

# MyUPMClassroom

Tu domótica para el aula.



UNIVERSIDAD  
POLITÉCNICA  
DE MADRID

## Entrega 2: Diseño, Implementación y Validación

### Fundamentos de Ingeniería del Software 2022

Universidad Politécnica de Madrid

E.T.S. de Ingeniería en Sistemas Informáticos

Departamento de Sistemas Informáticos

## CASO DE ESTUDIO: DESARROLLO DE MyUPMClassroom

El plazo para esta segunda entrega de la práctica finaliza el **22 de mayo de 2022 a las 23:59** horas. Para esta segunda entrega **se deberán tener en cuenta únicamente las siguientes funcionalidades del sistema:**

- Alta de usuarios (Alumnos, PDI y PAS).
- Alta de aulas.
- Suscripción de alumnos y PDI a aulas.

Se deberán realizar las siguientes actividades:

### 1. DISEÑO

- Diagrama de clases de diseño.** El diagrama de clases de análisis previamente elaborado deberá refinarse, de manera general, para formar un diagrama de clases de diseño. Para ello, deberá añadirse la información correspondiente como tipos de datos a los atributos, especificación de los parámetros y su tipo en los métodos y cualquier otro cambio que se considere necesario. El diagrama también deberá extenderse para añadir las interfaces y nuevas clases (por ejemplo, controladores). Se valorará positivamente aplicar patrones de diseño en las situaciones en las que sea posible y pertinente y evitar aplicar antipatrones de diseño.
- Diagrama de componentes.** Se debe organizar la aplicación en componentes arquitectónicos y generar un diagrama de componentes que los relacione a través de sus interfaces.
- Diagrama de despliegue.** Se deberá esbozar un diagrama de despliegue que represente la implantación del sistema. Para este diagrama (y únicamente para este diagrama) se deberán considerar todas las funcionalidades del sistema especificadas en el enunciado de la práctica.

### 2. IMPLEMENTACIÓN

- Generación automática de código.** A partir del diagrama de clases de diseño y mediante las posibilidades que ofrece StarUML, crear el esqueleto del proyecto.
- Código fuente en Java sobre Eclipse (Oxygen 3 o superior).** A partir de los modelos creados se codificará la parte de la aplicación correspondiente a las funcionalidades indicadas previamente en este documento (alta de usuarios, alta de aulas y suscripción de alumnos y PDI a aulas). No se debe implementar ninguna otra funcionalidad. Por ejemplo, si el acceso a una funcionalidad requiere autenticar al usuario, al no tener que implementarse la autenticación, la aplicación permitirá acceder a dicha funcionalidad a cualquier usuario. Las funcionalidades implementadas deberán ser usables al menos a través de una **interfaz funcional de texto**, con introducción de instrucciones por

## CASO DE ESTUDIO: DESARROLLO DE MyUPMClassroom

**Línea de comandos.** El proyecto a entregar debe tener la estructura generada por un proyecto Java básico de Maven, usando la plantilla *maven-archetype-quickstart*.

Para la implementación de la parte correspondiente al alta de usuarios, se facilitará una librería que deberéis utilizar y que simulará las siguientes funcionalidades, que serán necesarias durante el alta de un usuario:

- La comprobación de la validez de los correos institucionales en el servidor de la UPM.
- El cifrado de la contraseña.

Dicha librería, así como las instrucciones necesarias para su uso, se pondrán accesibles la semana de prácticas en la que se vaya a comenzar con la fase de implementación de la práctica.

Además, se podrán implementar las siguientes **FUNCIONALIDADES OPTATIVAS**:

- Diseño e implementación de una **interfaz gráfica**, que sustituiría la interfaz de texto.
- **Persistencia** de la información mediante ficheros o bases de datos.

En cualquier caso, debe poderse verificar una trazabilidad clara entre las partes codificadas y los modelos previos de análisis y diseño.

**D. Valoración crítica** sobre la aportación que ha tenido para la implementación la realización de las fases previas de ingeniería de software.

### 3. VERIFICACIÓN Y VALIDACIÓN

- A. Pruebas unitarias.** Se deben especificar las pruebas de caja negra de todos los métodos de la clase Alumno e implementar los casos de prueba con junit. Las pruebas se proporcionarán junto con el código fuente dentro de la estructura creada por el proyecto Maven.
- B. Pruebas de aceptación.** Se deben especificar, ejecutar y documentar las pruebas de validación de la funcionalidad responsable de la creación de nuevos alumnos en el sistema. Para acometer esta tarea se realizarán, si fuese necesario, los *mockups* pertinentes.
- C. Trazabilidad.** Permitir la trazabilidad desde el requisito responsable de la creación de nuevos alumnos en el sistema hasta las pruebas de aceptación de dicho requisito, pasando por todos los artefactos relacionados generados a lo largo de todo el proyecto.

## INSTRUCCIONES DE ENTREGA

El trabajo quedará registrado en Redmine y GitLab a fecha **22 de mayo de 2022 a las 23:59**. El *repositorio del proyecto* en GitLab se estructurará en las siguientes carpetas e incluirá los siguientes artefactos:

- **modelado**: contendrá el fichero con extensión .uml correspondiente al proyecto de *StarUML*, así como los diagramas en PNG o JPG y un documento PDF de recopilación.
- **construccion**: contendrá el **código fuente** y la carpeta de test con las **pruebas unitarias**.

El *repositorio del proyecto* deberá sincronizarse con el correspondiente proyecto haciendo uso del sistema de control de versiones.

Además, los estudiantes deberán realizar las siguientes tareas:

- El **líder del equipo de la fase de Pruebas** deberá subir un archivo PDF a la tarea de la entrega 2. En este archivo se detallará el nombre y los enlaces del proyecto en Redmine y GitLab. Además, se indicará el nombre y apellidos de todos los integrantes del equipo que realizan la entrega, identificando la fase del ciclo de vida del software en la que cada uno actúa como líder.
- Cada miembro del equipo deberá coevaluar a cada uno de sus compañeros de grupo (que firman la entrega) a través de la tarea de Moodle “evaluación de competencias transversales”.
- Si se han realizado funcionalidades optativas, estas se deberán describir brevemente en un documento PDF adicional denominado “Opcionales” que se incluirá en la carpeta de implementación.

## EVALUACIÓN

Si se realiza la entrega completa, incluidas las partes de la **implementación** consignadas como **OBLIGATORIAS**, la entrega se valorará sobre una puntuación máxima de **10**. La **realización de funcionalidades OPTATIVAS** permitirá obtener una puntuación extra de 2 puntos, hasta un máximo de **12** puntos.