

Documentación del Proyecto CalificaUPV

Ricardo E. Uriegas Ibarra - 2230122

18 de febrero de 2025

Índice

1. Descripción General	2
2. Pantallas (Wireframes/Mockups)	2
2.1. Pantalla de Inicio	2
2.2. Pantalla de Listado de Profesores	2
2.3. Pantalla de Detalle y Calificación del Profesor	2
2.4. Pantalla de Estadísticas y Resultados	2
2.5. Pantalla de Ayuda/Soporte	2
3. Recursos Funcionales y No Funcionales	3
3.1. Requisitos Funcionales	3
3.2. Requisitos No Funcionales	3
4. Documentación Técnica	3
4.1. Arquitectura del Sistema	3
4.2. Casos de Uso Principales	4
4.3. Especificación de APIs	4
4.4. Manual de Instalación y Despliegue	4
4.5. Pruebas y Calidad	5
5. Cronograma del Proyecto	5
6. Diseño de la Base de Datos	5
6.1. Tabla Profesores	6
6.2. Tabla Calificaciones	6

1. Descripción General

Objetivo:

Desarrollar una aplicación web en la que los alumnos de la UPV puedan calificar de forma anónima a sus profesores, utilizando una escala de 1 a 5 estrellas. La aplicación permitirá recopilar las calificaciones para generar estadísticas y ayudar a mejorar la calidad de la enseñanza.

Público objetivo:

Estudiantes de la UPV que deseen expresar su opinión sobre la labor de sus profesores de manera anónima.

2. Pantallas (Wireframes/Mockups)

Debido a la naturaleza del proyecto, se han definido las siguientes pantallas:

2.1. Pantalla de Inicio

- Breve introducción al funcionamiento de la aplicación.
- Acceso a la sección de calificaciones sin necesidad de registro (la identidad permanece anónima).

2.2. Pantalla de Listado de Profesores

- Visualización de una lista de profesores disponibles para calificar.
- Filtros de búsqueda (por departamento, nombre, etc.).

2.3. Pantalla de Detalle y Calificación del Profesor

- Información básica del profesor (nombre, departamento, foto, etc.).
- Sistema de calificación: opción para seleccionar entre 1 y 5 estrellas.
- Opción para dejar un comentario opcional.
- Botón para enviar la calificación.

2.4. Pantalla de Estadísticas y Resultados

- Visualización de la calificación promedio de cada profesor.
- Gráficos o tablas que muestren la distribución de las calificaciones.

2.5. Pantalla de Ayuda/Soporte

- Información sobre el uso de la plataforma.
- Preguntas frecuentes y contacto para soporte.

3. Recursos Funcionales y No Funcionales

3.1. Requisitos Funcionales

- **Calificación Anónima:**
 - Permitir a los alumnos calificar a los profesores sin identificar al usuario.
 - La calificación debe ser de 1 a 5 estrellas.
- **Visualización de Profesores:**
 - Mostrar un listado de profesores con opción de búsqueda y filtrado.
- **Detalle del Profesor y Calificación:**
 - Permitir ver la información básica del profesor y enviar una calificación.
 - Posibilidad de dejar un comentario opcional.
- **Estadísticas:**
 - Calcular y mostrar la calificación promedio y la distribución de las calificaciones.

3.2. Requisitos No Funcionales

- **Seguridad y Privacidad:**
 - Garantizar el anonimato de los alumnos al realizar las calificaciones.
 - Proteger la integridad de los datos.
- **Usabilidad:**
 - Interfaz sencilla e intuitiva.
 - Diseño responsivo para uso en dispositivos móviles y de escritorio.
- **Rendimiento:**
 - Respuesta rápida en la carga de profesores y envío de calificaciones.
- **Mantenibilidad:**
 - Código modular y bien documentado para facilitar futuras actualizaciones.

4. Documentación Técnica

4.1. Arquitectura del Sistema

- **Frontend:**
 - Aplicación web desarrollada con tecnologías modernas (por ejemplo, React.js, Angular o Vue.js).
- **Backend:**

- API REST para gestionar la obtención de la lista de profesores, envío de calificaciones y consulta de estadísticas.
 - Tecnologías sugeridas: `Node.js`, `Django` o `Flask`.
- **Base de Datos:**
 - Sistema relacional (MySQL, PostgreSQL) para almacenar la información de los profesores y las calificaciones.
 - **Servicios Adicionales:**
 - Integración de herramientas para visualización de datos (por ejemplo, `Chart.js` o `D3.js`) en la sección de estadísticas.

4.2. Casos de Uso Principales

- **CU1 – Visualización de Profesores:**
 - **Actor:** Alumno.
 - **Flujo:** El alumno accede a la lista de profesores, utiliza filtros y selecciona un profesor para ver su detalle.
- **CU2 – Calificación de un Profesor:**
 - **Actor:** Alumno.
 - **Flujo:** Desde el detalle del profesor, el alumno selecciona una calificación (1-5 estrellas), puede agregar un comentario opcional y envía la calificación de manera anónima.
- **CU3 – Visualización de Estadísticas:**
 - **Actor:** Cualquier usuario.
 - **Flujo:** El usuario consulta las calificaciones promedio y la distribución de las calificaciones de cada profesor.

4.3. Especificación de APIs

- **Endpoint de Profesores:** `/api/profesores` (GET para obtener la lista y detalle de profesores).
- **Endpoint de Calificaciones:** `/api/calificaciones` (POST para enviar una nueva calificación, GET para consultar estadísticas).

4.4. Manual de Instalación y Despliegue

Requisitos Previos:

- Servidores configurados (por ejemplo, AWS, Heroku, etc.).
- Variables de entorno para conexión a la base de datos y otros servicios.

Pasos:

1. Clonar el repositorio del proyecto.

2. Instalar las dependencias necesarias (por ejemplo, mediante `npm` o `pip`).
3. Configurar las variables de entorno.
4. Ejecutar las migraciones de la base de datos y los seeds si es necesario.
5. Desplegar el backend y el frontend.

4.5. Pruebas y Calidad

- **Pruebas Unitarias y de Integración:**

- Cobertura de las funcionalidades críticas (envío de calificaciones, obtención de estadísticas).

- **Pruebas de Usabilidad:**

- Testeo con alumnos de la UPV para asegurar la correcta comprensión y uso de la interfaz.

- **Pruebas de Seguridad:**

- Verificar el anonimato en el envío de calificaciones y la integridad de los datos.

5. Cronograma del Proyecto

A continuación se muestra un ejemplo de cronograma estimado (la duración puede variar según el equipo y alcance):

Fase	Duración Estimada	Actividades Clave
Análisis y Requerimientos	1 semana	Reunión inicial, definición de requisitos y alcance.
Diseño UI/UX	2 semanas	Diseño de wireframes, prototipos y validación con usuarios.
Desarrollo Frontend	3 semanas	Implementación de la interfaz, listado de profesores y pantalla de calificación.
Desarrollo Backend & API	3 semanas	Creación de endpoints para profesores y calificaciones, integración con la base de datos.
Integración y Testing	2 semanas	Pruebas unitarias, de integración y ajustes en el flujo de navegación.
Despliegue y Puesta en Producción	1 semana	Configuración de servidores, despliegue y monitoreo.
Documentación y Capacitación	1 semana	Elaboración de manuales y capacitación a administradores.

Total estimado: 13 – 14 semanas.

6. Diseño de la Base de Datos

A continuación se presenta un diseño relacional que incluye las siguientes tablas:

6.1. Tabla Profesores

Campo	Tipo de Dato	Descripción
id_profesor	INT (PK, Auto-Inc)	Identificador único del profesor
nombre	VARCHAR(100)	Nombre del profesor
departamento	VARCHAR(100)	Departamento o área de enseñanza
foto	VARCHAR(255)	URL de la foto del profesor

6.2. Tabla Calificaciones

Campo	Tipo de Dato	Descripción
id_calificacion	INT (PK, Auto-Inc)	Identificador único de la calificación
id_profesor	INT (FK)	Referencia al profesor (<code>Profesores.id_profesor</code>)
estrellas	INT	Calificación en estrellas (1-5)
comentario	TEXT	Comentario opcional
fecha	TIMESTAMP	Fecha y hora de la calificación