

# **Universidad ORT**

## **Facultad de Ingeniería**

### **Diseño de aplicaciones 1**

#### **Obligatorio 2**

**Prof. Gonzalo Laguna**

**Prof. Gabriel Pifarretti**

**Diego Balbi 198685**

**Joaquin Oribe 180521**

## Índice

1.Introducción .....	5
1.1 Producto.....	5
1.2 Propósito de la documentación.....	5
2. Descripción general .....	5
2.1 Objetivo.....	5
2.2 Alcance del producto .....	5
2.3 Usuarios potenciales.....	5
2.4 Actores .....	5
2.5 Funcionalidades del producto .....	6
2.6 Funcionalidades no implementadas .....	7
3.0 Especificación de requerimientos. ....	8
3.1 Requerimientos Funcionales. ....	8
3.1.1 RF1: Dar alta Colaborador .....	8
3.1.2 RF2: Dar de alta Administrador.....	8
3.1.3 RF3: Modificar Usuario .....	8
3.1.4 RF4: Dar de baja a Usuario .....	8
3.1.5 RF5: Dar alta Equipo .....	8
3.1.6 RF6: Modificación de Equipo .....	8
3.1.7 RF7: Dar de baja Equipo .....	8
3.1.8 RF8: Informe sobre pizarrones creados por equipo.....	8
3.1.9 RF9: Informe sobre comentarios resueltos por usuario.....	8
3.1.10 RF10: Dar de alta pizarrón .....	8
3.1.11 RF11: Modificar Pizarrón .....	9
3.1.12 RF12: Dar baja pizarrón .....	9
3.1.13 RF13: Agregar elemento al pizarrón .....	9
3.1.14 RF14: Modificar elemento de un pizarrón.....	9
3.1.15 RF15: Agregar comentario.....	9
3.1.16 RF16: Eliminar elemento de un pizarrón .....	9
3.1.17 RF17: Imprimir pizarrón.....	9
3.1.18 RF18: Conexión de elementos. ....	9
3.1.19 RF19: Ranking de usuarios.....	9
3.1.20 RF20: Reiniciar puntuación.....	9

3.1.21 RF21: Puntuación de usuarios.....	9
3.2 Requerimientos No Funcionales.....	10
3.2.1 RNF1 Lenguaje .....	10
3.2.2 RNF2 Mantenibilidad.....	10
3.2.3 RNF3 Metodología TDD .....	10
3.2.4 RNF4 Repositorio GIT .....	10
3.2.5 RNF5 Usabilidad .....	10
3.2.6 RNF6 Persistencia de datos .....	10
4. Casos de Uso .....	11
4.1 RF1 Alta Colaborador .....	11
4.2 RF2 Alta Administrador .....	12
4.3 RF3 Alta Colaborador .....	13
4.4 RF4 Baja Usuario .....	13
4.5 RF5 Dar alta de Equipo .....	14
4.6 RF6 Modificación de Equipo .....	15
4.7 RF7 Baja de equipo .....	16
4.8 RF8 Informes sobre pizarrones creados por equipos .....	16
4.9 RF9 Informe sobre comentarios resueltos por usuario .....	17
4.10 RF10 Alta Pizarrón.....	18
4.11 RF11 Modificar Pizarrón.....	19
4.12 RF12 Baja Pizarrón .....	20
4.13 RF13 Agregar elemento a pizarrón .....	20
4.14 RF14 Modificar elemento de un pizarrón .....	21
4.15 RF15 Agregar Comentario .....	21
4.16 RF16 Baja Elemento .....	22
4.17 RF17 Imprimir pizarrón .....	22
4.18 RF18 Conexión de elementos. ....	23
4.19 RF19 Ranking de usuarios.....	24
4.20 RF20 Reinicio de puntaje.....	24
4.21 RF21 Selección de puntajes. ....	25
7. Justificación de Diseño.....	25
7.1 Justificación de Diseño de programa .....	25
7.2 Utilización de patrones.....	27

7.2.1 Patron Facade .....	27
7.2.2 Patron Strategy .....	27
<b>8.</b> Evidencia de TDD. ....	28
<b>9.</b> Evidencia de Clean Code .....	29
<b>10.</b> Diagrama de clases .....	30
<b>11.</b> Cobertura del código .....	31
<b>12.</b> Pruebas.....	31
<b>13.</b> Diagrama de interacción. ....	32
13.1 Paquete persistence, Logic e Interface .....	32
13.2 Paquete Logic y Utility.....	32
<b>14.</b> Datos de Prueba .....	33

# 1.Introducción

## 1.1 Producto

El producto es un sistema de pizarrones interactivos multiusuario.

## 1.2 Propósito de la documentación

Durante esta documentación se informará sobre el proceso, servicios y restricciones del sistema de pizarrones interactivos multiusuario. Su propósito es facilitar el uso y demostrar el proceso de la construcción y decisiones del mismo.

# 2. Descripción general

## 2.1 Objetivo

El objetivo del sistema es generar un ambiente de estudio interactivo entre diferentes usuarios sencillo y útil, para poder estar al tanto a la evolución en las herramientas de colaboración para trabajos en conjunto.

## 2.2 Alcance del producto

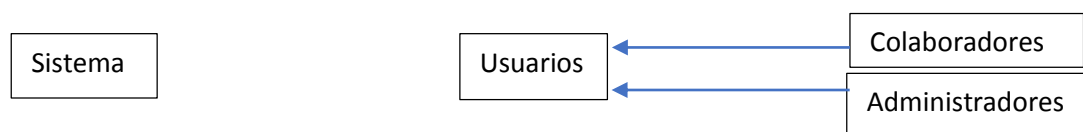
El sistema brinda un servicio de estudio interactivo con pizarrones que permite un fluido trabajo en equipo mediante su método de comentarios, los usuarios pueden realizar consultas a sus compañeros de su equipo por ayuda y los mismos pueden resolver su problema fácilmente. Este sistema es único en su campo.

El sistema admite tanto cuadros de texto como imágenes, permitiendo así más dinamismo y mejor comunicación en el pizarrón del equipo.

## 2.3 Usuarios potenciales

Aquellas personas interesadas en realizar trabajos en equipo mediante una plataforma multiusuario.

## 2.4 Actores



Los actores son:

- Sistema
- Usuarios:
  - Colaboradores
  - Administradores

## 2.5 Funcionalidades del producto

1. El usuario administrador podrá dar de alta a nuevos colaboradores al sistema.
2. El usuario administrador podrá dar de alta a nuevos administradores al sistema.
3. El usuario administrador podrá modificar las contraseñas de los usuarios que hacen el pedido de cambio.
4. El usuario administrador podrá dar de baja a usuarios existentes.
5. El usuario administrador podrá dar de alta a nuevos equipos al sistema.
6. El usuario administrador podrá modificar los datos de los equipos en el sistema.
7. El usuario administrador podrá dar de baja a equipos existentes del sistema.
8. El usuario administrador tendrá acceso a informes de los pizarrones creados por equipo.
9. El usuario administrador tendrá acceso a informes de los comentarios resueltos por usuario.
10. Los usuarios podrán crear pizarrones dentro del equipo que pertenecen.
11. Los usuarios podrán modificar información de los pizarrones a los que pertenecen.
12. Los usuarios administradores o el colaborador que creo el pizarrón pueden dar de baja el mismo.
13. Los usuarios podrán agregar elementos (cuadro de texto o imagen) a un pizarrón.
14. Los usuarios podrán modificar los elementos de un pizarrón, ya sea ubicación, tamaño o contenido.
15. Los usuarios podrán agregar comentarios a un elemento y luego podrán indicar si el mismo fue resuelto.
16. Los usuarios podrán eliminar elementos de un pizarrón.
17. Los usuarios podrán imprimir un pizarrón a un PDF o PNG.
18. Los usuarios podrán conectar elementos entre sí.
19. Los usuarios accederán al ranking de cada equipo.
20. Los usuarios administradores podrán resetear los puntajes de los usuarios.
21. Los administradores podrán establecer puntajes de experiencia a usuarios.
  - 21.1. Puntaje por crear pizarrón.
  - 21.2. Puntaje por eliminar pizarrón.
  - 21.3. Puntaje por agregar un elemento.
  - 21.4. Puntaje por agregar un comentario.
  - 21.5. Puntaje por marcar comentario como resuelto.

## 2.6 Funcionalidades no implementadas

8. El usuario administrador tendrá acceso a informes de los pizarrones creados por equipo.
9. El usuario administrador tendrá acceso a informes de los comentarios resueltos por usuario.
10. Los usuarios podrán crear pizarrones dentro del equipo que pertenecen.
11. Los usuarios podrán modificar información de los pizarrones a los que pertenecen.
12. Los usuarios administradores o el colaborador que creo el pizarrón pueden dar de baja el mismo.
13. Los usuarios podrán agregar elementos (cuadro de texto o imagen) a un pizarrón.
14. Los usuarios podrán modificar los elementos de un pizarrón, ya sea ubicación, tamaño o contenido.
15. Los usuarios podrán agregar comentarios a un elemento y luego podrán indicar si el mismo fue resuelto.
16. Los usuarios podrán eliminar elementos de un pizarrón.
17. Los usuarios podrán imprimir un pizarrón a un PDF o PNG.
18. Los usuarios podrán conectar elementos entre sí.
21. Los administradores podrán establecer puntajes de experiencia a usuarios.
  - 21.1. Puntaje por crear pizarrón.
  - 21.2. Puntaje por eliminar pizarrón.
  - 21.3. Puntaje por agregar un elemento.
  - 21.4. Puntaje por agregar un comentario.
  - 21.5. Puntaje por marcar comentario como resuelto.

Por razones de tiempo el equipo decidió no realizar pizarrón. Se consideró más valioso concentrar nuestros esfuerzos en hacer un trabajo prolijo acorde a los estándares pedidos, que intentar de hacer todo y mal.

Con respecto a puntuación, se mal interpreto lo pedido, y se trabajó alrededor de la malinterpretación. Nos dimos cuenta el día anterior de la entrega del error, y se decidió eliminar el progreso en el mismo.

## 3.0 Especificación de requerimientos.

### 3.1 Requerimientos Funcionales.

#### 3.1.1 RF1: Dar alta Colaborador

**Descripción:** Dar de alta a un nuevo usuario colaborador, se pide: nombre, apellido, mail, contraseña y fecha de nacimiento.

#### 3.1.2 RF2: Dar de alta Administrador

**Descripción:** Dar de alta a un nuevo usuario administrador, se pide: nombre, apellido, mail, contraseña y fecha de nacimiento.

#### 3.1.3 RF3: Modificar Usuario

**Descripción:** Se modifica la contraseña del usuario que ha requerido el cambio, se pide que ingrese su contraseña actual y la nueva contraseña.

#### 3.1.4 RF4: Dar de baja a Usuario

**Descripción:** Se da de baja el usuario indicado del sistema.

#### 3.1.5 RF5: Dar alta Equipo

**Descripción:** Se da de alta un equipo, se pide: nombre del equipo, descripción, su cantidad máxima de usuarios, y que usuarios van a integrarlo.

#### 3.1.6 RF6: Modificación de Equipo

**Descripción:** Se modifica la información del equipo sea su nombre, descripción, máxima cantidad de usuarios.

#### 3.1.7 RF7: Dar de baja Equipo

**Descripción:** Se da de baja el equipo seleccionado del sistema.

#### 3.1.8 RF8: Informe sobre pizarrones creados por equipo

**Descripción:** Se devuelve una vista rápida de la cantidad de pizarrones creados por equipo, filtrando por fecha de creación y equipos. Se muestran: Equipo creador, fecha de creación, fecha de última modificación, cantidad de elementos en el pizarrón.

#### 3.1.9 RF9: Informe sobre comentarios resueltos por usuario

**Descripción:** Se devuelve una vista rápida de los comentarios resueltos por usuario, filtrando por fecha de creación de comentario, resolución del mismo, de usuario creado y del usuario que lo resuelve. Se muestran: fecha de creación de comentario, usuario creador, usuario que resuelve, pizarrón al que pertenece el elemento y fecha de resuelto el comentario.

#### 3.1.10 RF10: Dar de alta pizarrón

**Descripción:** Se da de alta un pizarrón, se pide: su nombre, equipo al que pertenece, una breve descripción, un ancho y un alto.



#### 3.1.11 RF11: Modificar Pizarrón

**Descripción:** Se permite modificar la información del pizarrón, ya sea su nombre o descripción.

#### 3.1.12 RF12: Dar baja pizarrón

**Descripción:** Un usuario administrador o el creador del pizarrón podrá dar de baja el pizarrón.

#### 3.1.13 RF13: Agregar elemento al pizarrón

**Descripción:** El usuario perteneciente al pizarrón podrá ingresar imágenes o campos de texto al mismo. Se puede hacer mediante un drag and drop.

#### 3.1.14 RF14: Modificar elemento de un pizarrón

**Descripción:** El usuario perteneciente al pizarrón podrá modificar la información de los elementos pertenecientes al mismo.

#### 3.1.15 RF15: Agregar comentario

**Descripción:** Los usuarios pertenecientes a un pizarrón podrán agregar un comentario a un elemento, para hacer esto deberán ingresar la descripción del comentario.

#### 3.1.16 RF16: Eliminar elemento de un pizarrón

**Descripción:** Los usuarios pertenecientes a un pizarrón podrán eliminar elementos pertenecientes al mismo.

#### 3.1.17 RF17: Imprimir pizarrón

**Descripción:** Se podrá imprimir el pizarrón a un formato PDF o .PNG.

#### 3.1.18 RF18: Conexión de elementos.

**Descripción:** Se conectan dos elementos como uno nuevo.

#### 3.1.19 RF19: Ranking de usuarios.

**Descripción:** Se mostrará un ranking de los usuarios en el equipo seleccionado.

#### 3.1.20 RF20: Reiniciar puntuación.

**Descripción:** Se reiniciará el puntaje de los usuarios en el equipo seleccionado.

#### 3.1.21 RF21: Puntuación de usuarios.

**Descripción:** Se asignarán diferentes puntajes a diferentes acciones dentro de cada equipo a usuarios, por realizar dichas acciones.

## 3.2 Requerimientos No Funcionales

### 3.2.1 RNF1 Lenguaje

**Descripción:** El lenguaje a utilizar será C#.

### 3.2.2 RNF2 Mantenibilidad

**Descripción:** El sistema será construido siguiendo los estándares de Clean Code, haciendo que el código sea fácil de entender y mantener.

### 3.2.3 RNF3 Metodología TDD

**Descripción:** El sistema se desarrollará en su totalidad con la metodología TDD (Test Driven Development).

### 3.2.4 RNF4 Repositorio GIT

**Descripción:** Se utilizará el control de versionado de repositorio GIT de GitHub.

### 3.2.5 RNF5 Usabilidad

**Descripción:** Se dispondrá de un interfaz amigable y fácil de usar.

### 3.2.6 RNF6 Persistencia de datos

**Descripción:** Se hará persistencia en base de datos SQL, mediante Entity Framework.

## 4. Casos de Uso

### 4.1 RF1 Alta Colaborador

<b>CU1</b>	
<b>Descripción:</b> Dar de alta a un nuevo usuario colaborador, se pide: nombre, apellido, mail, contraseña y fecha de nacimiento.	
<b>Actores:</b> Usuario Administrador - Sistema	
<b>Pre-Condición:</b> -	
<b>Post-Condición:</b> Ingresa el usuario al sistema	
<b>Curso Básico</b>	
<b>Usuario Administrador</b>	<b>Sistema</b>
1. El administrador selecciona la opción de registrar usuario.	
2. El administrador ingresa los datos correspondientes (nombre, apellido, mail, contraseña y fecha de nacimiento).	
3. El administrador aprieta el botón registrar	
	4. Se verifican los datos
	5. El sistema registra el usuario
	6. fin
<b>Curso Alternativo</b>	
3.1 Se aprieta el botón cancelar. 3.1.2 Se cierra la ventana. 4.1. Se Ingresa un nombre invalido. 4.1.1 Se notifica el error. 4.1.2 Vuelve al paso 2. 4.2 Se ingresa un apellido invalido. 4.2.1 Se notifica el error 4.2.2 Vuelve al paso 2 4.3 Se ingresa un mail invalido. 4.3.1 Se notifica el error 4.3.2 Vuelve al paso 2 4.4 Se ingresa una contraseña invalida. 4.4.1 Se notifica el error 4.4.2 Vuelve al paso 2 4.5 Se ingresa una fecha de nacimiento invalida. 4.5.1 Se notifica el error 4.5.2 Vuelve al paso 2 4.6 El usuario ya está registrado 4.6.1 Se notifica el error 4.6.2 Vuelve al paso 2	

## 4.2 RF2 Alta Administrador

<b>CU2</b>	
<b>Descripción:</b> Dar de alta a un nuevo usuario administrador, se pide: nombre, apellido, mail, contraseña y fecha de nacimiento.	
<b>Actores:</b> Usuario Administrador - Sistema	
<b>Pre-Condición:</b> -	
<b>Post-Condición:</b> Ingresa el usuario al sistema	
<b>Curso Básico</b>	
<b>Usuario Administrador</b>	<b>Sistema</b>
1. El administrador selecciona la opción de registrar usuario.	
2. El administrador ingresa los datos correspondientes (nombre, apellido, mail, contraseña y fecha de nacimiento).	
3. El administrador aprieta el botón registrar	
	4. Se verifican los datos
	5. El sistema registra el usuario
	6. fin
<b>Curso Alternativo</b>	
<p>3.1 Se aprieta el botón cancelar.</p> <p>3.1.2 Se cierra la ventana.</p> <p>4.1. Se Ingresa un nombre invalido.</p> <p>4.1.1 Se notifica el error.</p> <p>4.1.2 Vuelve al paso 2.</p> <p>4.2 Se ingresa un apellido invalido.</p> <p>4.2.1 Se notifica el error</p> <p>4.2.2 Vuelve al paso 2</p> <p>4.3 Se ingresa un mail invalido.</p> <p>4.3.1 Se notifica el error</p> <p>4.3.2 Vuelve al paso 2</p> <p>4.4 Se ingresa una contraseña invalida.</p> <p>4.4.1 Se notifica el error</p> <p>4.4.2 Vuelve al paso 2</p> <p>4.5 Se ingresa una fecha de nacimiento invalida.</p> <p>4.5.1 Se notifica el error</p> <p>4.5.2 Vuelve al paso 2</p> <p>4.6 El usuario ya está registrado</p> <p>4.6.1 Se notifica el error</p> <p>4.6.2 Vuelve al paso 2</p>	

#### 4.3 RF3 Alta Colaborador

<b>CU3</b>	
<b>Descripción:</b> Se modifica la contraseña del usuario que ha requerido el cambio, se pide que ingrese su contraseña actual y la nueva contraseña.	
<b>Actores:</b> Usuario Administrador - Sistema	
<b>Pre-Condición:</b> -	
<b>Post-Condición:</b> Se modifican los datos del usuario	
<b>Curso Básico</b>	
<b>Usuario Administrador</b>	<b>Sistema</b>
1. El administrador selecciona la opción de modificar usuario.	
2. El administrador ingresa la contraseña nueva.	
3. El administrador aprieta el botón modificar.	
	4. Se verifican los datos
	5. El sistema modifica el usuario
	6. fin
<b>Curso Alternativo</b>	
3.1 Se aprieta el botón cancelar.	
3.1.2 Se cierra la ventana.	
4.1. Se Ingresa una contraseña invalida.	
4.1.1 Se notifica el error.	
4.1.2 Vuelve al paso 2.	

#### 4.4 RF4 Baja Usuario

<b>CU4</b>	
<b>Descripción:</b> Se da de baja un usuario.	
<b>Actores:</b> Usuario Administrador - Sistema	
<b>Pre-Condición:</b> El usuario a eliminar pertenece al sistema.	
<b>Post-Condición:</b> Se da de baja el usuario del sistema.	
<b>Curso Básico</b>	
<b>Usuario Administrador</b>	<b>Sistema</b>
1. El administrador selecciona la opción de baja de usuario.	
2. Se selecciona el usuario a dar de baja.	
3. El administrador aprieta el botón de baja	
	4. El sistema elimina el usuario
	5. fin
<b>Curso Alternativo</b>	
3.1 Se aprieta el botón cancelar.	
3.1.2 Se cierra la ventana.	

#### 4.5 RF5 Dar alta de Equipo

<b>CU5</b>	
<b>Descripción:</b> Se da de alta un equipo en el sistema.	
<b>Actores:</b> Usuario Administrador - Sistema	
<b>Pre-Condición:</b> -	
<b>Post-Condición:</b> Se da de alta el equipo.	
<b>Curso Básico</b>	
<b>Usuario Administrador</b>	<b>Sistema</b>
1. El administrador selecciona la opción de alta de equipo.	
2. Se ingresan los datos correspondientes (nombre, descripción, cantidad máxima de usuarios que ingresan).	
3. El administrador aprieta el botón de baja	
	4. Se validan los datos
	5. El sistema elimina el usuario
	6. fin
<b>Curso Alternativo</b>	
3.1 Se aprieta el botón cancelar. 3.1.2 Se cierra la ventana. 4.1. Se Ingresa un nombre invalido. 4.1.1 Se notifica el error. 4.1.2 Vuelve al paso 2. 4.2 Se ingresa una descripción invalida. 4.2.1 Se notifica el error 4.2.2 Vuelve al paso 2 4.3 Se ingresa una cantidad invalida. 4.3.1 Se notifica el error 4.3.2 Vuelve al paso 2 4.4 Se ingresa un usuario inválido. 4.4.1 Se notifica el error 4.4.2 Vuelve al paso 2	

#### 4.6 RF6 Modificación de Equipo

<b>CU6</b>	
<b>Descripción:</b> Se modifican los datos del equipo.	
<b>Actores:</b> Usuario Administrador - Sistema	
<b>Pre-Condición:</b> Existe el equipo	
<b>Post-Condición:</b> Se modifican los datos del equipo	
<b>Curso Básico</b>	
<b>Usuario Administrador</b>	<b>Sistema</b>
1. El administrador selecciona la opción de modificar equipo.	
2. El administrador ingresa los datos a modificar (nombre, descripción, cantidad máxima de usuarios, lista de usuarios).	
3. El administrador aprieta el botón modificar.	
	4. Se verifican los datos
	5. El sistema modifica el usuario
	6. fin
<b>Curso Alternativo</b>	
3.1 Se aprieta el botón cancelar. 3.1.2 Se cierra la ventana. 4.1. Se Ingresa un nombre invalido. 4.1.1 Se notifica el error. 4.1.2 Vuelve al paso 2. 4.2 Se ingresa una descripción invalida. 4.2.1 Se notifica el error 4.2.2 Vuelve al paso 2 4.3 Se ingresa una cantidad invalida. 4.3.1 Se notifica el error 4.3.2 Vuelve al paso 2 4.4 Se ingresa un usuario inválido. 4.4.1 Se notifica el error 4.4.2 Vuelve al paso 2	

#### 4.7 RF7 Baja de equipo

<b>CU7</b>	
<b>Descripción:</b> Se da de baja el equipo seleccionado del sistema.	
<b>Actores:</b> Usuario Administrador - Sistema	
<b>Pre-Condición:</b> Existe el equipo	
<b>Post-Condición:</b> Se da de baja el equipo.	
<b>Curso Básico</b>	
<b>Usuario Administrador</b>	<b>Sistema</b>
1. El administrador selecciona la opción de baja de equipo.	
2. El administrador selecciona el equipo.	
3. El administrador aprieta el botón eliminar.	
	4. El sistema elimina el equipo.
	5. fin
<b>Curso Alternativo</b>	
3.1 Se aprieta el botón cancelar.	
3.1.2 Se cierra la ventana.	

#### 4.8 RF8 Informes sobre pizarrones creados por equipos

<b>CU8</b>	
<b>Descripción:</b> Se devuelve una vista rápida de la cantidad de pizarrones creados por equipo, filtrando por fecha de creación y equipos. Se muestran: Equipo creador, fecha de creación, fecha de última modificación, cantidad de elementos en el pizarrón.	
<b>Actores:</b> Usuario Administrador - Sistema	
<b>Pre-Condición:</b> -	
<b>Post-Condición:</b> Se entrega el informe.	
<b>Curso Básico</b>	
<b>Usuario Administrador</b>	<b>Sistema</b>
1. El administrador selecciona la opción de información sobre pizarrones.	
2. El administrador aprieta el botón Ver.	
	3. El sistema muestra los datos.
	4. fin
<b>Curso Alternativo</b>	
2.1 Se aprieta el botón cancelar.	
2.1.2 Se cierra la ventana.	



#### 4.9 RF9 Informe sobre comentarios resueltos por usuario

<b>CU9</b>	
<b>Descripción:</b> Se devuelve una vista rápida de los comentarios resueltos por usuario, filtrando por fecha de creación de comentario, resolución del mismo, de usuario creado y del usuario que lo resuelve. Se muestran: fecha de creación de comentario, usuario creador, usuario que resuelve, pizarrón al que pertenece el elemento y fecha de resuelto el comentario.	
<b>Actores:</b> Usuario Administrador - Sistema	
<b>Pre-Condición:</b> -	
<b>Post-Condición:</b> Se muestra el informe pedido.	
<b>Curso Básico</b>	
<b>Usuario Administrador</b>	<b>Sistema</b>
1. El administrador selecciona la opción de Informe sobre comentarios resueltos.	
2. El administrador aprieta el botón Ver.	
	3. El sistema muestra el informe.
	4. fin
<b>Curso Alternativo</b>	
2.1 Se aprieta el botón cancelar.	
2.1.2 Se cierra la ventana.	

#### 4.10 RF10 Alta Pizarrón

<b>CU10</b>	
<b>Descripción:</b> Se da de alta un pizarrón, se pide: su nombre, equipo al que pertenece, una breve descripción, un ancho y un alto.	
<b>Actores:</b> Usuario - Sistema	
<b>Pre-Condición:</b> -	
<b>Post-Condición:</b> Se da de alta el pizarrón.	
<b>Curso Básico</b>	
<b>Usuario</b>	<b>Sistema</b>
1. El usuario selecciona la opción de alta pizarrón.	
2. El administrador ingresa los datos correspondientes (nombre, descripción, ancho y alto).	
3. El administrador aprieta el botón registrar.	
	4. Se verifican los datos
	5. El sistema ingresa el pizarrón.
	6. fin
<b>Curso Alternativo</b>	
3.1 Se aprieta el botón cancelar. 3.1.2 Se cierra la ventana. 4.1. Se Ingresa un nombre invalido. 4.1.1 Se notifica el error. 4.1.2 Vuelve al paso 2. 4.2 Se ingresa una descripción invalida. 4.2.1 Se notifica el error 4.2.2 Vuelve al paso 2 4.3 Se ingresa una altura invalida. 4.3.1 Se notifica el error 4.3.2 Vuelve al paso 2 4.4 Se ingresa un ancho inválido. 4.4.1 Se notifica el error 4.4.2 Vuelve al paso 2	

#### 4.11 RF11 Modificar Pizarrón

<b>CU11</b>	
<b>Descripción:</b> Se permite modificar la información del pizarrón, ya sea su nombre o descripción.	
<b>Actores:</b> Usuario - Sistema	
<b>Pre-Condición:</b> -	
<b>Post-Condición:</b> Se modifican los datos del pizarrón	
<b>Curso Básico</b>	
<b>Usuario</b>	<b>Sistema</b>
1. El usuario selecciona la opción de modificar pizarrón.	
2. El administrador ingresa los datos a modificar (nombre, descripción, ancho, alto).	
3. El administrador aprieta el botón modificar.	
	4. Se verifican los datos
	5. El sistema modifica el pizarron
	6. fin
<b>Curso Alternativo</b>	
3.1 Se aprieta el botón cancelar. 3.1.2 Se cierra la ventana. 4.1. Se Ingresa un nombre invalido. 4.1.1 Se notifica el error. 4.1.2 Vuelve al paso 2. 4.2 Se ingresa una descripción invalida. 4.2.1 Se notifica el error 4.2.2 Vuelve al paso 2 4.3 Se ingresa una altura invalida. 4.3.1 Se notifica el error 4.3.2 Vuelve al paso 2 4.4 Se ingresa un ancho inválido. 4.4.1 Se notifica el error 4.4.2 Vuelve al paso 2	

#### 4.12 RF12 Baja Pizarrón

<b>CU12</b>	
<b>Descripción:</b> Un usuario administrador o el creador del pizarrón podrá dar de baja el pizarrón.	
<b>Actores:</b> Usuario Administrador/creador del pizarrón - Sistema	
<b>Pre-Condición:</b> Existe un pizarrón.	
<b>Post-Condición:</b> Se da de baja el pizarrón	
<b>Curso Básico</b>	
<b>Usuario Administrador/Creador</b>	<b>Sistema</b>
1. El administrador/creador selecciona la opción de baja de pizarrón.	
2. El administrador aprieta el botón modificar.	
	3. El sistema da de baja el pizarrón
	4. fin
<b>Curso Alternativo</b>	
3.1 Se aprieta el botón cancelar.	
3.1.2 Se cierra la ventana.	

#### 4.13 RF13 Agregar elemento a pizarrón

<b>CU13</b>	
<b>Descripción:</b> El usuario perteneciente al pizarrón podrá ingresar imágenes o campos de texto al mismo. Se puede hacer mediante un drag and drop.	
<b>Actores:</b> Usuario - Sistema	
<b>Pre-Condición:</b> Existe pizarrón	
<b>Post-Condición:</b> Se agrega elemento al pizarrón	
<b>Curso Básico</b>	
<b>Usuario</b>	<b>Sistema</b>
1. El usuario selecciona el tipo de elemento a agregar.	
2. El usuario ingresa el elemento.	
	3. El sistema muestra el elemento en el pizarrón
	4. fin
<b>Curso Alternativo</b>	
1.1 No se selecciona elemento.	
1.1.2 No pasa nada.	

#### 4.14 RF14 Modificar elemento de un pizarrón

<b>CU14</b>	
<b>Descripción:</b> El usuario perteneciente al pizarrón podrá modificar la información de los elementos pertenecientes al mismo.	
<b>Actores:</b> Usuario - Sistema	
<b>Pre-Condición:</b> Existe elemento a modificar	
<b>Post-Condición:</b> Se modifican el elemento	
<b>Curso Básico</b>	
<b>Usuario</b>	<b>Sistema</b>
1. El usuario selecciona la opción de modificar elemento.	
2. El usuario modifica.	
3. El usuario aprieta el botón modificar.	
	4. Se verifican los datos
	5. El sistema modifica el elemento
	6. fin
<b>Curso Alternativo</b>	
3.1 Se aprieta el botón cancelar.	
3.1.2 Se cierra la ventana.	
4.1. Se Ingresa una modificación invalida.	
4.1.1 Se notifica el error.	
4.1.2 Vuelve al paso 2.	

#### 4.15 RF15 Agregar Comentario

<b>CU15</b>	
<b>Descripción:</b> Los usuarios pertenecientes a un pizarrón podrán agregar un comentario a un elemento, para hacer esto deberán ingresar la descripción del comentario.	
<b>Actores:</b> Usuario - Sistema	
<b>Pre-Condición:</b> Existe elemento que se va a comentar	
<b>Post-Condición:</b> Se agrega el comentario al elemento	
<b>Curso Básico</b>	
<b>Usuario</b>	<b>Sistema</b>
1. El usuario selecciona el elemento a comentar.	
2. El usuario ingresa el comentario.	
3. El usuario aprieta el botón de ingresar comentario.	
	4. Se verifican los datos
	5. El sistema ingresa el comentario
	6. fin
<b>Curso Alternativo</b>	
3.1 Se aprieta el botón cancelar.	
3.1.2 Se cierra la ventana.	
4.1. Se Ingresa un comentario invalido.	
4.1.1 Se notifica el error.	
4.1.2 Vuelve al paso 2.	

#### 4.16 RF16 Baja Elemento

<b>CU16</b>	
<b>Descripción:</b> Los usuarios pertenecientes a un pizarrón podrán eliminar elementos pertenecientes al mismo.	
<b>Actores:</b> Usuario - Sistema	
<b>Pre-Condición:</b> Existe el elemento	
<b>Post-Condición:</b> Se modifican los datos del usuario	
<b>Curso Básico</b>	
<b>Usuario</b>	<b>Sistema</b>
1. El usuario selecciona el elemento.	
2. El usuario aprieta el botón eliminar.	
	3. El sistema borra el elemento
	4. fin
<b>Curso Alternativo</b>	
2.1 Se aprieta el botón cancelar.	
2.1.2 Se cierra la ventana.	

#### 4.17 RF17 Imprimir pizarrón

<b>CU1</b>	
<b>Descripción:</b> Se podrá imprimir el pizarrón a un formato PDF o .png.	
<b>Actores:</b> Usuario - Sistema	
<b>Pre-Condición:</b> Existe pizarrón	
<b>Post-Condición:</b> Se modifican los datos del usuario	
<b>Curso Básico</b>	
<b>Usuario</b>	<b>Sistema</b>
1. El usuario selecciona el tipo.(pdf o png)	
2. El usuario selecciona la opción de imprimir pizarrón.	
	3. El sistema modifica el usuario
	4. fin
<b>Curso Alternativo</b>	
3.1 Se aprieta el botón cancelar.	
3.1.2 Se cierra la ventana.	

#### 4.18 RF18 Conexión de elementos.

<b>CU18</b>	
<b>Descripción:</b> Se conecta dos elementos formando uno nuevo.	
<b>Actores:</b> Usuario - Sistema	
<b>Pre-Condición:</b> Existe pizarrón y almenos dos elementos.	
<b>Post-Condición:</b> Quedan conectados dos elementos.	
<b>Curso Básico</b>	
<b>Usuario</b>	<b>Sistema</b>
1. El usuario selecciona un elemento.	
2. El usuario presiona el boton de conectar	
3. El usuario selecciona otro elemento.	
4. Se presiona el boton enlazar.	
	5. El sistema crea una conexión entre los dos elementos.
	6. fin
<b>Curso Alternativo</b>	
3.1 Se selecciona el mismo elemento.	
3.1.1 El sistema avisa sobre el error, vuelve al caso 1.	
4.1 Se aprieta el botón cancelar.	
4.1.1 Se cierra la ventana.	

#### 4.19 RF19 Ranking de usuarios.

<b>CU19</b>	
<b>Descripción:</b> Se muestran los puntajes de los usuarios del equipo seleccionado.	
<b>Actores:</b> Usuario - Sistema	
<b>Pre-Condición:</b> Existen usuarios y equipos, y los mismos pertenecen al equipo.	
<b>Post-Condición:</b> Muestra los ranking del equipo.	
<b>Curso Básico</b>	
<b>Usuario</b>	<b>Sistema</b>
1. El usuario selecciona un equipo.	
2. El usuario presiona el boton de mostrar.	
	3. El sistema despliega el ranking de los usuarios
	4. fin
<b>Curso Alternativo</b>	
2.1 Se presiona el botón cancelar.	
2.1.1 Se cierra la ventana y se vuelve al menu principal.	

#### 4.20 RF20 Reinicio de puntaje

<b>CU20</b>	
<b>Descripción:</b> Se selecciona un equipo y se borran los puntajes.	
<b>Actores:</b> Administrador - Sistema	
<b>Pre-Condición:</b> Existen usuarios y equipos, y los mismos pertenecen al equipo.	
<b>Post-Condición:</b> Borra los puntajes del usuario en el equipo.	
<b>Curso Básico</b>	
<b>Usuario</b>	
1. El usuario administrador selecciona un equipo.	
2. El usuario presiona el botón Borrar puntuación.	
	3. El sistema reinicia el puntaje
	4. fin
<b>Curso Alternativo</b>	
2.1 Se presiona el botón cancelar.	
2.1.1 Se cierra la ventana y se vuelve al menú principal.	
2.2 No se selecciona un equipo anteriormente.	
2.2.1 Se muestra un error solicitando que seleccione un equipo.	



#### 4.21 RF21 Selección de puntajes.

<b>CU21</b>	
<b>Descripción:</b> Se selecciona cuanto valor van a tener ciertas acciones de puntajes.	
<b>Actores:</b> Administrador - Sistema	
<b>Pre-Condición:</b> -	
<b>Post-Condición:</b> Deja registrado el valor de los puntajes de las acciones.	
<b>Curso Básico</b>	
<b>Usuario</b>	<b>Sistema</b>
1. El usuario administrador selecciona un valor para cada acción.	
2. El usuario presiona el botón Asignar puntuaciones.	
	3. El sistema guarda los nuevos valores.
	4. fin
<b>Curso Alternativo</b>	
2.1 Se presiona el botón cancelar.	
2.1.1 Se cierra la ventana y se vuelve al menú principal.	

## 7. Justificación de Diseño

### 7.1 Justificación de Diseño de programa

En el diseño realizado intentamos que fuera más óptimo para nuestro sistema, tratando que el mismo sea mantenible y consistente frente a cambios. Siempre diseñamos nuestro código tratando de lograr que el mismo no tuviera demasiada fragilidad y sea poco rígido.

Una consideración de diseño fue implementar una generalización, en términos de código una herencia. Teniendo como súper clase a "User" de las cuales heredan "Administrator" y "Colaborator". Para poder reusar código, pero principalmente por el poder que el polimorfismo nos brinda y también previendo en el futuro de que haya algún otro tipo distinto de usuario como podría ser un observador y que además prevenimos también que en caso de que los usuarios tengan comportamientos más específicos se pueden separar perfectamente.

Luego decidimos implementar otra herencia de "Element", de quien heredan "Image" y "TextBox". En un primer momento habíamos decidido implementarlos por separado, pero luego vimos que ambas tenían variables en común, por ende, decidimos generar la herencia, haciendo que en el futuro si se decide ingresar un elemento nuevo, se podrá con facilidad.

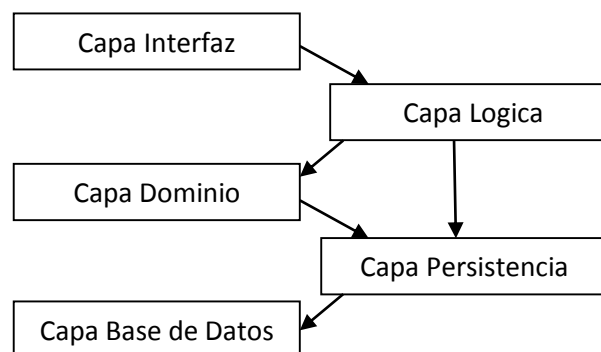
Gracias a como teníamos construido el obligatorio 1, el pasaje a base de datos fue sencillo, simplemente hacer pequeños cambios en las clases del paquete persistence, para que, en vez de guardar en memoria, se guarde en la base de datos.

Pero se debió realizar unos pequeños cambios, se cambiaron de lugar las interfaces de las funciones que solían estar en dominio se cambiaron para lógica, para bajar el acoplamiento innecesario entre paquetes.

Dentro de los paquetes de lógica y persistencia, la misma se divide en dos partes, implementación e interfaz. Esta decisión se tomó para implementar el principio de inversión de dependencia, haciendo que, si se quiere cambiar la implementación en un futuro, esta no cause problema siempre y cuando respete el contrato de la interfaz.

Dividimos nuestro obligatorio en 3 capas, con 2 auxiliares como mediadores entre ellas:

- **Capa de interfaz:** Es lo que ve el usuario. Es encargada de mostrarle, solicitarle y capturar la información. Esta capa se comunica con la capa de la lógica, y no realiza ninguna validación salvo comprobaciones de formato de la información que sean triviales.
- **Capa de lógica:** Es donde están las restricciones impuestas por el dominio. En esta capa se procesa toda la información que ingresa el usuario y la maneja las mismas, luego se conecta con el paquete de persistencia para guardar la información en la base de datos si es necesario.
- **Capa de dominio:** En esta capa se encuentran las clases las cuales se van a basar los objetos en el obligatorio, tiene como nexo a las otras capas a la capa de lógica y la capa de persistencia.
- **Capa de persistencia:** En esta capa es donde se hace el nexo con la base de datos, separamos este paquete porque consideramos que este tipo de operaciones no debe mezclarse con otras funciones, ya que reduciría la cohesión del código.
- **Capa de base de datos:** En esta capa se encuentra la base de datos, de donde guardamos toda la información a través de la persistencia.



Adicionalmente se utiliza un paquete de Utility, el cual brinda ciertas validaciones generales que pueden ser utilizadas por cualquier paquete.

## 7.2 Utilización de patrones

### 7.2.1 Patron Facade

Se utiliza una clase de fachada llamada Instance, la cual actúa como medio de comunicación entre la lógica y la interfaz.

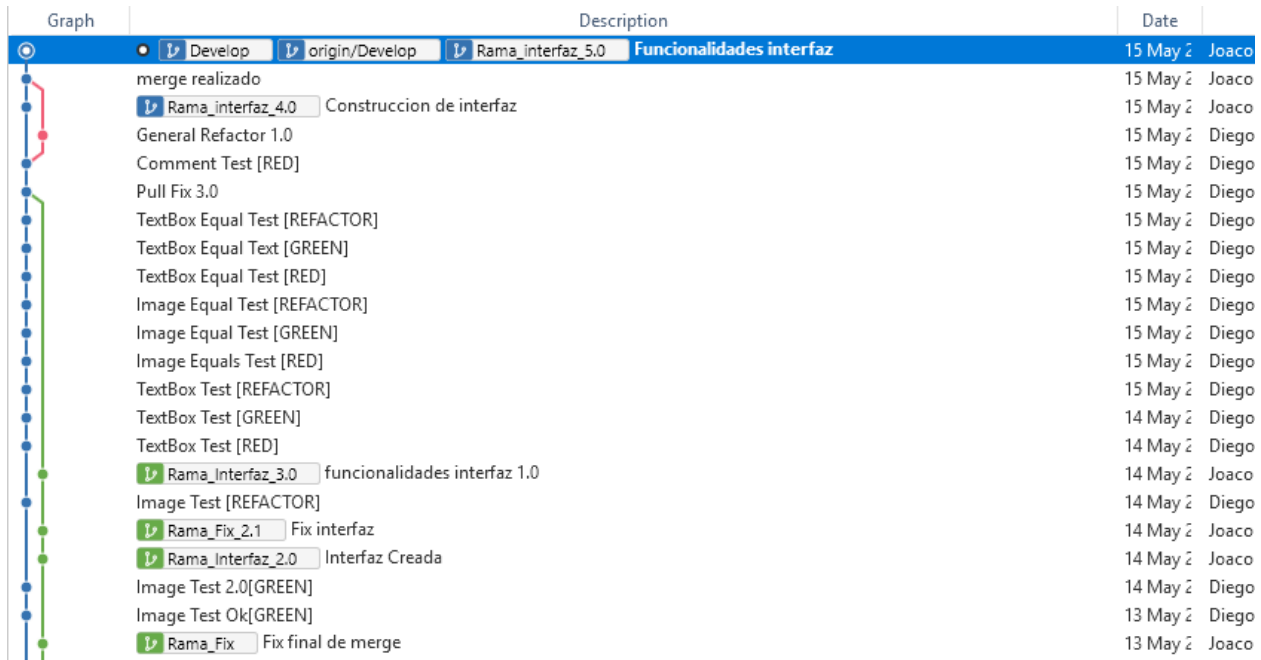
La clase contiene interfaces de lógica la cual se pueden llamar desde la interfaz, sin importar de que parte de la lógica viene, ya que lo que le importa es solo que la clase fachada tenga la información.

Esto baja el acoplamiento entre la interfaz y el dominio, y también permite que si se cambia el dominio, no afecta a la interfaz, ya que si se siguen cumpliendo los contratos de las interfaces la fachada permite que su funcionamiento siga correcto.

### 7.2.2 Patron Strategy

Se utilizó el patrón strategy para las clases de interfaz de funciones, esto nos facilitó para funciones compartidas por clases diferentes, como ITextboxHandler e IImageHandler, ambas heredan de la interfaz IElementHandler pero las clases que heredan son utilizadas por sus respectivos manejadores de maneras distintas.

## 8. Evidencia de TDD.



Se estuvo trabajando de forma estructurada y siguiendo los lineamientos de TDD, para lograr un trabajo de calidad y profesional. Se comenzó con la técnica TDD implementando las pruebas de los Requerimientos Funcionales mencionados al comienzo de la documentación. Primero se creaban las pruebas, luego se codificaba lo mínimo indispensable para que pudieran pasar, luego se pasaba en limpio estableciendo clean code. A medida que se avanzaba, se iba moldeando el código de forma correcta.

En el segundo obligatorio, tuvimos inconvenientes, el cual fuimos arreglando sobre la marcha, perjudicando el proceso TDD, el cual se vio reflejado en el Code Coverage.

Dirección del repositorio:

<https://github.com/joacooribe/180521-198685.git>

## 9. Evidencia de Clean Code

El sistema fue diseñado tomando en cuenta los principios de clean code, brindando un código fácil de leer, entender y modificar.

Se implementaron nombres correctos, que indican correctamente cuál es su función. Gracias a esto, el trabajo en equipo fue sencillo, ya que ambos sabíamos que hizo cada uno y siempre respetando los estándares del lenguaje de C#.

No hay comentarios en el código, dejando en claro que lo codificado está suficientemente claro la implementación del mismo.

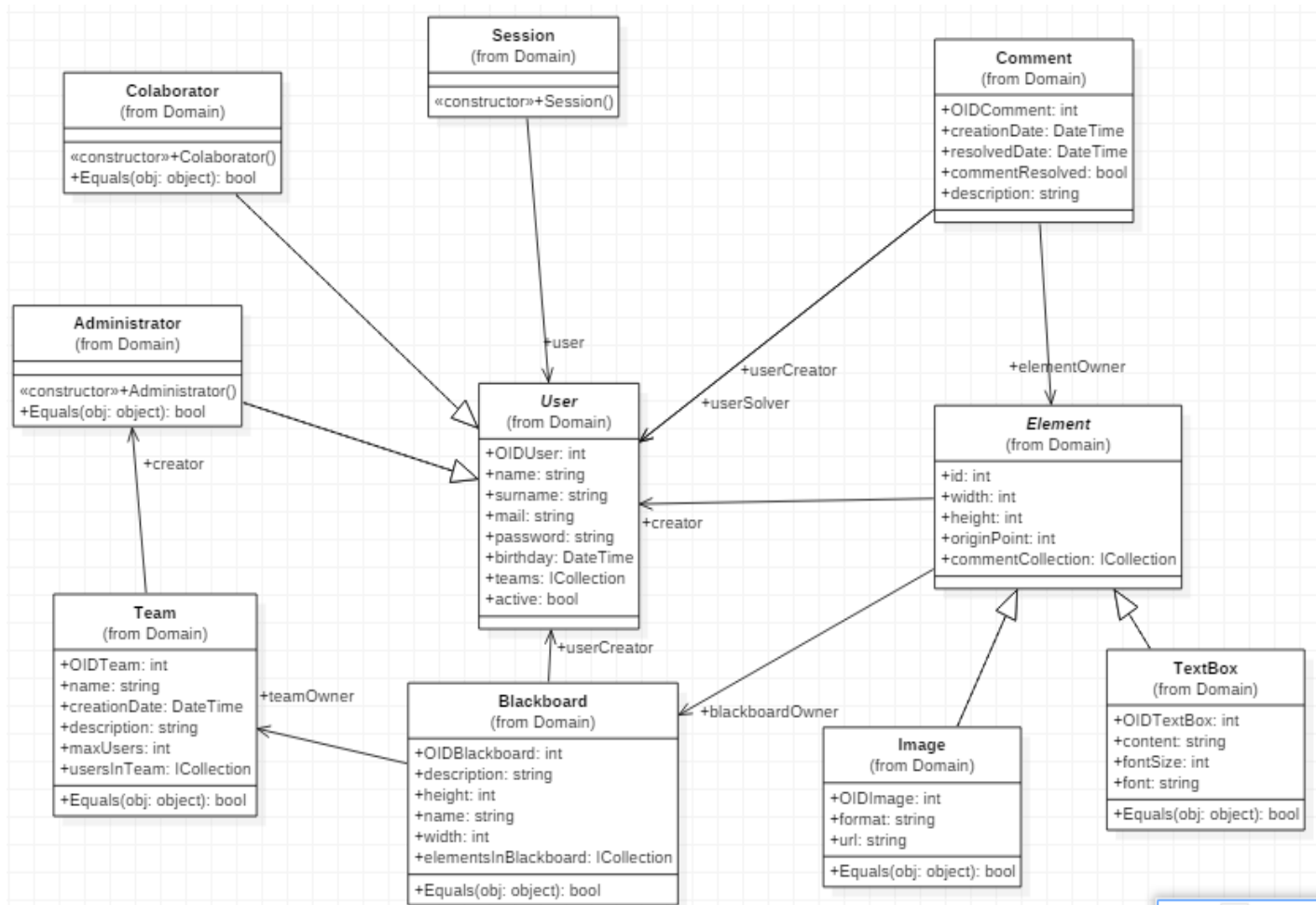
Todo el desarrollo del Dominio fue realizado mediante la técnica TDD, en un futuro que el código fuera modificado sin temor a introducir nuevos errores, para lograr esto se tuvieron en cuenta las 5 características de las pruebas automatizadas (Fast, Independent, Repetable, Self-Validating, Timely).

Respecto a la separación de las responsabilidades de cada parte del código, se buscó que La solución está correctamente separada por proyectos, en los cuales se encapsulan las excepciones, la lógica, la persistencia, la interfaz, las utilidades, las pruebas y el dominio.

Se respetó el principio de abierto y cerrado, haciendo que los métodos que comparten clases hereditarias, sean fáciles de expandir sin necesidad de modificar el código.

Se codifico en inglés, para que el código pueda ser entendido internacionalmente.

## 10. Diagrama de clases

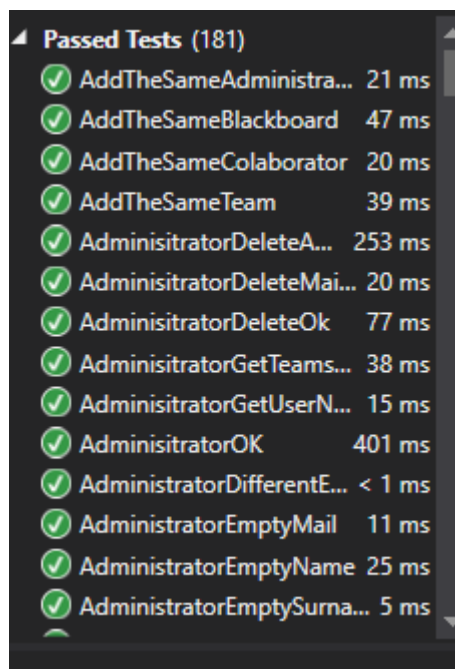


## 11. Cobertura del código

Hierarchy	Not Covered (Blocks)	Not Covered (% Blocks)	Covered (Blocks)	Covered (% Blocks)
joaco_DESKTOP-VST24DA 2017-...	340	10.98 %	2756	89.02 %
utility.dll	2	1.74 %	113	98.26 %
test.dll	137	9.75 %	1268	90.25 %
persistence.dll	141	17.89 %	647	82.11 %
logic.dll	47	8.23 %	524	91.77 %
exceptions.dll	0	0.00 %	13	100.00 %
domain.dll	13	6.37 %	191	93.63 %

No tenemos el 100% porque nos desviamos del proceso de TDD durante esta segunda parte, creando funciones antes de las pruebas.

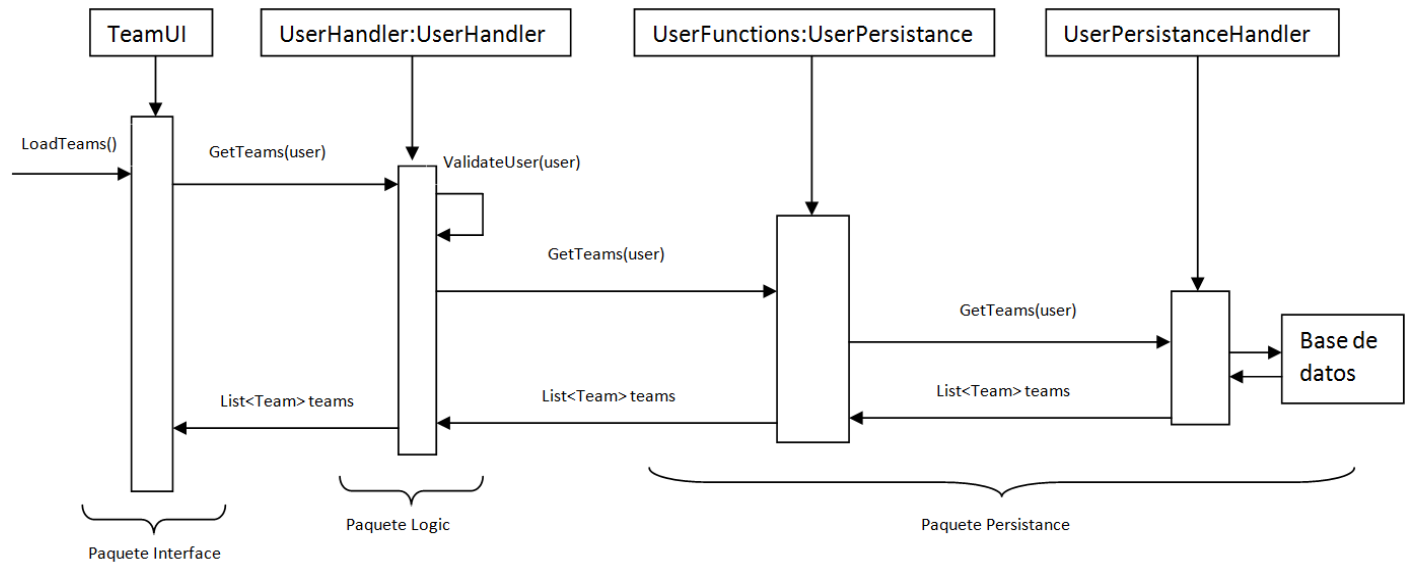
## 12. Pruebas



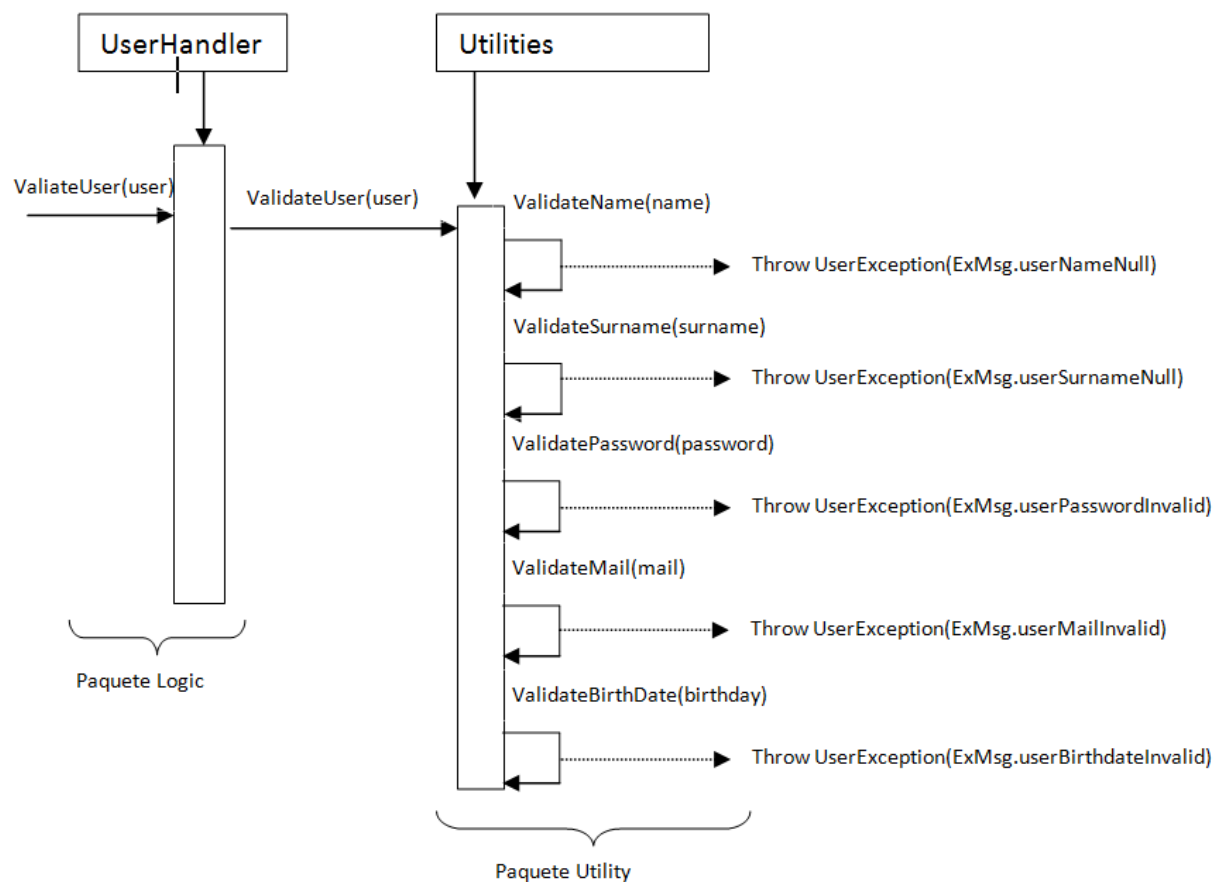
Durante nuestro proceso creamos varias pruebas que validan que nuestro programa anda correctamente.

## 13. Diagrama de interacción.

### 13.1 Paquete persistence, Logic e Interface



### 13.2 Paquete Logic y Utility





## 14. Datos de Prueba

### Usuarios

#### Administrador

OIDUser	Name	Surname	Mail	Password	BirthDay	Active	Teams
0	Admin	Istrator	admin@admin.com	a11111	09/10/1992	True	Team1,Team2

#### Colaborator

OIDUser	Name	Surname	Mail	Password	Birthday	Active	Teams
1	Colab	Orator	colab@gmail.com	Colaborator1	09/10/1992	True	Team2

### Team

OIDTeam	Name	Creator	CreationDate	Description	MaxUsers	UsersInTeam
0	Team1	Admin	now	This is a description	5	Admin
1	Team2	Admin	now	Description	3	Admin, Colab