



Mirmidon Company

Proyecto realizado por:



Cristina Diez



Alejandro Huedo



Andres Perez

ÍNDICE:

Control de cambios.....	2
Introducción.....	3
Descripción Funcional.....	4 y 5
Diagramas.....	6, 7 y 8
Tablas.....	9
Normalización.....	10 y 11
Código SQL.....	12
Equipo.....	13
Enlaces.....	14

Control de cambios:

<i>Versión</i>	<i>Cambios realizados</i>
<i>V1.0</i>	Documento inicial con el diagrama Entidad-Relación.
<i>V1.1</i>	Desarrollo del código e implementación del Diagrama de Clases.

Introducción:

El objetivo de este documento se van a especificar las características que tiene el proyecto ([Mirmidon's Project](#)) así como el proceso y versiones del mismo. Será actualizado a medida que se realicen los diferentes cambios y se introduzcan las nuevas versiones del proyecto. También se adjuntarán los distintos esquemas y diagramas que ayuden a la comprensión de la funcionalidad del programa.

Descripción funcional:

La aplicación gestionará los proyectos integradores de los diferentes ciclos: DAM, ASIR, DAW y “Animación 3D, juegos y entornos interactivos”.

Permitirá dar de alta los proyectos en su ciclo correspondiente y con los alumnos de dicho proyecto. También será posible dar de baja los proyectos y modificarlos.

La aplicación tendrá una primera pantalla de login donde el usuario debe elegir si quiere entrar como alumno; en este caso únicamente podrá acceder a las secciones de consultas de la aplicación y no podrá modificar nada, o podrá elegir la opción de entrar como profesor y tener acceso total a la aplicación.

La ventana principal poseerá un menú en la parte superior con cuatro secciones diferenciadas, Alumnos, Ciclos, Proyectos y Registrar; además de un mensaje de bienvenida y un botón para poder “desloguearte” en cualquier momento en caso de que hayas entrado como profesor.

En la pestaña de Alumnos tendrá los 3 accesos básicos: añadir, eliminar y modificar; cualquiera de los 3 nos dará acceso a una nueva sección donde podremos realizar la acción que hayamos seleccionado.

La pestaña ciclos seguirá el mismo esquema que el anterior, en cambio proyectos, además de poseer lo mismo que los dos anteriores, tendrá una sección llamada consulta donde podremos básicamente consultar los diferentes proyectos registrados así como ver los alumnos que hay en ellos.

Por último habrá una sección, solo accesible a profesores, donde podrán registrar nuevos profesores en la base de datos.

Seguiremos el siguiente esquema:

- Ciclos:
 - Alta
 - Baja
 - Modificación
- Alumnos:
 - Alta
 - Baja
 - Modificación
- Proyectos:
 - Alta
 - Baja
 - Modificación
 - Consulta
 - Listado
- Registrar
 - Registrar Profesor.

Diagramas:

Diagrama de Entidad-Relación:

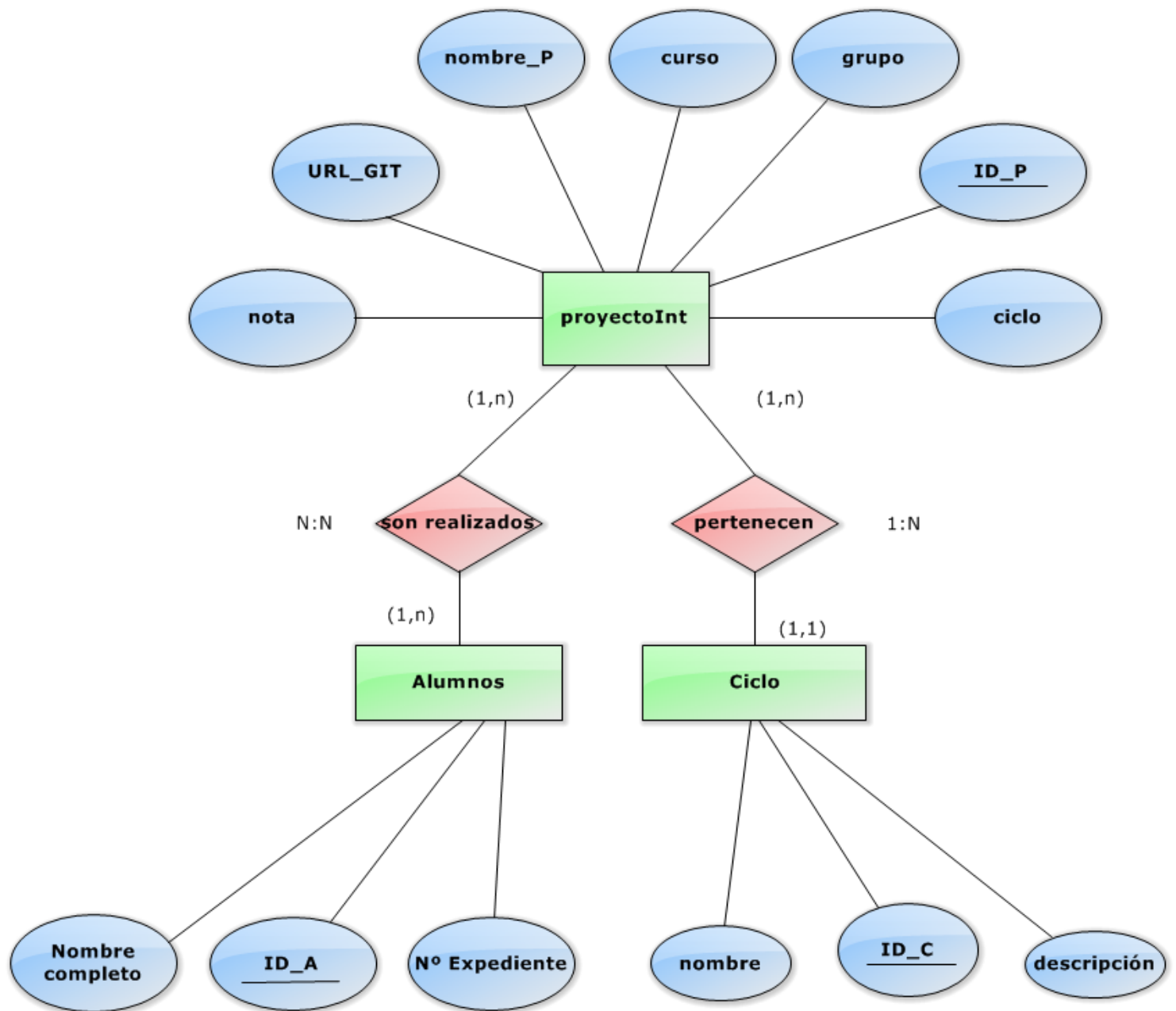


Diagrama Caso de Uso:

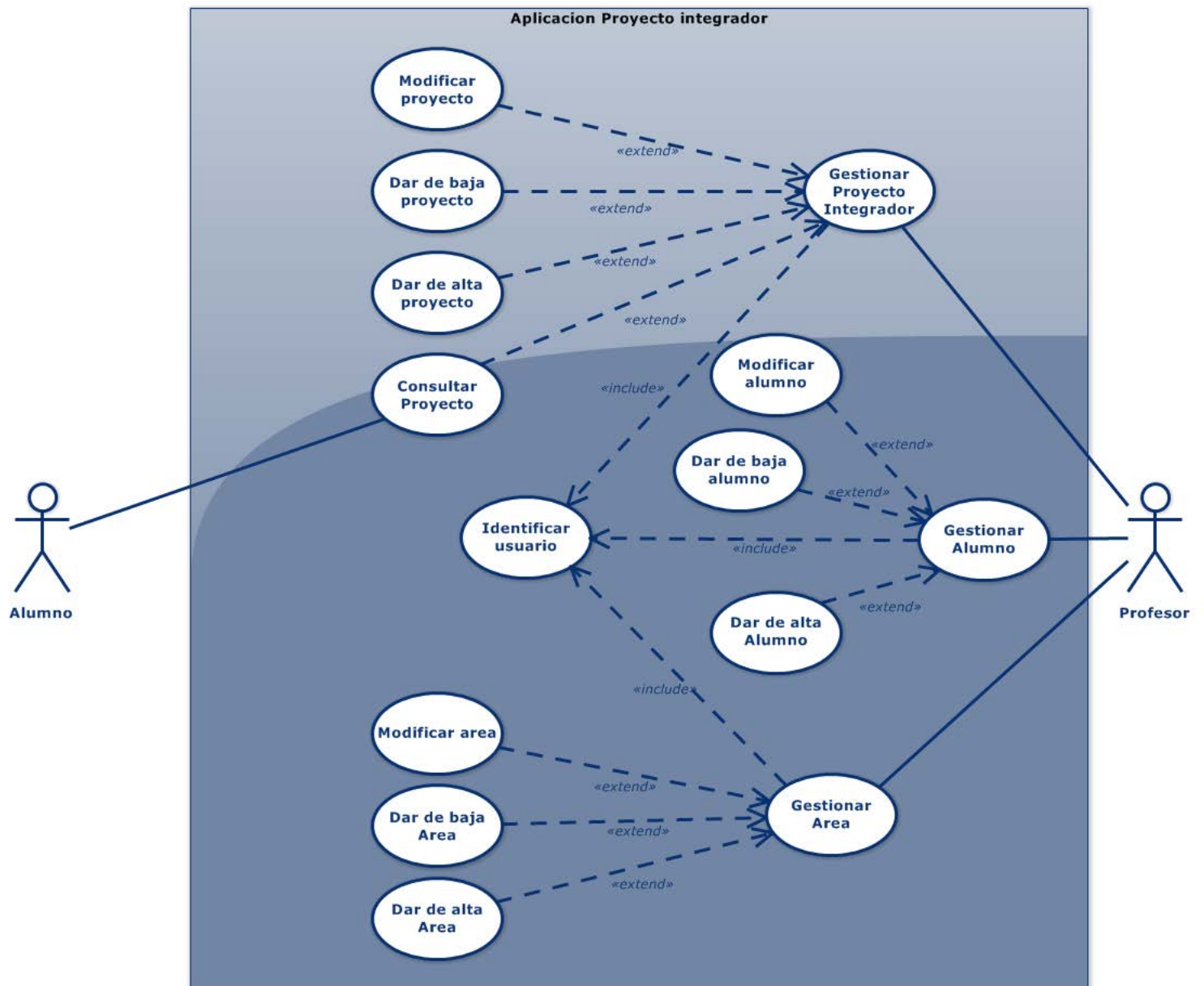
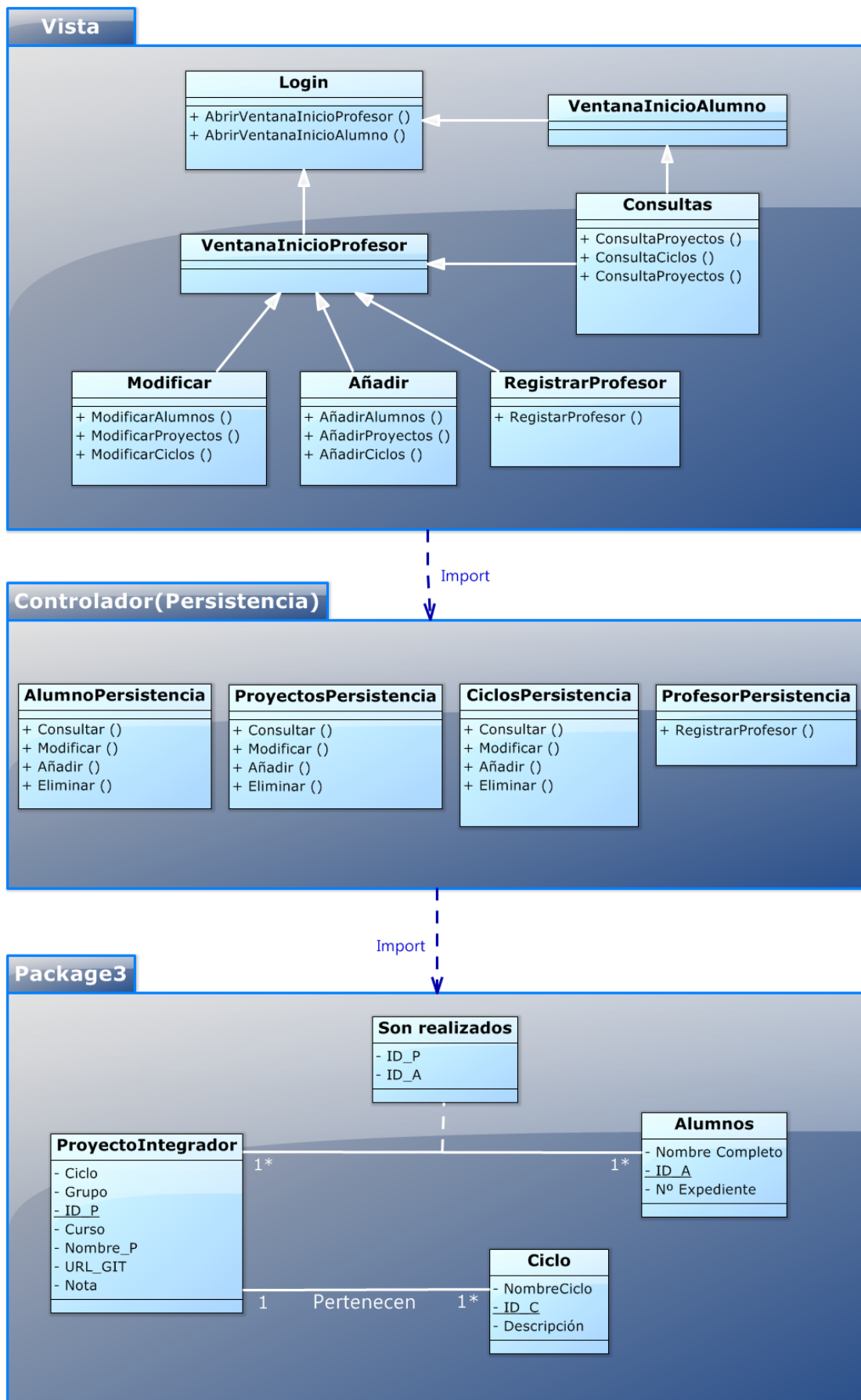


Diagrama de Clases(MVC):



Tablas:

- **ProyectoInt** {ID_P, nombre_P, curso, URL_GIT, nota, grupo , ciclo} La clave ajena ciclo hace referencia a la clave primaria de la tabla ciclo.
- **Alumnos** {ID_A, nºexpediente, nombre completo}
- **Ciclo** { ID_C, nombre, descripción}
- **Son realizados** {Proyecto, Alumno} La clave ajena Proyecto hace referencia a la clave primaria ID_P de la tabla proyectoInt. La clave ajena Alumno hace referencia a la clave primaria ID_A de la tabla Alumnos.

Normalización:

Primera Forma Normal FN1

- La Primera Forma Normal Establece que las columnas repetidas deben eliminarse y colocarse en tablas separadas.

Proyecto Integrador	Alumnos	Ciclo
ID_P	ID_A	ID_C
URL_Git	Nombre Completo	Nombre
Nombre_P	Nº Expediente	Descripción
Curso		
Grupo		
Nota		
Ciclo		

Esta formalizada en 1FN porque cada atributo tiene un único valor.

Segunda Forma Normal FN2

- En la Segunda Forma Normal se deben eliminar y separar las dependencias parciales dentro de sus propias tablas. Una dependencia parcial es un término que describe a aquellos datos que no dependen de la clave de la tabla para identificarlos.

Proyecto Integrador	Alumnos	Ciclo
ID_P	ID_A	ID_C
URL_Git	Nombre Completo	Nombre
Nombre_P	Nº Expediente	Descripción
Curso		
Grupo		
Nota		
Ciclo		

Esta formalizada en 2FN porque los atributos no dependen de otras claves.

Tercera Forma Normal FN3

- La tercera forma normal todos los valores deben identificarse únicamente por la clave. La independencia de las columnas implica que debe poder cambiar cualquier columna que no sea clave sin que ninguna otra columna resulte afectada.

Proyecto Integrador	Alumnos	Ciclo	Realizan
ID_P	ID_A	ID_C	ID_P
URL_Git	Nombre Completo	Nombre	ID_A
Nombre_P	Nº Expediente	Descripción	
Curso			
Grupo			
Nota			
Ciclo			

Añadimos la tabla Realizan con los atributos ID_P de Proyecto y ID_A de alumnos para formalizar en FN3.

Código SQL:

Alumno:

Añadir un alumno:

```
insert into alumnos (nombre, numExpediente) values (?,?)
```

Eliminar alumno donde se introduzca el número de expediente de este:

```
Delete from alumnos where numExpediente
```

Actualizar un alumno:

```
update alumnos set nombre = ?,numExpediente = ? where ID_A = ?
```

Realizar una consulta de alumnos

```
select ID_A from alumnos where numExpediente = ?
```

Ciclo:

Añadir un ciclo:

```
insert into ciclos (nombre, descripcion) values (?,?)
```

Borrar un ciclo escogido por el nombre:

```
Delete from ciclos where nombre =?
```

Actualizar un alumno:

```
update ciclos set nombre = ?,descripcion = ? where id_c = ?
```

Cargar todos los ciclos:

```
Select * from ciclos
```

Consultar los ciclos por ID:

```
select id_c from ciclos where nombre = ?
```

Proyecto:

Añadir un proyecto:

```
insert into proyectos (nombre,url,nota,año,curso,grupo,ciclo) values  
(?,?,?, ?, ?, ?, ?)
```

Para crear la relación entre las tablas proyecto y alumnos lo haremos mediante la tabla se_realizan:

```
insert into se_realizan (proyecto, alumno) values (?,?)
```

Borrar un proyecto:

Primero eliminamos las referencias al proyecto en la tabla se_realizan

```
Delete from se_realizan where proyecto =?
```

Y después en la tabla proyectos buscando por el nombre:

```
Delete from proyectos where nombre = ?
```

Cargar todos los proyectos:

```
Select * from proyectos
```

Consultar los proyectos por ID:

```
select id_p from proyectos where nombre = ?
```

Equipo:

Scrum Master:

El scrum master será Andrés, será el encargado de coordinar las diferentes tareas así como llevar a cabo la división de las tareas y encargarse del cumplimiento de los plazos establecidos para los diferentes sprints.

Cloud Master:

El cloud master sera Alejandro, será el encargado de mantener actualizado github con las nuevas versiones de la aplicación y mantener waffle actualizado.

Designer:

El designer será Cristina , se encargará del maquetamiento de la aplicación, es la persona más creativa del equipo y realizará los documentos del proyecto.

Developers:

Los 3 integrantes del equipo actuarán como developers de la aplicación desarrollando el código en java.

Enlaces:

GitHub: <https://github.com/Andres-93/ProyectoIntegrador1-DAM>

Waffle: <https://waffle.io/Andres-93/ProyectoIntegrador1-DAM>

Vídeo: <https://www.youtube.com/watch?v=ERIJf1EJ0hQ>