

# Desarrollo web para la evaluación por competencias

# Índice general

Índice de figuras .....	1
Índice de cuadros .....	1
Índice de código o algoritmos .....	1
Resumen .....	1
I. Definición de objetivos .....	1
II. Análisis de antecedentes y aportación realizada .....	1
III. Análisis temporal y de costes de desarrollo .....	1
Planificación .....	2
Retrasos e incidencias .....	5
Reuniones .....	6
Costes .....	6
IV. Análisis de requisitos, diseño e implementación .....	7
Requisitos .....	7
Casos de uso .....	9
Diagramas .....	38
Mockups .....	38
Justificaciones de las tecnologías y herramientas utilizadas .....	45
Gestión de código .....	50
Gestión de versiones .....	51
Fuentes e Imágenes .....	52
Implementación del cálculo .....	53
Adaptabilidad del diseño en diferentes resoluciones .....	54
Traducción .....	55
Instalación .....	56
V. Manual de usuario .....	57
VI. Pruebas .....	57
Pruebas de automáticas .....	57
Pruebas de formulario .....	57
Pruebas de borrado .....	57
Pruebas de listado .....	58
VII. Comparativas con otras alternativas .....	58
VIII. Conclusiones y desarrollos futuros .....	58
Glosario .....	58
Casos prácticos de competencias .....	58
Bibliografía .....	59

## **Índice de figuras**

## **Índice de cuadros**

## **Índice de código o algoritmos**

## **Resumen**

## **I. Definición de objetivos**

Los objetivos del proyecto son:

- Desarrollar una aplicación web que permita dar funcionalidad a los profesores inscritos para la correcta evaluación mediante competencias.
- Posibilitar la utilización de la aplicación para todos los niveles de la enseñanza.
- Implementar distintos métodos de cálculo para la evaluación y no dificultar el desarrollo de otros posibles métodos.
- Permitir utilizar la aplicación para el sistema de evaluación tradicional.
- Crear una interfaz simple que permita utilizar la aplicación de forma intuitiva.
- Permitir la correcta visualización en distintas pantallas de sobremesa y alguna de móvil sin perder funcionalidad.
- Internacionalizar la aplicación para que esté disponible en español e inglés.

## **II. Análisis de antecedentes y aportación realizada**

## **III. Análisis temporal y de costes de desarrollo**

# Planificación

La planificación se realizara mediante Scrum, constituidos en nuestro caso por cuatro Sprints, que no son más que una serie de tiempo prefijado con el fin de crear un incremento utilizable y potencialmente entregable. A lo largo del desarrollo de nuestro proyecto habrá Sprints consecutivos, divididos en meses y estos a su vez en semanas con duración constante. Todo esto con el fin de empezar el día 1 de Enero y terminar el día 30 de Abril.

Una vez realizado cada Sprint se verá reflejado en el proyecto como un incremento en el versionado del mismo, para que así se puedan ver los cambios y avances del mismo.

*Entre corchetes se muestra la fecha en la que debe estar finalizada la tarea*

## Sprint 1: 1 de Enero al 28 de Enero.

Semana 1: 1 de Enero al 7 de Enero	Semana 2: 8 de Enero al 14 de Enero	Semana 3: 15 de Enero al 21 de Enero	Semana 4: 22 de Enero al 28 de Enero
Modelo de datos. <i>[3 de Enero]</i>	CU-03 Edición de alumnos. <i>[11 de Enero]</i>	Populate v2. <i>[20 de Enero]</i>	Despliegue de la versión 0.1. <i>[25 de Enero]</i>
Populate v1. <i>[7 de Enero]</i>	CU-04 Eliminación de alumnos. <i>[13 de Enero]</i>	CU-09 Listado de Grupos de clase. <i>[16 de Enero]</i>	Replanificación. <i>[27 de Enero]</i>
CU-01 Listado de alumnos. <i>[4 de Enero]</i>	CU-07 Edición de profesores. <i>[11 de Enero]</i>	CU-10 Creación de Grupos de clase. <i>[18 de Enero]</i>	
CU-02 Creación de alumnos. <i>[5 de Enero]</i>	CU-08 Eliminación de profesores. <i>[13 de Enero]</i>	CU-11 Edición de Grupos de clase. <i>[19 de Enero]</i>	
CU-05 Listado de profesores. <i>[4 de Enero]</i>		CU-12 Eliminación de Grupos de clase. <i>[20 de Enero]</i>	
CU-06 Creación de profesores. <i>[5 de Enero]</i>			

## Sprint 2: 29 de Enero al 25 de Febrero.

<b>Semana 1: 29 de Enero al 4 de Febrero</b>	<b>Semana 2: 5 de Febrero al 11 de Febrero</b>	<b>Semana 3: 12 de Febrero al 18 de Febrero</b>	<b>Semana 4: 19 de Febrero al 25 de Febrero</b>
CU-13 Asignar alumnos a sus respectivos grupos de clase.  <i>[1 de Febrero]</i>	CU-47 Edición de evaluaciones  <i>[8 de Febrero]</i>	CU-23 Asignación de competencias.  <i>[17 de Febrero]</i>	Despliegue de la versión 0.2.  <i>[22 de Febrero]</i>
CU-49 Listado de evaluaciones.  <i>[2 de Febrero]</i>	CU-48 Eliminar evaluaciones.  <i>[9 de Febrero]</i>	CU-24 Listado de administradores.  <i>[13 de Febrero]</i>	Replanificación.  <i>[24 de Febrero]</i>
CU-46 Creación de evaluaciones.  <i>[4 de Febrero]</i>	CU-18 Asignar asignatura a profesores.  <i>[11 de Febrero]</i>	CU-25 Creación de administradores.  <i>[15 de Febrero]</i>	
CU-14 Listado de asignaturas.  <i>[30 de Enero]</i>	CU-19 Listado de competencias.  <i>[6 de Febrero]</i>	CU-26 Eliminación de administradores.  <i>[16 de Febrero]</i>	
CU-15 Creación de asignaturas.  <i>[1 de Febrero]</i>	CU-20 Creación de competencias.  <i>[8 de Febrero]</i>	CU-27 Edición de usuarios.  <i>[18 de Febrero]</i>	
CU-16 Edición de asignaturas.  <i>[3 de Febrero]</i>	CU-21 Edición de competencias.  <i>[10 de Febrero]</i>		
CU-17 Eliminación de asignaturas.  <i>[4 de Febrero]</i>	CU-22 Eliminación de competencias.  <i>[11 de Febrero]</i>		

### **Sprint 3: 26 de Febrero al 25 del Marzo.**

<b>Semana 1: 26 de Febrero al 4 de Marzo</b>	<b>Semana 2: 5 de Marzo al 11 de Marzo</b>	<b>Semana 3: 12 de Marzo al 18 de Marzo</b>	<b>Semana 4: 19 de Marzo al 25 de Marzo</b>
CU-28 Listar sus grupos de clase.  <i>[27 de Febrero]</i>	CU-34 Edición de una actividad.  <i>[7 de Marzo]</i>	CU-40 Asignar competencias a un ejercicio con una intensidad.  <i>[15 de Marzo]</i>	Despliegue de la versión 0.3  <i>[22 de Marzo]</i>

<b>Semana 1: 26 de Febrero al 4 de Marzo</b>	<b>Semana 2: 5 de Marzo al 11 de Marzo</b>	<b>Semana 3: 12 de Marzo al 18 de Marzo</b>	<b>Semana 4: 19 de Marzo al 25 de Marzo</b>
CU-29 Listar sus asignaturas.  <i>[1 de Marzo]</i>	CU-35 Eliminación de una actividad.  <i>[8 de Marzo]</i>	CU-36 Ver las calificaciones de un alumno.  <i>[16 de Marzo]</i>	Replanificación.  <i>[24 de Marzo]</i>
CU-30 Listar las competencias de una asignatura.  <i>[3 de Marzo]</i>	CU-37 Creación de ejercicios.  <i>[7 de Marzo]</i>	CU-42 Modificar sus datos de usuario.  <i>[18 de Marzo]</i>	
CU-31 Listar sus alumnos.  <i>[29 de Febrero]</i>	CU-38 Edición de un ejercicio.  <i>[9 de Marzo]</i>	Responsive.  <i>[18 de Marzo]</i>	
CU-32 Listar actividades.  <i>[1 de Marzo]</i>	CU-39 Eliminación de un ejercicio.  <i>[10 de Marzo]</i>		
CU-33 Creación de actividades.  <i>[3 de Marzo]</i>	Responsive.  <i>[11 de Marzo]</i>		

#### **Sprint 4: 26 de Marzo al 29 de Abril.**

<b>Semana 1: 26 de Marzo al 1 de Abril</b>	<b>Semana 2: 2 de Abril al 8 de Abril</b>	<b>Semana 3: 9 de Abril al 15 de Abril</b>	<b>Semana 4: 16 de Abril al 22 de Abril</b>	<b>Semana 5: 23 de Abril al 29 de Abril</b>
CU-50 Listar bloques (evaluaciones).  <i>[29 de Marzo]</i>	CU-41 Puntuar las evaluaciones, las actividades, los ejercicios y las competencias.  <i>[7 de Abril]</i>	CU-45 Ver competencias de un alumno.  <i>[14 de Abril]</i>	CU-43 Generar informe de un grupo completo.  <i>[19 de Abril]</i>	Despliegue de la versión 1.0  <i>[28 de Abril]</i>
CU-51 Crear bloques (evaluaciones).  <i>[30 de Marzo]</i>	Responsive.  <i>[8 de Abril]</i>	Responsive.  <i>[15 de Abril]</i>	CU-44 Generar informe de un alumno.  <i>[21 de Abril]</i>	
CU-52 Editar bloques (evaluaciones).  <i>[31 de Marzo]</i>			Responsive.  <i>[22 de Abril]</i>	

Semana 1: 26 de Marzo al 1 de Abril	Semana 2: 2 de Abril al 8 de Abril	Semana 3: 9 de Abril al 15 de Abril	Semana 4: 16 de Abril al 22 de Abril	Semana 5: 23 de Abril al 29 de Abril
CU-53 Eliminar bloques (evaluaciones).				
[1 de Abril]				
Responsive.				
[1 de Abril]				

## Retrasos e incidencias

### Retrasos

Tarea	Fecha prevista	Fecha realizada
CU-02 Creación de alumnos.	5 de Enero	7 de Enero
CU-09 Listado de Grupos de clase.	16 de Enero	17 de Enero
Despliegue de la versión 0.1.	25 de Enero	28 de Enero
Replanificación.	27 de Enero	28 de Enero
Despliegue versión 1.0.	27 de Abril	29 de Abril

### Incidencias

Tarea	Fecha de la incidencia
Añadir CRUD de evaluaciones.	28 de Enero
Reordenación casos de uso.	28 de Enero
Error al mostrar botones del navbar.	26 de Febrero
Error al copiar los ejercicios.	19 de Marzo
Error al ocultar botones al copiar.	25 de Marzo
Error de selección de nivel para los cálculos de evaluación por competencias.	16 de Abril
Error de selección de nivel para los cálculos de evaluación por competencias.	16 de Abril
Error 404 de fuente.	19 de Abril
Error de actualización de tipo de evaluación.	22 de Abril
Error de cálculo de la nota de un ejercicio.	22 de Abril
Error de intervalo de nota.	22 de Abril
Error al asignar una competencia.	27 de Abril
Error al desasignar una competencia.	27 de Abril

# Reuniones

En el apartado de las reuniones, tuvimos de varios tipos:

- Una reunión semanal con el tutor, con el fin de recibir feedback y ver el avance del proyecto. En las que uno de los compañeros enseñaba el proyecto y el otro cogía apuntes.
- Una reunión fija de dos horas o más entre los miembros del grupo una vez acabada la reunión con el tutor. En estas reuniones íbamos asignando y reorganizando las tareas, así como hablando de las incidencias y dudas con respecto a las tareas.
- alguna reunión esporádica de horas indefinidas.

## Costes

El proyecto está estimado en **300 horas** de trabajo por persona, basándonos en los 12 créditos del Trabajo Fin de Grado. Por lo tanto, el proyecto supondrá un coste temporal inicial de **600 horas**. Además, los dos miembros del equipo tienen el mismo puesto.

## Salarios

Suponemos que cada miembro cobra unos **8,5 euros la hora**, basándonos en salarios de puestos reales. Por lo que 300 horas sumarán un coste de **2550 euros**. También supondremos un **35%** de costes de la Seguridad Social. Estos costes deberían ser calculados mensualmente, así que su coste real, que supondría un incremento entre 30 y 35%. Escogemos el más alto.

Costes por hora (€/hora)	Salario total (€)	Coste de la Seguridad Social (€)	Coste total por persona (€)	Coste total de la empresa (€)
8,5	2550	892,5	3442,5	6885

## Amortizaciones

Lo único que tendríamos que amortizar son los ordenadores que suponen un coste total de **2000 euros**. Este tipo de elemento se permite amortizar un **20%** anualmente.

Valor total (€)	Coste 2020 (€)	Coste 2021 (€)	Coste total (€)
2000	400	400	800

## Licencias

La única licencia que hemos utilizado es Photoshop que cuesta **24,19 euros al mes** y que hemos utilizado durante 5 meses.

Coste (€/mes)	Coste total (€)
24,19	120,95



## Despliegue

Para el despliegue se utilizó la versión *Hobby* de Heroku, que cuesta **7 dólares**, que son unos **5,74 euros**. Se ha utilizado durante 5 meses.

Coste (€/mes)	Coste total (€)
5,74	28,7

## Coste total

Salarios (€)	Amortizaciones (€)	Licencias (€)	Despliegue (€)	Coste total (€)
6885	800	120,95	28,7	7834,65

## Coste real

# IV. Análisis de requisitos, diseño e implementación

## Requisitos

### Requisitos de Información:

**RI-01:** Una Competencia debe guardar un código de identificación, un nombre y una descripción y su tipo. Las Competencias se dividen en concretas (nivel 1), específicas (nivel 2) y generales (nivel 3). Además debe guardar el peso sobre el nivel superior, las puntuaciones de los alumnos y, si es una competencia específica, el peso sobre la asignatura asociada.

**RI-02:** Una Asignatura debe guardar un nombre, un nivel, el grado, una descripción y las competencias asociadas a ella.

**RI-03:** Un Profesor debe guardar el año de nacimiento, sus iniciales, un nombre de usuario, una contraseña, un nombre, unos apellidos y un email. Además debe guardar las asignaturas asociadas.

**RI-04:** Un Administrador debe guardar el año de nacimiento, sus iniciales, un nombre de usuario, una contraseña, un nombre, unos apellidos y un email.

**RI-05:** Una Evaluación debe guardar un nombre, una fecha de inicio y de fin, un período, una asignatura, una o más actividades. Las Evaluaciones pueden ser final o parciales. Las parciales deben guardar el peso sobre la final. Si una evaluación parcial es creada por un profesor se llamará Bloque.

**RI-06:** Un Grupo debe guardar un nombre, un nivel, un grado, una línea, un profesor, una asignatura, una evaluación y una lista de alumnos.

**RI-07:** Una Actividad debe guardar un título, una fecha, un indicador para saber si es una

recuperación, una lista de ejercicios, una evaluación asociada y el peso sobre ella, un grupo, y una asignatura.

**RI-08:** Un Ejercicio debe guardar un peso sobre la actividad, un enunciado, una nota y una lista de competencias. Además debe guardar el peso de cada competencia sobre el ejercicio y su intensidad.

**RI-09:** Un Estudiante debe guardar un nombre, unos apellidos, una fecha de nacimiento, sus iniciales y las notas en evaluaciones, actividades, ejercicios y competencias.

## **Requisitos Funcionales:**

**RF 01:** El sistema permitirá a los administradores poder ver el listado de alumnos, así como crear, editar y eliminar dichos alumnos.

**RF 02:** El sistema permitirá a los administradores poder ver el listado de profesores, así como crear y eliminar dichos profesores.

**RF 03:** El sistema permitirá a los administradores poder ver el listado de asignaturas, así como crear, editar y eliminar dichas asignaturas.

**RF 04:** El sistema permitirá a los administradores poder ver el listado de evaluaciones, así como crear, editar y eliminar dichas evaluaciones.

**RF 05:** El sistema permitirá a los administradores poder ver el listado de grupos de clase, así como crear, editar y eliminar dichos grupos.

**RF 06:** El sistema permitirá a los administradores poder asignar y desasignar alumnos a un grupo de clase.

**RF 06:** El sistema permitirá a los administradores poder asignar y desasignar asignaturas a un profesor.

**RF 07:** El sistema permitirá a los administradores poder ver el listado de competencias, así como crear, editar y eliminar dichas competencias.

**RF 08:** El sistema permitirá a los administradores poder asignar y desasignar competencias a una asignatura.

**RF 09:** El sistema permitirá a los administradores poder ver el listado de administradores, así como crear y eliminar dichos administradores.

**RF 10:** El sistema permitirá a los administradores poder editar un usuario.

**RF 11:** El sistema permitirá a los profesores poder listar sus grupos de clase.

**RF 12:** El sistema permitirá a los profesores poder listar sus asignaturas.

**RF 13:** El sistema permitirá a los profesores poder listar las competencias de una asignatura.

**RF 14:** El sistema permitirá a los profesores poder seleccionar el tipo de evaluación para las evaluaciones de un grupo de clase.

**RF 15:** El sistema permitirá a los profesores poder listar los alumnos de un grupo de clase.

**RF 16:** El sistema permitirá a los profesores poder ver el listado de actividades, así como crear, editar y eliminar dichas actividades. Además permitirá copiar actividades que ya han sido creadas.

**RF 17:** El sistema permitirá a los profesores poder ver las calificaciones de un alumno.

**RF 18:** El sistema permitirá a los profesores poder crear ejercicios, así como editar y eliminar dichos ejercicios.

**RF 19:** El sistema permitirá a los profesores poder asignar competencias a un ejercicio con una intensidad. Además permitirá editar y desasignar las competencias ya asignadas.

**RF 20:** El sistema permitirá a los profesores poder puntuar evaluaciones, actividades, ejercicios y competencias.

**RF 21:** El sistema permitirá a los profesores poder generar informe de un grupo completo, así como el de un alumno.

**RF 22:** El sistema permitirá a los profesores poder ver las competencias de un alumno.

**RF 23:** El sistema permitirá a los profesores poder ver el listado de bloques, así como crear, editar y eliminar dichos bloques.

**RF 24:** El sistema permitirá a los usuario registrados poder modificar sus datos de perfil.

## Requisitos No Funcionales:

**RNF 01:** El sistema permitirá a los usuarios poder usar la aplicación en español y en inglés.

## Casos de uso

Para poder ver los casos de uso de manera interactiva vaya al siguiente enlace:

- [Administrador](#)
- [Profesor](#)

**CU-01** Listado de alumnos.

Versión: 1.1 (17/05/2021)

Dependencias:

- RF 01.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite el listado de alumnos inscrito*.

Secuencia normal:

1. El administrador desea obtener la lista de alumnos inscritos.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de alumnos.
4. El administrador obtiene el listado de alumnos.

#### **CU-02** Creación de alumnos.

Versión: 1.1 (17/05/2021)

Dependencias:

- RF 01.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite la creación de un alumno*.

Secuencia normal:

1. El administrador desea crear un alumno.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de alumnos.
4. El administrador obtiene el listado de alumnos.
5. El administrador solicita crear un alumno.
6. El administrador crea un alumno.

#### **CU-03** Edición de alumnos.

Versión: 1.1 (17/05/2021)

Dependencias:

- RF 01.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite la edición de un alumno*.

Secuencia normal:

1. El administrador desea editar un alumno.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de alumnos.
4. El administrador obtiene el listado de alumnos.
5. El administrador solicita la edición un alumno.
6. El administrador edita un alumno.

#### **CU-04** Eliminación de alumnos.

Versión: 1.1 (17/05/2021)

Dependencias:

- RF 01.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite la eliminación de un alumno*.

Secuencia normal:

1. El administrador desea eliminar un alumno.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de alumnos.
4. El administrador obtiene el listado de alumnos.
5. El administrador solicita la eliminación un alumno.
6. El administrador elimina un alumno.

#### **CU-05** Listado de profesores.

Versión: 1.1 (17/05/2021)

Dependencias:

- RF 02.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite el listado de profesores inscrito*.

Secuencia normal:

1. El administrador desea obtener la lista de profesores inscritos.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de profesores.
4. El administrador obtiene el listado de profesores.

#### **CU-06 Creación de profesores.**

Versión: 1.1 (17/05/2021)

Dependencias:

- RF 02.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite la creación de un profesor*.

Secuencia normal:

1. El administrador desea crear un profesor.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de profesores.
4. El administrador obtiene el listado de profesores.
5. El administrador solicita crear un profesor.
6. El administrador crea un profesor.

#### **CU-07 Edición de profesores.**

Versión: 1.1 (17/05/2021)

Dependencias:

- RF 10.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite la edición de un profesor*.

Secuencia normal:

1. El administrador desea editar un profesor.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de profesores.
4. El administrador obtiene el listado de profesores.
5. El administrador solicita la edición un profesor.
6. El administrador edita un profesor.

#### **CU-08 Eliminación de profesores.**

Versión: 1.1 (17/05/2021)

Dependencias:

- RF 02.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite la eliminación de un profesor.*

Secuencia normal:

1. El administrador desea eliminar un profesor.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de profesores.
4. El administrador obtiene el listado de profesores.
5. El administrador solicita la eliminación un profesor.
6. El administrador elimina un profesor.

#### **CU-09 Listado de Grupos de clase.**

Versión: 1.1 (17/05/2021)

Dependencias:

- RF 05.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite el listado de grupo de clase.*

Secuencia normal:

1. El administrador desea obtener la lista de grupos de clase.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de grupos de clase.
4. El administrador obtiene el listado de grupos de clase.

#### **CU-10 Creación de Grupos de clase.**

Versión: 1.1 (17/05/2021)

Dependencias:

- RF 05.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite la creación de un grupo de clase.*

Secuencia normal:

1. El administrador desea crear un grupo de clase.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de grupos de clase.
4. El administrador obtiene el listado de grupos de clase.
5. El administrador solicita crear un grupo de clase.
6. El administrador crea un grupo de clase.

#### **CU-11 Edición de Grupos de clase.**

Versión: 1.1 (17/05/2021)

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite la edición de un grupo de clase.*

Dependencias:

- RF 05.



Secuencia normal:

1. El administrador desea editar un profesor.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de grupos de clase.
4. El administrador obtiene el listado de grupos de clase.
5. El administrador solicita la edición un grupo de clase.
6. El administrador edita un grupo de clase.

#### **CU-12 Eliminación de Grupos de clase.**

Versión: 1.1 (17/05/2021)

Dependencias:

- RF 05.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite la eliminación de un grupo de clase.*

Secuencia normal:

1. El administrador desea eliminar un grupo de clase.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de grupos de clase.
4. El administrador obtiene el listado de grupos de clase.
5. El administrador solicita la eliminación un grupo de clase.
6. El administrador elimina un grupo de clase.

#### **CU-13 Asignar alumnos a sus respectivos grupos de clase.**

Versión: 1.1 (17/05/2021)

Dependencias:

- RF 06.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador asignar alumnos a sus respectivos grupos de clase.*

**Secuencia normal:**

1. El administrador desea asignar alumnos a sus respectivos grupos de clase.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de grupos de clase.
4. El administrador obtiene el listado de grupos de clase.
5. El administrador solicita la lista de alumnos no inscritos.
6. El administrador obtiene la lista de alumnos no inscritos.
7. El administrador solicita la asignación de un alumno.

**CU-14** Listado de asignaturas.

Versión: 1.1 (17/05/2021)

**Dependencias:**

- RF 03.

**Descripción:** El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite el listado de asignaturas.*

**Secuencia normal**

1. El administrador desea obtener la lista de asignaturas.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de asignaturas.
4. El administrador obtiene el listado de asignaturas.

**CU-15** Creación de asignaturas.

Versión: 1.0 (17/05/2021)

**Dependencias:**

- RF 03.

**Descripción:** El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite la creación de una asignatura.*

Secuencia normal:

1. El administrador desea crear una asignatura.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de asignaturas.
4. El administrador obtiene el listado de asignaturas.
5. El administrador solicita crear una asignatura.
6. El administrador crea una asignatura.

#### **CU-16 Edición de asignaturas.**

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 03.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite la edición de una asignatura.*

Secuencia normal:

1. El administrador desea editar una asignatura.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de asignaturas.
4. El administrador obtiene el listado de asignaturas.
5. El administrador solicita la edición una asignatura.
6. El administrador edita una asignatura.

#### **CU-17 Eliminación de asignaturas.**

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 03.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite la eliminación de una asignatura.*

Secuencia normal:

1. El administrador desea eliminar un alumno.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de asignaturas.
4. El administrador obtiene el listado de asignaturas.
5. El administrador solicita la eliminación una asignatura.
6. El administrador elimina una asignatura.

**CU-18** Asignar asignatura a profesores.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 06.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador asignar una asignatura a un profesor.*

Secuencia normal:

1. El administrador desea asignar una asignatura a un profesor.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de profesores.
4. El administrador obtiene el listado de profesores.
5. El administrador solicita la lista de asignaturas no inscritas.
6. El administrador obtiene la lista de asignaturas no inscritas.
7. El administrador solicita la asignación de una asignatura.

**CU-19** Listado de competencias.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 07.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite el listado de competencias.*

Secuencia normal:

1. El administrador desea obtener la lista de competencias.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de competencias.
4. El administrador obtiene el listado de competencias.

**CU-20** Creación de competencias.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 07.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite la creación de una competencia*.

Secuencia normal:

1. El administrador desea crear una competencia.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de competencias.
4. El administrador obtiene el listado de competencias.
5. El administrador solicita crear una competencia.
6. El administrador crea una competencia.

**CU-21** Edición de competencias.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 07.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite la edición de una competencia*.

Secuencia normal:

1. El administrador desea editar una competencia.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de competencias.
4. El administrador obtiene el listado de competencias.
5. El administrador solicita la edición una competencia.
6. El administrador edita una competencia.

#### **CU-22 Eliminación de competencias.**

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 07.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite la eliminación de una competencia.*

Secuencia normal:

1. El administrador desea eliminar una competencia.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de competencias.
4. El administrador obtiene el listado de competencias.
5. El administrador solicita la eliminación una competencia.
6. El administrador elimina una competencia.

#### **CU-23 Asignación de competencias.**

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 08.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite la asignación de competencias a una asignatura.*

Secuencia normal:

1. El administrador desea asignar una competencia a una asignatura.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de asignaturas.
4. El administrador obtiene el listado de asignaturas.
5. El administrador solicita la lista de competencias.
6. El administrador obtiene el listado de competencias.
7. El administrador asigna la competencia de una asignatura.

#### **CU-24** Listado de administradores.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 09.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite el listado de administradores*.

Secuencia normal:

1. El administrador desea obtener la lista de administradores.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de administradores.
4. El administrador obtiene el listado de administradores.

#### **CU-25** Creación de administradores.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 09.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite la creación de un nuevo administrador*.

Secuencia normal:

1. El adminsitrador desea crear un administrador.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de administradores.
4. El administrador obtiene el listado de administradores.
5. El administrador solicita crear un administrador.
6. El administrador crea un administrador.

**CU-26** Eliminación de administradores.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 09.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite la eliminación de un administrador*.

Secuencia normal:

1. El adminsitrador desea eleminar un administrador.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de administradores.
4. El administrador obtiene el listado de administradores.
5. El administrador solicita la eliminación un administrador.
6. El administrador elimina un administrador.

**CU-28** Listar sus grupos de clase.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 11.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el profesor solicite listar sus grupos de clase*.



Secuencia normal:

1. El profesor desea listar sus grupos de clase.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la lista de sus grupos de clase.
4. El profesor obtiene el listado de sus grupos de clase.

**CU-29** Listar sus asignaturas.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 12.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el profesor solicite listar sus asignaturas*.

Secuencia normal:

1. El profesor desea listar sus asignaturas.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la lista de sus asignaturas.
4. El profesor obtiene el listado de sus asignaturas.

**CU-30** Listar las competencias de una asignatura.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 13.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el profesor solicite listar las competencias de una asignatura*.

Secuencia normal:

1. El profesor desea listar las competencias de una asignatura.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la lista de competencias de una asignatura.
4. El profesor obtiene el listado de competencias de una asignatura.

**CU-31** Listar sus alumnos.

Versión: 1.1 (17/05/2021)

Dependencias:

- RF 15.

Precondición: El profesor ha seleccionado el tipo de evaluación.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el profesor solicite listar sus alumnos*.

Secuencia normal:

1. El profesor desea listar sus alumnos.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la lista de grupos de clase.
4. El profesor obtiene el listado de grupos de clase.
5. El profesor solicita la lista de sus alumnos.
6. El profesor obtiene el listado de sus alumnos.

**CU-32** Listar actividades.

Versión: 1.1 (17/05/2021)

Dependencias:

- RF 16.

Precondición: El profesor ha seleccionado el tipo de evaluación.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el profesor solicite actividades*.

Secuencia normal:

1. El profesor desea listar actividades.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la lista de actividades.
4. El profesor obtiene el listado de actividades.

**CU-33** Creación de actividades.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 16.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el profesor solicite la creación de una actividad*.

Secuencia normal:

1. El profesor desea crear una actividad.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la lista de sus grupos.
4. El profesor obtiene el listado de sus grupos.
5. El profesor solicita las actividades de un grupo.
6. El profesor obtiene las actividades de un grupo.
7. El profesor crea una actividad.

**CU-34** Edición de una actividad.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 16.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el profesor solicite la edición de una actividad*.

Secuencia normal:

1. El profesor desea editar una actividad.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la lista de sus grupos.
4. El profesor obtiene el listado de sus grupos.
5. El profesor solicita las actividades de un grupo.
6. El profesor obtiene las actividades de un grupo.
7. El profesor solicita los detalles de la actividad.
8. El profesor obtiene los detalles de la actividad.
9. El profesor edita una actividad.

**CU-35** Eliminación de una actividad.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 16.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el profesor solicite la eliminación de una actividad*.

Secuencia normal:

1. El profesor desea editar una actividad.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la lista de sus grupos.
4. El profesor obtiene el listado de sus grupos.
5. El profesor solicita las actividades de un grupo.
6. El profesor obtiene las actividades de un grupo.
7. El profesor solicita los detalles de la actividad.
8. El profesor obtiene los detalles de la actividad.
9. El profesor elimina una actividad.

**CU-36** Ver las calificaciones de un alumno.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 17.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el profesor solicite ver las calificaciones de un alumno*.

Secuencia normal:

1. El profesor desea ver las calificaciones de un alumno.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la lista de sus grupos.
4. El profesor obtiene el listado de sus grupos.
5. El profesor solicita la lista de alumnos de ese grupo.
6. El profesor obtiene la lista de alumnos de ese grupo.
7. El profesor ve las califiaciones de un alumno.

**CU-37** Listar ejercicios.

Versión: 1.1 (17/05/2021)

Dependencias:

- RF 18.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el profesor solicite el listado de ejercicio*.

Secuencia normal:

1. El profesor desea listar ejercicios.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la lista de ejercicios.
4. El profesor obtiene el listado de ejercicios.

**CU-38** Creación de ejercicios.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 18.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el profesor solicite la creación de un ejercicio*.

Secuencia normal:

1. El profesor desea crear un ejercicio.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la lista de sus grupos.
4. El profesor obtiene el listado de sus grupos.
5. El profesor solicita las actividades de un grupo.
6. El profesor obtiene las actividades de un grupo.
7. El profesor solicita los detalles de la actividad.
8. El profesor obtiene los detalles de la actividad.
9. El profesor crea un ejercicio.

**CU-39** Edición de un ejercicio.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 18.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el profesor solicite la edición de un ejercicio*.

Secuencia normal:

1. El profesor desea editar un ejercicio.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la lista de sus grupos.
4. El profesor obtiene el listado de sus grupos.
5. El profesor solicita las actividades de un grupo.
6. El profesor obtiene las actividades de un grupo.
7. El profesor solicita los detalles de la actividad.
8. El profesor obtiene los detalles de la actividad.
9. El profesor edita un ejercicio.

**CU-40** Eliminación de un ejercicio.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 18.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el profesor solicite la eliminación de un ejercicio*.

Secuencia normal:

1. El profesor desea eliminar un ejercicio.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la lista de sus grupos.
4. El profesor obtiene el listado de sus grupos.
5. El profesor solicita las actividades de un grupo.
6. El profesor obtiene las actividades de un grupo.
7. El profesor solicita los detalles de la actividad.
8. El profesor obtiene los detalles de la actividad.
9. El profesor elimina un ejercicio.

**CU-41** Asignar competencias a un ejercicio con una intensidad.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 19.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el profesor solicite la asignar competencias a un ejercicio con una intensidad.*

Secuencia normal:

1. El profesor desea asignar competencias a un ejercicio con una intensidad.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la lista de sus grupos.
4. El profesor obtiene el listado de sus grupos.
5. El profesor solicita las actividades de un grupo.
6. El profesor obtiene las actividades de un grupo.
7. El profesor solicita los detalles de la actividad.
8. El profesor obtiene los detalles de la actividad.
9. El profesor solicita modificar un ejercicio.
10. El profesor asigna una competencia a un ejercicio con una intensidad.

**CU-42** Puntuar las evaluaciones, las actividades, los ejercicios y las competencias.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 20.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el profesor solicite puntuar las evaluaciones, las actividades, los ejercicios y las competencias.*



Secuencia normal:

1. El profesor desea puntuar las evaluaciones, las actividades, los ejercicios y las competencias.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la lista de sus grupos.
4. El profesor obtiene el listado de sus grupos.
5. El profesor solicita la lista de alumnos de ese grupo.
6. El profesor obtiene la lista de alumnos de ese grupo.
7. El profesor solicita ver las calificaciones.
8. El profesor obtiene las calificaciones.
9. El profesor solicita calificar las actividades.
10. El profesor solicita calificar los ejercicios.
11. El profesor solicita calificar las competencias.
12. El profesor puntúa las competencias.
13. El profesor obtiene las calificaciones de las competencias, ejercicios y actividades y evaluación.

**CU-43** Modificar sus datos de usuario.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 24.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el profesor solicite modificar sus datos de usuario*.

Secuencia normal:

1. El profesor desea modificar sus datos de usuario.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la modificación sus datos de usuario.
4. El profesor modifica sus datos de usuario.

**CU-44** Generar informe de un grupo completo.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 21.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el profesor solicite generar informe de un grupo completo*.

Secuencia normal:

1. El profesor desea generar informe de un grupo completo.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la lista de grupos de clase.
4. El profesor obtiene el listado de grupos de clase.
5. El profesor solicita la lista de alumnos.
6. El profesor obtiene el listado de alumnos.
7. El profesor solicita el informe.
8. El profesor exporta el informe.

**CU-45** Generar informe de un alumno.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 21.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el profesor solicite generar informe de un alumno*.

Secuencia normal:

1. El profesor desea generar informe de un alumno.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la lista de grupos de clase.
4. El profesor obtiene el listado de grupos de clase.
5. El profesor solicita la lista de alumnos.
6. El profesor obtiene el listado de alumnos.
7. El profesor ve las calificaciones de un alumno.
8. El profesor exporta el informe.

#### **CU-46** Ver competencias de un alumno.

Versión: 1.0 (17/05/2021)

##### Dependencias:

- RF 22.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el profesor solicite ver las competencias de un alumno*.

##### Secuencia normal:

1. El profesor desea ver las competencias de un alumno.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la lista de grupos de clase.
4. El profesor obtiene el listado de grupos de clase.
5. El profesor solicita la lista de alumnos.
6. El profesor obtiene el listado de alumnos.
7. El profesor solicita las competencias de un alumno.
8. El profesor obtiene las competencias de un alumno.

#### **CU-47** Creación de evaluaciones.

Versión: 1.0 (17/05/2021)

##### Dependencias:

- RF 04.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite la creación de una evaluación*.

##### Secuencia normal:

1. El administrador desea crear una evaluación.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de evaluaciones.
4. El administrador obtiene el listado de evaluaciones.
5. El administrador solicita crear una evaluación.
6. El administrador crea una evaluación.

**CU-48** Edición de evaluaciones.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 04.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite la edición de una evaluación*.

Secuencia normal:

1. El adminsitrador desea editar una evaluacion.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de evaluaciones.
4. El administrador obtiene el listado de evaluaciones.
5. El administrador solicita la edición una evaluación.
6. El administrador edita una evaluación.

**CU-49** Eliminación de evaluaciones.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 04.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite la eliminación de una evaluación*.

Secuencia normal:

1. El adminsitrador desea eliminar una evaluación.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de evaluaciones.
4. El administrador obtiene el listado de evaluaciones.
5. El administrador solicita la eliminación una evaluación.
6. El administrador elimina una evaluación.

**CU-50** Listado de evaluaciones.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 04.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite el listado de las evaluaciones*.

Secuencia normal:

1. El adminsitrador desea obtener la lista de evaluaciones.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de grupos de clase.
4. El administrador obtiene el listado de evaluaciones.

**CU-51** Listado de bloques (evaluaciones).

Versión: 1.1 (17/05/2021)

Dependencias:

- RF 23.

Precondición: El profesor ha seleccionado el tipo de evaluación.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el profesort solicite el listado de los bloques*.

Secuencia normal:

1. El profesor desea obtener la lista de bloques.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la lista de grupos de clase.
4. El profesor obtiene el listado de grupos de clase.
5. El profesor solicita la lista de bloques de un grupo de clase.
6. El profesor obtiene el listado de bloques.

**CU-52** Creación de bloques (evaluaciones).

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 23.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el profesort solicite la creación de un bloque*.

Secuencia normal:

1. El profesor desea crear un bloque.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la lista de grupos de clase.
4. El profesor obtiene el listado de grupos de clase.
5. El profesor solicita la lista de bloques de un grupo de clase.
6. El profesor obtiene el listado de bloques.
7. El profesor solicita la creación de un bloque.
8. El profesor crea un bloque.

**CU-53** Edición de bloques (evaluaciones).

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 23.

Precondición: El profesor ha creado un bloque.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el profesor solicite la edición de un bloque*.

Secuencia normal:

1. El profesor desea editar un bloque.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la lista de grupos de clase.
4. El profesor obtiene el listado de grupos de clase.
5. El profesor solicita la lista de bloques de un grupo de clase.
6. El profesor obtiene el listado de bloques.
7. El profesor solicita la edición de un bloque.
8. El profesor edita un bloque.

**CU-54** Eliminación de bloques (evaluaciones).

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 23.

Precondición: El profesor ha creado un bloque.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el profesort solicite la eliminación de un bloque*.

Secuencia normal:

1. El profesor desea eliminar un bloque.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la lista de grupos de clase.
4. El profesor obtiene el listado de grupos de clase.
5. El profesor solicita la lista de bloques de un grupo de clase.
6. El profesor obtiene el listado de bloques.
7. El profesor solicita la eliminación de un bloque.
8. El profesor elimina un bloque.

**CU-55** Seleccionar el tipo de evaluación para las evaluaciones de un grupo de clase.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 14.

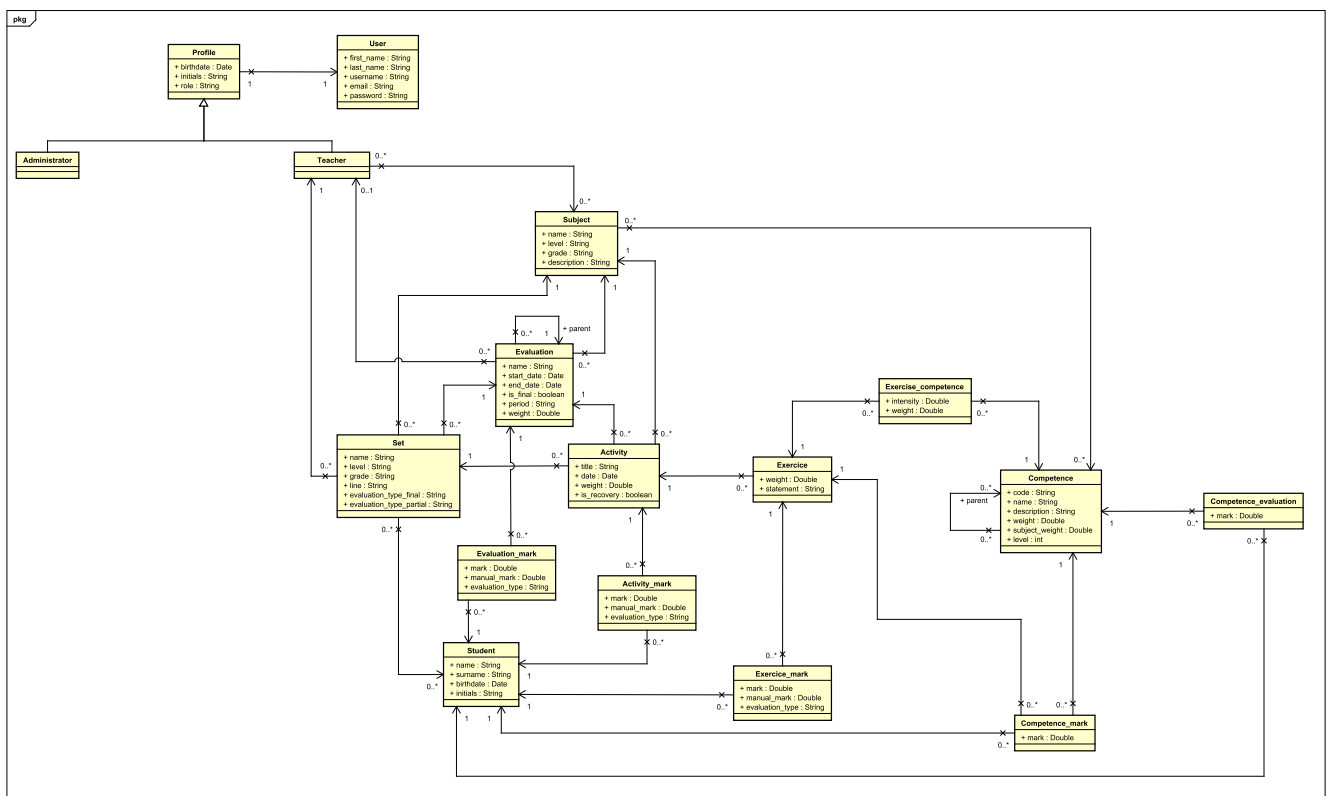
Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el profesor solicite seleccionar el tipo de evaluación para las evaluaciones de un grupo de clase.

Secuencia normal:

1. El profesor desea seleccionar el tipo de evaluación para las evaluaciones de un grupo de clase.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la lista de grupos de clase.
4. El profesor obtiene el listado de grupos de clase.
5. El profesor solicita seleccionar el tipo de evaluación.
6. El profesor seleccionar el tipo de evaluación.

Postcondición: Se ha habilitado las opciones de listar bloques, actividades y estudiantes.

## Diagramas



## Mockups

En esta sección mostraremos algunos de los mockups creados previamente al desarrollo de la aplicación. El resto de mockups se encuentran disponibles dentro del repositorio de GitHub, en la



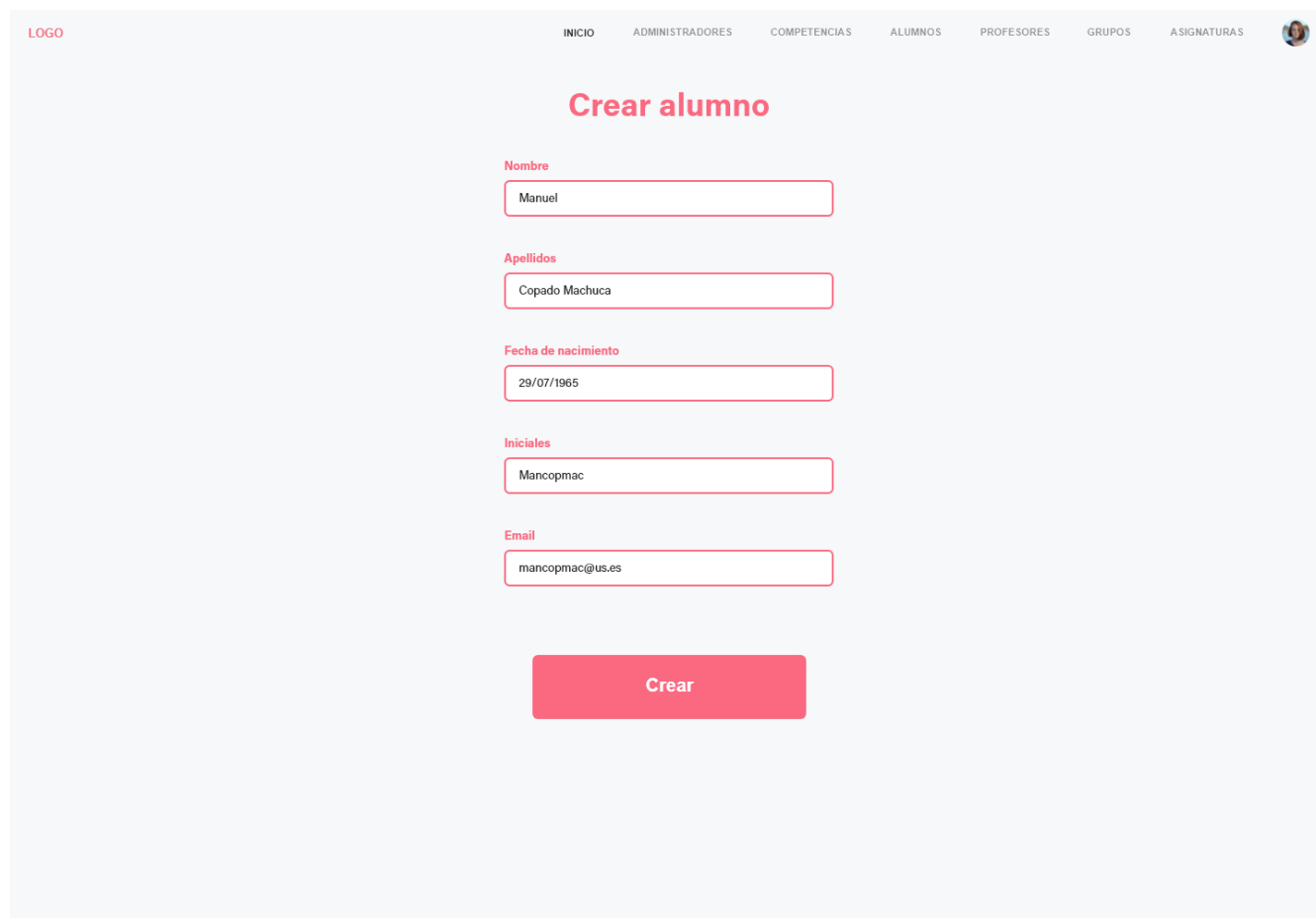
sección docs/mockups.

También en el directorio de docs podremos encontrar mockups-interactivos, en los que podremos ver de manera interactiva todos los mockups creados.

Dividiremos esta guía en dos partes, administrador y profesor.

- Administrador:

Aquí podremos ver la creación de un alumno:



The screenshot shows a web application interface for creating a student. At the top, there is a navigation bar with the following items: LOGO, INICIO, ADMINISTRADORES, COMPETENCIAS, ALUMNOS, PROFESORES, GRUPOS, ASIGNATURAS, and a user profile icon. The main heading is 'Crear alumno'. Below this, there are five input fields, each with a label in red text above it: 'Nombre' (containing 'Manuel'), 'Apellidos' (containing 'Copado Machuca'), 'Fecha de nacimiento' (containing '29/07/1965'), 'Iniciales' (containing 'Mancopmac'), and 'Email' (containing 'mancopmac@us.es'). At the bottom of the form is a red button labeled 'Crear'.

La creación de un grupo:

LOGO

INICIO

ADMINISTRADORES

COMPETENCIAS

ALUMNOS

PROFESORES

GRUPOS

ASIGNATURAS

Crear grupo

Nivel

Select...

Curso

Select...

Línea

Select...

Asignatura

Matemática

Matemáticas 2

Matemáticas Primaria

Tutor

Fernando F

Fernando Fernández González

Fernando Fernández Arévalo

Crear

El listado de asignación de unos alumnos a un grupo:

LOGO

INICIO

ADMINISTRADORES

COMPETENCIAS

ALUMNOS

PROFESORES

GRUPOS

ASIGNATURAS

Asignación de alumnos a grupo

Lina Bo


Buscar

BASIC INFO	COMPANY	PHONE	TAGS	CREATED DATE
<input type="checkbox"/> <div>Lina Bo</div> lina@adobe.com	Adobe	012 389 438	POPULAR NEW JOB	09 Sep 2017 →
<input type="checkbox"/> <div>Lina Bo</div> lina@adobe.com	Adobe	012 389 438	POPULAR NEW JOB	09 Sep 2017 →
<input type="checkbox"/> <div>Lina Bo</div> lina@adobe.com	Adobe	012 389 438	POPULAR NEW JOB	09 Sep 2017 →
<input type="checkbox"/> <div>Lina Bo</div> lina@adobe.com	Company Name	012 389 438	POPULAR NEW JOB	09 Sep 2017 →
<input type="checkbox"/> <div>Lina Bo</div> lina@adobe.com	Adobe	012 389 438	POPULAR NEW JOB	09 Sep 2017 →
<input type="checkbox"/> <div>Lina Bo</div> lina@adobe.com	Adobe	012 389 438	POPULAR NEW JOB	09 Sep 2017 →
<input type="checkbox"/> <div>Lina Bo</div> lina@adobe.com	Adobe	012 389 438	POPULAR NEW JOB	09 Sep 2017 →
<input type="checkbox"/> <div>Lina Bo</div> lina@adobe.com	Adobe	012 389 438	POPULAR NEW JOB	09 Sep 2017 →
<input type="checkbox"/> <div>Lina Bo</div> lina@adobe.com	Adobe	012 389 438	POPULAR NEW JOB	09 Sep 2017 →
<input type="checkbox"/> <div>Lina Bo</div> lina@adobe.com	Adobe	012 389 438	POPULAR NEW JOB	09 Sep 2017 →
<input type="checkbox"/> <div>Lina Bo</div> lina@adobe.com	Adobe	012 389 438	POPULAR NEW JOB	09 Sep 2017 →
<input type="checkbox"/> <div>Lina Bo</div> lina@adobe.com	Adobe	012 389 438	POPULAR NEW JOB	09 Sep 2017 →

La eliminación de estos de un grupo:






LOGO

INICIOADMINISTRADORESCOMPETENCIASALUMNOSPROFESORESGRUPOSASIGNATURAS



## Eliminar alumnos de un grupo

Buscar

	BASIC INFO	COMPANY	PHONE	TAGS	CREATED DATE	
<input type="checkbox"/>	 Lina Bo lina@adobe.com	Adobe	012 389 438	POPULARNEWJOB	09 Sep 2017	✕
<input type="checkbox"/>	 Lina Bo lina@adobe.com	Adobe	012 389 438	POPULARNEWJOB	09 Sep 2017	✕
<input type="checkbox"/>	 Lina Bo lina@adobe.com	Adobe	012 389 438	POPULARNEWJOB	09 Sep 2017	✕
<input type="checkbox"/>	 Lina Bo lina@adobe.com	Company Name	012 389 438	POPULARNEWJOB	09 Sep 2017	✕
<input type="checkbox"/>	 Lina Bo lina@adobe.com	Adobe	012 389 438	POPULARNEWJOB	09 Sep 2017	✕

- Profesor:

El listado de alumnos de un grupo al que pertenece:


LOGO

INICIO

MIS ACTIVIDADES






MIS GRUPOS

MIS ASIGNATURAS



Alumnos Grupo 3° B

Informes

BASIC INFO		COMPANY	PHONE	TAGS				
<input type="checkbox"/>	 Lina Bo lina@adobe.com	Adobe	012 389 438	POPULAR	NEW	JOB	Competencias	Calificación
<input type="checkbox"/>	 Lina Bo lina@adobe.com	Adobe	012 389 438	POPULAR	NEW	JOB	Competencias	Calificación
<input type="checkbox"/>	 Lina Bo lina@adobe.com	Adobe	012 389 438	POPULAR	NEW	JOB	Competencias	Calificación
<input type="checkbox"/>	 Lina Bo lina@adobe.com	Company Name	012 389 438	POPULAR	NEW	JOB	Competencias	Calificación
<input type="checkbox"/>	 Lina Bo lina@adobe.com	Adobe	012 389 438	POPULAR	NEW	JOB	Competencias	Calificación

La asignación de una competencia a una asignatura:

LOGO

INICIO

ADMINISTRADORES


COMPETENCIAS

ALUMNOS

PROFESORES













GRUPOS

ASIGNATURAS




Asignación de competencia a asignatura

Buscar

BASIC INFO		COMPANY	PHONE	TAGS			CREATED DATE	
<input type="checkbox"/>	 Lina Bo lina@adobe.com	Adobe	012 389 438	POPULAR	NEW	JOB	09 Sep 2017	→
<input type="checkbox"/>	 Lina Bo lina@adobe.com	Adobe	012 389 438	POPULAR	NEW	JOB	09 Sep 2017	→
<input type="checkbox"/>	 Lina Bo lina@adobe.com	Adobe	012 389 438	POPULAR	NEW	JOB	09 Sep 2017	→
<input type="checkbox"/>	 Lina Bo lina@adobe.com	Company Name	012 389 438	POPULAR	NEW	JOB	09 Sep 2017	→
<input type="checkbox"/>	 Lina Bo lina@adobe.com	Adobe	012 389 438	POPULAR	NEW	JOB	09 Sep 2017	→
<input type="checkbox"/>	 Lina Bo lina@adobe.com	Adobe	012 389 438	POPULAR	NEW	JOB	09 Sep 2017	→
<input type="checkbox"/>	 Lina Bo lina@adobe.com	Adobe	012 389 438	POPULAR	NEW	JOB	09 Sep 2017	→
<input type="checkbox"/>	 Lina Bo lina@adobe.com	Adobe	012 389 438	POPULAR	NEW	JOB	09 Sep 2017	→
<input type="checkbox"/>	 Lina Bo lina@adobe.com	Adobe	012 389 438	POPULAR	NEW	JOB	09 Sep 2017	→
<input type="checkbox"/>	 Lina Bo lina@adobe.com	Adobe	012 389 438	POPULAR	NEW	JOB	09 Sep 2017	→
<input type="checkbox"/>	 Lina Bo lina@adobe.com	Adobe	012 389 438	POPULAR	NEW	JOB	09 Sep 2017	→
<input type="checkbox"/>	 Lina Bo lina@adobe.com	Adobe	012 389 438	POPULAR	NEW	JOB	09 Sep 2017	→

Poder ver las calificaciones de un alumno:

LOGO

INICIOMIS ACTIVIDADESMIS GRUPOSMIS ASIGNATURAS

## Competencias Generales Alberto Souto Cruz

<b>C-01:</b> Nombre C-01 Descripción C-01	<b>7.2</b> <a href="#">Competencias Especificas</a>
<b>C-02:</b> Nombre C-02 Descripción C-02	<b>7.2</b> <a href="#">Competencias Especificas</a>
<b>C-03:</b> Nombre C-03 Descripción C-03	<b>6.3</b> <a href="#">Competencias Especificas</a>
<b>C-04:</b> Nombre C-04 Descripción C-04	<b>9.1</b> <a href="#">Competencias Especificas</a>

Poder puntuar las calificaciones mediante competencias:


LOGO

INICIO

MIS ACTIVIDADES

MIS GRUPOS

MIS ASIGNATURAS



Calificaciones Alberto Souto Cruz

Competencias Ejercicio 3

Competencia 1: Conocimientos para la realización de mediciones, cálculos, valoraciones, tasaciones, estudios, informes, planificación de tareas y otros trabajos análogos de informática.

Nota

-

Guardar

Competencia 2: Capacidad para resolver problemas con iniciativa, toma de decisiones, autonomía y creatividad. Capacidad para saber comunicar y transmitir los conocimientos, habilidades y destrezas de la profesión.

Nota

-

Guardar

Modificar un ejercicio:


LOGO

INICIO

MIS ACTIVIDADES

MIS GRUPOS

MIS ASIGNATURAS



Modificar ejercicio



Enunciado

Estimación del porcentaje de alumnos que pagan 550 € o más por la matrícula en una población de tamaño infinito.

Peso






Select...

Modificar

	BASIC INFO	COMPANY	PHONE	Peso	Intensidad	
<input type="checkbox"/>	 Lina Bo lina@adobe.com	Adobe	012 389 438	Select	Select	→ ×
<input type="checkbox"/>	 Lina Bo lina@adobe.com	Adobe	012 389 438	Select	Select	→ ×

Capacidad o

Buscar

	BASIC INFO	COMPANY	PHONE	Peso	Intensidad	
<input type="checkbox"/>	 Lina Bo lina@adobe.com	Adobe	012 389 438	Select	Select	→
<input type="checkbox"/>	 Lina Bo lina@adobe.com	Adobe	012 389 438	Select	Select	→
<input type="checkbox"/>	 Lina Bo lina@adobe.com	Adobe	012 389 438	Select	Select	→
<input type="checkbox"/>	 Lina Bo lina@adobe.com	Company Name	012 389 438	Select	Select	→
<input type="checkbox"/>	 Lina Bo lina@adobe.com	Adobe	012 389 438	Select	Select	→

# Justificaciones de las tecnologías y herramientas utilizadas

## Tecnologías 🍄

### Django

Este es uno de los mejores entornos de trabajo, o [Framework](#) de Python muy completo y con una buena documentación. Tenemos experiencia en él, ya que lo hemos utilizado en otras asignaturas, y nos gustaría seguir aprendiendo y adquirir experiencia en Django para nuestro futuro profesional.

### Alternativas

Existen otros frameworks de Python como:

- Flask.
- Bottle.
- Pyramid.

Bottle y Flask son microframework mucho más simples que Django y se utilizan para proyectos más sencillos que el nuestro.

Pyramid es una buena opción, mucho más ligero que Django, esto provoca que sea menos completo. Django no necesita otras librerías para el desarrollo de la aplicación, al contrario que Pyramid, que aparte del enrutamiento y la autenticación, necesita librerías externas.

NodeJs es otra buena alternativa. Es un framework que está creciendo continuamente en popularidad con multitud de librerías que lo nutren. Esto también juega en su contra, ya que NodeJs no dispone de una reputación tan sólida, ni unas librerías externas tan bien documentadas como Django. Tampoco dispone de un sistema sólido de seguridad como el que tiene Django, el cual, es muy fácil de implementar. En otro apartado, Django permite una velocidad de desarrollo muy alta, algo de lo que no dispone NodeJs, por lo que la elección de este último extendería más la etapa de desarrollo, y lo haría menos económico.

### Referencias

- <https://www.monocubed.com/django-vs-node-js/>
- <https://www.upgrad.com/blog/django-vs-nodejs-difference-between-django-and-nodejs/>
- <https://openwebinars.net/blog/los-4-mejores-frameworks-para-aplicaciones-de-python/>
- <http://ccbv.co.uk/>
- <https://insights.stackoverflow.com/survey/2019#technology>
- <https://insights.stackoverflow.com/survey/2020#technology-programming-scripting-and-markup-languages-all-respondents>

## Materialize

Materialize es un **Framework** dispuesto a facilitar el diseño web en todo tipo de escenarios, escritorio o móvil. Habiendo sido recomendada y después de haber pasado por un proceso de selección, su amplia gama de posibilidades y recursos, junto a su facilidad de uso, fue sin duda el detonante de su elección.

### Alternativas

Existen algunas alternativas como:

- Bootstrap
- Foundation

A diferencia de las alternativas, Materialize se basa en la normativa Material design. La creación de esta normativa por parte de Google hace que sea algo con lo que los usuarios están más familiarizados a usar. Es de esperar que su soporte.

Materialize no necesita jQuery para su uso, aunque puedes usarlo si así lo deseas.

Otro punto a su favor en comparación con Bootstrap o Foundation es su estética, la cual es muy atractiva, obteniendo un resultado homogéneo.

Una de los puntos negativos por parte de Bootstrap son los problemas entre versión y versión, ya que cambia la nomenclatura de sus elementos, cosa que con Materialize no ocurre.

A diferencia de sus alternativas, Materialize tiene una gran documentación, la cual, es más fácil de entender para alguien que acaba de empezar, así como una amplia cantidad de ejemplos visuales.

Gráficamente Materialize esta muy por encima de Bootstrap o Foundation, debido a su variedad de color y a sus componentes adicionales.

### Referencias



- <http://www.rafelsanso.com/bootstrap-vs-material-design/>
- <https://stackshare.io/stackups/foundation-vs-materialize#:~:text=You%20can%20quickly%20prototype%20and,framework%20based%20on%20Material%20Design.>
- [https://www.slant.co/versus/505/521/~foundation\\_vs\\_materialize](https://www.slant.co/versus/505/521/~foundation_vs_materialize)

## PostgreSQL

Es una base de datos SQL gratuita diseñada para ambientes de alto volumen. Tenemos experiencia en esta base de datos por varias asignaturas. Además dispone de una herramienta gráfica muy útil que facilita mucho su utilización.

### Alternativas

Existen otras alternativas como:

- MySQL
- MariaDB

MySQL es la opción más extendida, pero su licencia puede ser un problema, ya que está sujeta a las decisiones de Oracle. La segunda alternativa es MariaDB, un fork de MySQL de código abierto. La gran diferencia de PostgreSQL con esta última, como hemos comentado anteriormente, es que está pensada para alto volumen. Este ha sido el origen de nuestra elección.

## Heroku

Es una plataforma en nube gratuita que hemos utilizado anteriormente que está recomendada cuando se usa junto a Django y PostgreSQL.

### Alternativas

Existen algunas alternativas como:

- AWS
- Firebase

A diferencia de la versión gratuita de Firebase, la cual, es bastante simple, teniendo limitaciones en el espacio de almacenamiento y el número de usuarios simultáneos. Por tanto, para trabajar con ella de manera profesional, deberás pagar, en cambio, Heroku es gratuito para aplicaciones de poco consumo.

Comparándola con Heroku, AWS no es apto para personas principiantes, ya que la complejidad del mismo es bastante elevada.

En comparación con las alternativas expuestas, Heroku permite desplegar versiones, rollbacks, gestionar dependencias, como también desarrollar casi con cualquier lenguaje de programación: Java, Ruby, NodeJS...

#### Referencias

- <https://stackshare.io/stackups/firebase-vs-heroku>
- <https://rubygarage.org/blog/heroku-vs-amazon-web-services>

### Travis

Es una herramienta de integración continua que hemos usado en varias asignaturas que nos va a permitir ejecutar nuestros test y comprobar si han surgido fallos al añadir código al repositorio, además vamos a poder conectarlo con SonarCloud.

#### Alternativa

La segunda opción es Jenkins. El mayor ventaja que podemos destacar de Travis frente Jenkins es no que necesita ningún tipo de instalación, solo se configura un solo archivo y se puede utilizar a la perfección.

#### Referencias

\* <https://www.guru99.com/jenkins-vs-travis.html> \* <https://www.lambdatest.com/blog/travis-ci-vs-jenkins/>

### Sonarcloud

Nos va a permitir analizar el código de nuestra aplicación de forma automática conectándose a GitHub mediante Travis.

#### Alternativas

Existen algunas alternativas como:

- Codacy
- SonarQube

A diferencia de Sonarcloud Codacy no tiene soporte para Git privado o Azure DevOps git.

SonarQube tienen un mantenimiento de código más costoso, haciendo que sea necesario invertir más tiempo, así como generar "falsos positivos".

#### Referencias

- <https://stackshare.io/stackups/codacy-vs-codebeat-vs-sonarqube>
- <https://stackshare.io/stackups/codacy-vs-sonarqube>

#### Asciidoc

La forma más habitual de redactar sería en formato Word, pero veíamos necesario que toda la documentación estuviese junto al código. Por eso, elegimos Asciidoc, que nos permite redactar nuestros documentos de forma personalizada con un lenguaje sencillo y fácil de aprender.

#### Alternativa

Como segunda opción tenemos a Markdown, un lenguaje de marcado ligero más extendido que Asciidoc, por lo que hay muchas más herramientas que lo soportan. Al contrario que Asciidoc, carece de un estándar, por lo que al redactar algo con una ligera complejidad, cada herramienta puede interpretarlo de distinta forma.

## Herramientas

#### Adobe XD

Elegimos Adobe XD por la gran versatilidad en sus apartados para la realización de mockups para el proyecto. Dada a su gran facilidad de uso y sus múltiples plantillas y extensiones, su elección fue en nuestra opinión lo más recomendable. También nos ha permitido hacer mockups interactivos de forma sencilla.

#### Visual Studio Code

Es un editor de código desarrollado por Microsoft y que está siendo muy usado estos últimos años. Nos permite programar en múltiples lenguajes de forma personalizada con las numerosas extensiones que acompañan al editor. Además implementa una consola de depuración y está vinculado directamente con Git. Lo elegimos, aparte de los puntos anteriores, porque tenemos experiencia en él y es fácil de usar.

#### Clockify

Nos permite medir nuestro gasto temporal en nuestras tareas de manera sencilla. Existen muchas herramientas de este tipo, pero esta es la que más hemos utilizado, y nos sirve perfectamente para nuestro trabajo.

## Github

Poco que decir en este apartado, GitHub es la plataforma de control de versiones más utilizada. Dado su conocimiento de uso por parte de los miembros del grupo de trabajo, decidimos sin dudarle su elección.

## Jira

Jira es una herramienta de gran utilidad para equipos de todo tipo. Siendo también de gran ayuda para gestión de trabajo en todo tipo de casos, y en especial para el desarrollo software. Con todo esto y con el conocimiento de que se usa en entornos laborales de grandes empresas como Everis, decidimos utilizarla y aprender como funciona.

## Slack

Es una herramienta de comunicación que se usa en nuestro entorno profesional, además es mucho más formal que Whatsapp o Telegram, que se mezcla el trabajo con conversaciones informales.

# Gestión de código

## Gestión de ramas

Ramas por defecto:

- master
- dev

En la rama **dev** se incluirán todas las ramas de desarrollo con **pull requests**. La rama **master** solo se podrán añadir desde la rama **dev**, para incluir la release.

El resto de ramas seguirán el siguiente esquema:

**<codigo\_rama>/<codigo\_sprint>/<nombre\_tarea>**

Códigos de rama (<codigo\_rama>):

- feat/ (nueva feature)
- fix/ (bug fix)
- hotfix/ (bug fixes en producción)
- refactor/ (refactorización de código)
- docs/ (cambios de documentación)
- test/ (añadir o refactorizar tests)
- release/ (nueva release)

Códigos de sprint (<codigo\_sprint>):

- s0 (previo al comienzo del desarrollo de código)

- s1/s2/s3/s4 (sprints 1, 2, 3, 4 respectivamente)
- d (tras la finalización del desarrollo, realización de memoria)

Nombre de las tareas (<nombre\_tarea>): Nombre especificado en Jira, o abreviación. Por ejemplo, CU-07 Edición de profesores, o simplemente, CU-07. Se pueden añadir varias tareas separadas por &.

## Gestión de commits

Los commits deben seguir el siguiente patrón: [<tipo\_commit>] <nombre\_commit>. Además se puede añadir una descripción opcionalmente.

Tipo de commit (<tipo\_commit>):

- feat (nueva feature)
- fix (bug fix)
- refactor (refactorización de código)
- doc (cambios de documentación)
- test (añadir o refactorizar tests)
- release (nueva release)

## Pull requests

Para añadir funcionalidad a la rama **dev** se abrirá una pull request. El que haya desarrollado la funcionalidad se asignará a la pull request y asignará al compañero como revisor. Para que la pull request se de como válida debe de pasar el control y análisis de Travis y Sonarcloud, además del visto bueno del revisor.

Una vez que se haya hecho merge a dev, la rama se eliminará.

## Gestión de versiones

La especificación de la versión está marcada por el siguiente patrón: **X.Y.Z**, donde:

- **X**: Versión mayor.
- **Y**: Versión menor.
- **Z**: Versión en la que se solventan bugs de la versión anterior.

Versiones existentes:

- **0.1**: Versión que incorpora todo el sprint 1. Incluye la primera parte de la funcionalidad del administrador, así como el modelo de datos y el primer *populate*.
- **0.2**: Versión que incorpora todo el sprint 2. Incluye más funcionalidad del administrador.
- **0.2.1**: Corrección de algunos errores de la versión **0.2**.
- **0.3**: Versión que incorpora todo el sprint 3. Incluye la mayor parte de la funcionalidad del

profesor.

- **0.4:** Versión que incorpora todo el sprint 4. Incluye el resto de funcionalidad.

## Fuentes e Imágenes

A continuación hablaremos de la fuente escogida y de dónde hemos seleccionado las imágenes mostradas en la aplicación.

### Fuente

La opción a la hora de seleccionar una fuente ha sido seleccionado con el fin de evitarnos problemas jurídicos, así como, poderla usar en aplicaciones de fines lucrativos.

### Elección

Nos decantamos por la Fuente Ubuntu, la cual obtuvimos del siguiente link:

<https://design.ubuntu.com/font/>

La licencia de esta fuente es copyleft permite que las tipografías sean «usadas, estudiadas, modificadas y redistribuidas libremente» cumpliendo los términos estipulados en ella.

La documentación de la licencia se encuentra en la carpeta fonts, cuyo nombre es "LICENSE.txt".

### Imágenes

La opción a la hora de seleccionar las imágenes para la web fue difícil, ya que buscábamos un banco de imágenes que nos permitiera el uso comercial pensando en un hipotética uso de la misma y sin derechos de autor.

### Elección

Nos decantamos por la web <https://unsplash.com/>, la cual tiene un gran número de imágenes muy útiles, y una licencia que nos permitía lo nombrado anteriormente.

Con respecto a la licencia, dice lo siguiente:

*Unsplash photos are made to be used freely. Our license reflects that.*

- *All photos can be downloaded and used for free.*
- *Commercial and non-commercial purposes.*
- *No permission needed (though attribution is appreciated!).*

*What is not permitted*

- *Photos cannot be sold without significant modification.*
- *Compiling photos from Unsplash to replicate a similar or competing service.*

Para más información puede consultar dicha licencia en el siguiente enlace: <https://unsplash.com/>

## Implementación del cálculo

Existen muchas formas de realizar el cálculo de las notas, y dado que no podemos implementarlos todos, se implementará de tal forma que sea posible añadir nueva funcionalidad para el cálculo. Para que el profesor pueda modificar como se va a realizar el cálculo, se ha decidido almacenar esa información el grupo, así todos los alumnos tendrán el mismo método y no suponga un coste computacional muy alto. Se ha separado la selección del método del cálculo, se puede elegir un método de cálculo para la evaluación final y otro para las evaluaciones parciales, para que sea más flexible.

Todos los cálculos que se realizan en el proyecto están normalizados, por lo que los pesos son bastante flexibles y no está limitado de 0 a 10 o de 0 a 1, sino que está limitado de 0 a 999,99. Hemos pensado que el número es lo suficientemente grande para que no sea un impedimento en su uso.

Métodos de cálculo implementados:

- Evaluaciones finales:
  - Por competencias: Media ponderada de las competencias de nivel dos.
  - Por evaluaciones (Sin recuperación): Media ponderada de las evaluaciones parciales.
  - Por evaluaciones (Con recuperación): Si hay recuperaciones, la media ponderada de estas, sino media ponderada de las evaluaciones parciales.
- Evaluaciones parciales:
  - Por todas las actividades: Media ponderada de todas las actividades asociadas a la evaluación.
  - Por recuperaciones: Si hay recuperaciones, la media ponderada de estas, sino media ponderada de todas las actividades asociadas a la evaluación.

## Añadir nuevos métodos de cálculo

Los pasos son los siguientes:

- En primer lugar, añadir en *forms.py* el valor del método en base de datos y como se va a mostrar al usuario. Se deberá modificar `CHOICES_EVALUATION_TYPE_FINAL_EN` y `CHOICES_EVALUATION_TYPE_FINAL` para las evaluaciones finales y/o `CHOICES_EVALUATION_TYPE_PARTIAL_EN` y `CHOICES_EVALUATION_TYPE_PARTIAL` para las parciales.

```
# Choices
CHOICES_YES_NO = ((False, "No"), (True, "Sí"))
CHOICES_YES_NO_EN = ((False, "No"), (True, "Yes"))
CHOICES_LEVEL = (("1º", "1º"), ("2º", "2º"), ("3º", "3º"), ("4º", "4º"), ("5º", "5º"), ("6º", "6º"))
CHOICES_GRADE = (
    ("PrimarySchool", "Educación Primaria"),
    ("SecondaryEducation", "Educación Secundaria"),
    ("SixthForm", "Bachillerato"),
    ("FurtherEducation", "Grado Medio o Superior"),
    ("University", "Grado Universitario"))
CHOICES_GRADE_EN = (
    ("PrimarySchool", "Primary School"),
    ("SecondaryEducation", "Secondary Education"),
    ("SixthForm", "Sixth Form"),
    ("FurtherEducation", "Further Education"),
    ("University", "University"))
CHOICES_EVALUATION_TYPE_FINAL_EN = (
    ("BY_COMPETENCES", "By Competences"),
    ("BY_EVALUATION_NO_RECOVERY", "By Evaluations (No recovery)"),
    ("BY_EVALUATION_RECOVERY", "By Evaluations (Recovery)"))
CHOICES_EVALUATION_TYPE_FINAL = (
    ("BY_COMPETENCES", "Por Competencias"),
    ("BY_EVALUATION_NO_RECOVERY", "Por evaluaciones (Sin recuperación)"),
    ("BY_EVALUATION_RECOVERY", "Por evaluaciones (Con recuperación)"))
CHOICES_EVALUATION_TYPE_PARTIAL_EN = (("BY_ALL_ACTIVITIES", "By all Activities"), ("BY_RECOVERY_ACTIVITIES", "By Recovery Activities"))
CHOICES_EVALUATION_TYPE_PARTIAL = (("BY_ALL_ACTIVITIES", "Por todas las Actividades"), ("BY_RECOVERY_ACTIVITIES", "Por Recuperaciones"))
```

- En segundo lugar, implementar el nuevo método de cálculo en *services.py* en la clase *MarkService()*.
- En último lugar, modificar el método *calculate\_evaluation\_mark* de la clase *MarkService()* dentro de *services.py* para añadir en el condicional el nuevo método.

```
def calculate_evaluation_mark(self, evaluation: models.Evaluation, set_object: models.Set, student: models.Student) -> None:
    if evaluation.is_final:
        if set_object.evaluation_type_final == "BY_COMPETENCES":
            self.calculate_evaluation_mark_by_competences(evaluation = evaluation, set_object = set_object, student = student)
        elif set_object.evaluation_type_final == "BY_EVALUATION_NO_RECOVERY":
            self.calculate_evaluation_mark_by_partial_evaluations(evaluation = evaluation, set_object = set_object, student = student)
        elif set_object.evaluation_type_final == "BY_EVALUATION_RECOVERY":
            self.calculate_evaluation_mark_by_recovery_activities(evaluation = evaluation, set_object = set_object, student = student)
    else:
        if set_object.evaluation_type_partial == "BY_ALL_ACTIVITIES":
            self.calculate_evaluation_mark_by_all_activities(evaluation = evaluation, set_object = set_object, student = student)
        elif set_object.evaluation_type_partial == "BY_RECOVERY_ACTIVITIES":
            self.calculate_evaluation_mark_by_recovery_activities(evaluation = evaluation, set_object = set_object, student = student)
```

## Adaptabilidad del diseño en diferentes resoluciones

La resolución nativa de la aplicación es 1920x1080, dado que es la resolución en la que se ha desarrollado la aplicación. La visualización de la aplicación en resoluciones de escritorio son correctas, pero las resoluciones de dispositivos móviles han sido las más complicadas de realizar. El único dispositivo móvil en el que hemos trabajado en su adaptabilidad, es decir en su [diseño responsive](#), es el **iPhone X** porque es de los dispositivos más grandes, por lo tanto, el más fácil de implementar.

El tratamiento de los formularios en pantallas más pequeñas no ha sido complicado dado que [Materialize](#) nos daba las herramientas necesarias para hacerlo. El problema ha recaído en las listas, dado que hay mucha información repartida horizontalmente, por lo que hemos tenido que omitir información. En casi todas las lista se ha agrupado los botones con un cuadro emergente, es decir, un [modal](#), para aprovechar la mayor cantidad de espacio posible. Esto ha permitido que no se pierda funcionalidad.

En el resto de resoluciones móviles tampoco se pierde funcionalidad, pero cuanto menor es la pantalla peor se va a ver.



# Traducción

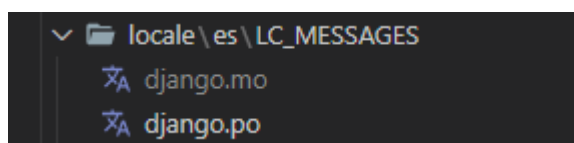
Para la traducción hemos utilizado **gettext** y la etiqueta de Django *trans*, los cuales nos permite traducir elementos concretos de las vistas y obtener información del idioma actual de la aplicación.

Para internacionalizar las vistas utilizamos la etiqueta *trans* sobre la palabra o conjunto de palabras que queremos traducir.

## Ejemplo de etiqueta de traducción

```
<h6 class="centered">
    {% trans "Set" %}: {{set_object.name}}
</h6>
```

Después debemos tener un archivo de configuración de cada idioma al que queramos traducir.



Para crear dicho archivo y compilarlo debemos utilizar los siguientes comando:

## Comando de creación de archivo de configuración

```
django-admin makemessages -l es
```

## Comando de compilación

```
py manage.py compilemessages
```

cambiando *es* (español) por el idioma correspondiente.

Sin embargo, para traducir los errores de formulario no es posible utilizar estas etiquetas, por que hemos implementado un método, valiéndose del método de gettext llamado *get\_language*, que hace saltar los errores de formulario en el idioma correcto.

## Método de error de validación

```
def raise_error(self, en_message: str, es_message: str):

    if get_language() == 'en':
        raise ValidationError(
            en_message)
    else:
        raise ValidationError(
            es_message)
```

# Instalación

## Prerequisitos

- Instalación de Python3: <https://www.python.org/downloads/>
- Instalación PostgreSQL: <https://www.postgresql.org/download/>

## (Opcional) Entorno virtual

*Comando de instalación del entorno virtual*

```
pip install virtualenvwrapper-win
```

*Comando de creación del entorno virtual*

```
mkvirtualenv <NombreEntorno>
```

*Comando de uso del entorno virtual*

```
workon <NombreEntorno>
```

## Configuración

- Creamos la base de datos en PostgreSQL.
- Creamos el archivo `local_settings.py`, lo rellenamos con la copia del contenido de `settings.py` y configuramos el apartado **DATABASES** para conectarlo a nuestra base de datos.
- Instalamos los requisitos e migramos el modelo.

*Comando de instalación de requisitos del proyecto*

```
pip install -r requirements.txt
```

*Comando de creación de las migraciones*

```
py manage.py makemigrations
```

*Comando para migrar a la base de datos*

```
py manage.py migrate
```

## Comandos usuales

Comando para poblar la base de datos

```
py manage.py runscript populate
```

Comando para ejecutar el servidor

```
py manage.py runserver
```

## V. Manual de usuario

## VI. Pruebas

### Pruebas de automáticas

En el desarrollo no se ha realizado pruebas automáticas dado que necesitan una mayor inversión de tiempo que las pruebas manuales, las cuales cumplen su función correctamente. Además no es lo suficientemente grande y complejo para necesitar la automatización de las pruebas. Hemos visto más importante invertir el tiempo en el desarrollo que en implementar pruebas automáticas.

### Pruebas de formulario

- Comprobamos que la vista este correctamente traducida.
- Verificamos la obligatoriedad de los campos enviandolo completamente vacío y rellenando uno a uno, comprobando si se muestra el mensaje de obligatoriedad.
- Comprobamos las restricciones de modelo, tanto longitud como rango, y vemos si se muestra el error correspondiente. Esto se hace en ambos idiomas.
- Lo enviamos correctamente y verificamos si se creado o modificado y que lleva a la vista correspondiente.
- Comprobamos si el botón de *Volver* lleva a la vista correspondiente.
- Verificamos que no se puede acceder al formulario si no tienes permiso (**GET**).
- Comprobamos si cambiando la url del formulario no se puede crear o modificar sin permiso (**POST**).
- Comprobamos el *reponsive*.

### Pruebas de borrado

- Comprobamos que la vista este correctamente traducida.
- Comprobamos si el botón de *Volver* lleva a la vista correspondiente.
- Lo enviamos correctamente y verificamos si se borrado y que lleva a la vista correspondiente.
- Verificamos que no se puede acceder a la opción de borrar si no tienes permiso (**GET**).

- Comprobamos si cambiando la url del formulario no se puede borrar sin permiso (**POST**).
- Comprobamos el *reponsive*.

## Pruebas de listado

- Comprobamos que la vista este correctamente traducida.
- Comprobamos que la vista este correctamente traducida.
- Comprobamos que no se pueda acceder a un listado sin permiso mediante url.
- Comprobamos si el botón de *Volver* lleva a la vista correspondiente.
- Comprobamos el *reponsive*.

## VII. Comparativas con otras alternativas

## VIII. Conclusiones y desarrollos futuros

## Glosario

- **Actividad:** Tarea que crea un profesor para evaluar una o varias competencias de un alumno. Pueden ser pruebas, exámenes, ejercicios de clases, salidas extraescolares, trabajos grupales, etc.
- **Competencia concreta:** Competencia de nivel 1. Competencia muy delimitada, asociada a una asignatura, que se puede puntuar directamente.
- **Competencia específica:** Competencia de nivel 2. Competencia estrechamente relacionada con una asignatura. Se calcula a partir de las competencias concretas.
- **Competencia general:** Competencia de nivel 3. Competencia de alto nivel que engloba varias asignaturas y habilidades.
- **Diseño responsive:** Consiste en configurar los elementos de la vista de la web para adaptarlos para diferentes resoluciones.
- **Ejercicio:** Parte de la que se componen las actividades. Por ejemplo, en un examen de matemáticas, puede ser que un ejercicio sea realizar una ecuación.
- **Grupo:** Conjunto de alumnos asignados a una asignatura de un nivel específico. Por ejemplo, una clase de 3ºB de matemáticas.
- **Modal:** Es un cuadro emergente que se muestra por encima de la pantalla.
- **Framework:** Es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática.

## Casos prácticos de competencias

Niveles	Nombre en la aplicación	Enseñanza primaria/ESO/Bachillerato	Títulos universitarios
Nivel 1	Competencias concretas	Estándares de aprendizaje	Criterios del profesor
Nivel 2	Competencias específicas	Criterios de evaluación	Competencias generales y específicas
Nivel 3	Competencias generales	Competencias clave	Competencias básicas

## Bibliografía

- [https://www.boe.es/diario\\_boe/txt.php?id=BOE-A-2014-2222](https://www.boe.es/diario_boe/txt.php?id=BOE-A-2014-2222)
- <https://www.boe.es/boe/dias/2020/12/30/pdfs/BOE-A-2020-17264.pdf>
- <https://www.us.es/estudiar/que-estudiar/oferta-de-gradados/grado-en-ingenieria-informatica-ingenieria-del-software>
- <https://www.us.es/estudiar/que-estudiar/oferta-de-gradados/grado-en-pedagogia>
- <http://www.juntadeandalucia.es/educacion/descargasrecursos/curriculo-primaria/pdf/PDF/textocompleto.pdf>
- [https://www.boe.es/diario\\_boe/txt.php?id=BOE-A-2015-37](https://www.boe.es/diario_boe/txt.php?id=BOE-A-2015-37)
- <https://docs.djangoproject.com/en/3.2/topics/i18n/translation/>
- [https://www.agenciatributaria.es/AEAT.internet/Inicio/Segmentos/Empresas\\_y\\_profesionales/Empresas/Impuesto\\_sobre\\_Sociedades/Periodos\\_impositivos\\_a\\_partir\\_de\\_1\\_1\\_2018/Base\\_imponible/Amortizacion/Tabla\\_de\\_coeficientes\\_de\\_amortizacion\\_lineal\\_.shtml](https://www.agenciatributaria.es/AEAT.internet/Inicio/Segmentos/Empresas_y_profesionales/Empresas/Impuesto_sobre_Sociedades/Periodos_impositivos_a_partir_de_1_1_2018/Base_imponible/Amortizacion/Tabla_de_coeficientes_de_amortizacion_lineal_.shtml)
- <https://www.adobe.com/es/creativecloud/plans.html?filter=design&plan=individual&promoid=TKZTLDFL&mv=other>
- <https://www.heroku.com/pricing>