendrá una arquitectura basada en MVC, para lo que se podrá usar el ejemplo disponible en <https://github.com/javiertuya/samples-test-dev>.

Sprint 1: Los equipos recibirán las historias de usuario para el primer sprint, en RedKanban, estimarán, distribuirán el trabajo y comenzarán el desarrollo.

Sprints 2 y 3: El comienzo del sprint se organizará de la siguiente forma:

- **Día 1:** Los equipos elaborarán las historias de usuario incluyendo los criterios de aceptación. Posteriormente, las estimarán y repartirán el trabajo.
- **Día 2:** Los equipos pedirán al profesor que valide las historias de usuario. En el caso de no tener las historias para validar o no realizar la petición se descontarán como MÍNIMO 0,5ptos, según el retraso acumulado.
- **Días 2-5:** Conversaciones equipo-profesor para aclarar, detallar, refinar... las historias de usuario y acordar el sprint backlog. Se procurará que este proceso sea lo más rápido posible para comenzar cuanto antes del desarrollo.
- **Día 6 (primer día de la segunda semana del sprint):** las historias de usuario deben estar aceptadas. En el caso de no tener las historias aceptadas, se descontarán como MÍNIMO 0,5ptos. según el retraso acumulado.

Además, en estos sprints:

- Se exigirá el uso de Git. Será obligatorio el uso de una rama por cada historia de usuario y el repositorio será GitHub, que deberá ser compartido con el profesor
- Se exigirá que se mantenga actualizado el archivo de documentación con el modelo de datos y prototipos de pantallas (realizados a mano), al menos, que serán utilizados para discutir con el profesor.

- Cada historia de usuario debe realizar una partición en vertical de la funcionalidad, por lo que el desarrollo de esta historia deberá incluir todo lo necesario en la vista, controlador y modelo.
- Como consecuencia de lo anterior, no se permite dividir funcionalidades en función de la arquitectura (p.e. una historia para la vista y otra diferente para el modelo). Esto es así para evitar la necesidad de coordinar el diseño y definir la responsabilidad de la implementación de cada historia, independizando lo más posible la implementación de cada historia respecto de otras.

Último día: Presentaciones de los trabajos por parte de los equipos.