

UNIVERSIDAD ORT URUGUAY

Diseño de Aplicaciones 1 Segundo Obligatorio

Participantes:

Federico Cetraro (193221)

Fabián Grobert(194738)

Docentes:

Gonzalo Laguna

Gabriel Piffarretti

Índice

Índice	2
Introducción	3
Objetivo	3
Justificación del diseño	3
Requerimientos Funcionales	4
Requerimientos No Funcionales	8
Casos de Uso	9
Diagrama de paquetes	39
Diagrama de secuencia	40
Evidencia TDD:	41
Clean Code	43
Practicas aplicadas	43
Evidencia de Clean Code	44
UML	46
Manual de instalación	47

Introducción

Objetivo

El objetivo de este trabajo es el desarrollo de una aplicación de escritorio. El sistema deberá ser capaz de realizar todas las funcionalidades.

Justificación del diseño

El grupo tomó la decisión de diseñar la arquitectura de la solución en 4 capas separadas en distintos paquetes. Cada una de estas capas cumple con una función determinada dentro del sistema: interfaz, handlers (lógica del sistema), persistencia, y entidades del dominio. Estas capas se comunican entre sí con ciertas restricciones para asegurar un menor acoplamiento de las clases de la solución.

En el paquete BlackBoards podemos encontrar todas las clases que tienen un significado en la realidad del problema a solucionar. Hay doce clases principales que son User, Administrator, Collaborator, Team, Comment, EstablishedScoreTeam, ScoreInUserTeam, Item, Picture, TextBox, Connection y BlackBoard que representan los objetos que el sistema debe manipular. Luego, están las clases auxiliares que se utilizarán como atributos de las clases principales. En estas clases auxiliares podemos encontrar Score, ValidationReturn, Dimension, Coordinate, DirectionType.

En el paquete Handlers se encuentran las clases que se ocupan de la lógica del software, hay una por cada clase que representa algo de la realidad en el dominio, lo que da como resultado doce clases. Cada una de estas clases se ocupa de realizar las validaciones y de tener todos los métodos necesarios para cumplir con todas las funcionalidades requeridas.

Por último, está el paquete UIBlackBoards, en este se encuentran todas las clases que se utilizan para manejar la interfaz de la solución.

Requerimientos Funcionales

RF1 – Alta de Usuario

Descripción: El sistema permite registrar un usuario nuevo a la aplicación. Pide ingresar: nombre, apellido, e-mail, fecha de nacimiento y contraseña. También se deberá indicar si se trata de un administrador o usuario corriente.

Prioridad: Alta

RF2 – Modificar Usuario

Descripción: El sistema permite modificar los datos de un usuario ya existente, es decir cambiar los atributos de uno ya registrado.

Prioridad: Alta

RF3 – Baja de Usuario

Descripción: El sistema permite borrar un usuario existente.

Prioridad: Alta

RF4– Registrar Equipo

Descripción: El sistema tiene que posibilitar al usuario registrar un equipo. Pide ingresar: su nombre, la fecha de creación, una breve descripción de sus tareas y la cantidad máxima de usuarios que pueden pertenecer al mismo. Se solicitará como requisito mínimo que en todo momento, un equipo debe tener mínimo un usuario.

Prioridad: Alta

RF5 – Modificar Equipo

Descripción: El sistema tiene que posibilitar al usuario modificar un equipo registrado previamente.

Prioridad: Alta

RF6 – Eliminar Equipo

Descripción: El sistema tiene que posibilitar al usuario eliminar un equipo registrado previamente.

Prioridad: Alta

RF7 – Informe Pizarrones creados por equipo

Descripción: El sistema permite visualizar la cantidad de pizarrones creados por equipo, con la posibilidad de filtrarlos por fecha de creación y equipos. En este se muestran: equipo creador, fecha de creación, fecha de última modificación, cantidad de elementos en el pizarrón.

Prioridad: Baja

RF8 – Comentarios Resueltos por Usuario

Descripción: El sistema tiene que posibilitar al usuario ver los datos de comentarios resueltos por el usuario, con la posibilidad de filtrarlos por fecha de creación de comentario, de resolución del mismo, de usuario creado y del usuario que lo resuelve. Los datos a mostrar son: fecha de creación de comentario, usuario creador, usuario que resuelve, pizarrón al que pertenece el elemento y fecha de resuelto el comentario.

Prioridad: Baja

RF9 – Alta de Pizarrón

Descripción: El sistema permite ingresar un nuevo pizarrón. Pide ingresar: nombre, equipo al que pertenece, breve descripción, su ancho y alto.

Prioridad: Alta

RF10 – Modificar Pizarrón

Descripción: El sistema permite que todos los usuarios pertenecientes al equipo del pizarrón podrán modificar sus datos.

Prioridad: Baja

RF11 – Baja de Pizarrón

Descripción: El sistema permite a los usuarios administradores o el usuario creador borrar el pizarrón

Prioridad: Alta

RF12 – Visualizar un pizarrón

Descripción: El sistema tiene que posibilitar al usuario ver la lista de pizarrones de un equipo, se podrá seleccionar uno y abrirlo.

Prioridad: Alta

RF13 – Agregar un elemento al Pizarrón

Descripción: El sistema permite agregar un elemento al pizarrón. El ancho y alto del elemento está predefinido por defecto, la posición se indica arrastrando el elemento hacia ella. Los elementos pueden ser imágenes (jpg, png, gif o jpeg) o cuadros de texto (se deberá indicar su contenido, tamaño de letra y tipo de letra).

Prioridad: Alta

RF14 – Modificar un elemento de un pizarrón

Descripción: El sistema tiene que posibilitar al usuario modificar los elementos. Esta modificación puede ser tanto de su ubicación en el pizarrón, tamaño o contenido.

Para modificar la posición de un elemento en la pizarra se debe seleccionar y arrastrar un elemento en la misma. Para modificar el tamaño de un elemento, deberá seleccionar uno de sus bordes o esquinas y demarcar el tamaño final.

Prioridad: Baja

RF15 – Agregar Comentario a un Elemento

Descripción: El sistema permite a los usuarios pertenecientes al equipo comentar sobre los elementos del pizarrón.

Prioridad: Alta

RF16 -Marcar Comentario como realizado

Descripción: El sistema permite a los usuarios pertenecientes al equipo del pizarrón actual marcar como realizado los comentarios que se ya se tomaron en cuenta.

Prioridad: Baja

RF17 – Imprimir Pizarrón a PDF o PNG

Descripción: El sistema permite generar un archivo pdf o png que muestre al pizarrón.

Prioridad: Baja

RF18 –Eliminar un elemento de un pizarrón

Descripción: El sistema tiene que posibilitar al usuario eliminar elementos de los pizarrones.

Prioridad: Alta

RF19 -Iniciar sesión

Descripción: El sistema permite iniciar sesión a un usuario ya existente ingresando su nombre de usuario y su contraseña correctamente.

Prioridad: Alta

RF20 -Cerrar sesión

Descripción: El sistema permite al usuario cerrar su sesión actual.

Prioridad: Alta

RF21 -Generar una contraseña

Descripción: El sistema permite a los usuarios Administradores generar una nueva contraseña por defecto.

Prioridad: Baja

RF22 – Reiniciar puntajes

Descripción: El sistema permite a los administradores, para cada equipo, reiniciar los puntajes de los usuarios para el equipo seleccionado.

Prioridad: Baja

RF23 – Ver puntajes

Descripción: El sistema permite a los usuarios ver los puntajes de los equipos a los cuales pertenezca.

Prioridad: Alta

RF24 – Crear conexiones

Descripción: El sistema permite a los usuarios crear conexiones entre 2 elementos.

Pide ingresar: los dos elementos, un nombre, una descripción (opcional) y podrá tener direccionalidad, bidireccionalidad o ninguna de las dos.

Prioridad: Alta

Requerimientos No Funcionales

RNF1 – Sistema Operativo

Descripción: El sistema tiene que funcionar en Windows 7 o superior.

RNF2 – Uso simple

Descripción: nueve de cada diez personas deben de aprender a utilizar el sistema en menos de dos minutos.

RNF3- Lenguaje de desarrollo

Descripción: El sistema se debe desarrollar en c#.

RNF4- Idioma

Descripción: El idioma del software debe ser español.

RNF5- Generar datos de prueba

Descripción: El sistema tiene que permitir generar datos de prueba de manera automática.

RNF6- Clean code

Descripción: El sistema se debe desarrollar siguiendo las prácticas de clean code.

RNF7- Portable

Descripción: El software debe ser portátil.

RNF8- Persistencia de los datos

Descripción: El software debe permitir que los datos del sistema sean persistidos en una base de datos.

Casos de Uso

Caso de uso 1: Alta de Usuario

ID: CU1	
Nombre: Alta de Usuario	
Descripción: Se quiere ingresar un usuario al sistema.	
Actores: Administrador	
Pre-condición:	
Curso normal:	
Administrador	Sistema
1- El administrador ingresa nombre, apellido, e-mail, fecha de nacimiento y contraseña deseados. También se deberá indicar si se trata de un administrador o usuario corriente. y selecciona el botón aceptar	
	2- El sistema registra un usuario
Curso Alternativo:	
1.1- El administrador deja vacía la casilla del nombre, apellido, e-mail, fecha de nacimiento y/o contraseña . El sistema muestra un mensaje de error. 1.2- El administrador ingresa un e-mail ya registrado en el sistema . El sistema muestra un mensaje de error. 1.3- El administrador ingresa una contraseña menor a 4 cifras . El sistema muestra un mensaje de error. 1.4- El administrador ingresa 2 contraseñas diferentes . El sistema muestra un mensaje de error.	

Post-condición: El Usuario es ingresado correctamente al sistema.

Caso de uso 2: Modificar Usuario

ID: CU2	
Nombre: Modificar Usuario	
Descripción: Se quiere modificar un usuario del sistema.	
Actores: Administrador	
Pre-condición:	
Curso normal:	
Administrador	Sistema
1- El administrador ingresa nombre, apellido, e-mail, fecha de nacimiento y/o contraseña nuevos. También se deberá indicar si se trata de un administrador o usuario corriente. y selecciona el botón aceptar	
	2- El sistema modifica el usuario seleccionado existente en el sistema.
Curso Alternativo:	
1.1- El administrador deja vacía la casilla del nombre, apellido, e-mail, fecha de nacimiento y/o contraseña . El sistema muestra un mensaje de error. 1.2- El administrador ingresa un e-mail ya registrado en el sistema . El sistema muestra un mensaje de error. 1.3- El administrador ingresa una contraseña menor a 4 cifras . El sistema muestra un mensaje de error. 1.4- El administrador ingresa 2 contraseñas diferentes . El sistema muestra un mensaje de error.	
Post-condición: El usuario es modificado correctamente en el sistema.	

Caso de uso 3: Eliminar Usuario

ID: CU3	
Nombre: Eliminar Usuario	
Descripción: Se quiere eliminar un usuario.	
Actores: Administrador	
Pre-condición: El Usuario fue previamente registrado en el sistema.	
Curso normal:	
Administrador	Sistema
1- El administrador selecciona el usuario y clickea en eliminar.	
	2- El sistema elimina el usuario indicado y los equipos que se quedaron sin miembros.
Curso Alternativo:	
1.1- El administrador no selecciona un usuario para eliminar. El sistema muestra un mensaje de error. 1.2- El administrador se selecciona a sí mismo para eliminar. El sistema muestra un mensaje de error.	
Post-condición: El Usuario es eliminado del sistema.	

Caso de uso 4: Registrar Equipo

ID: CU4	
Nombre: Registrar Equipo	
Descripción: Se quiere registrar un equipo.	
Actores: Administrador	
Pre-condición:	
Curso normal:	
Usuario	Sistema
1- El administrador ingresa: el nombre del equipo, una breve descripción de sus tareas, selecciona los miembros del equipo y la cantidad máxima de usuarios que pueden pertenecer al mismo.	
	2- El sistema registra el equipo
Curso Alternativo:	
1.1- El administrador no ingresa nombre del equipo, una breve descripción, no agrega a ningún usuario y/o cantidad máxima de usuarios. El sistema muestra un mensaje de error. 1.2- El administrador ingresa un nombre de equipo ya creado. El sistema muestra un mensaje de error. 1.3- El administrador ingresa valores que no sean numéricos en la cantidad máxima de usuarios. El sistema muestra un mensaje de error.	
Post-condición: El equipo es registrado en el sistema.	

Caso de uso 5: Modificar Equipo

ID: CU5	
Nombre: Modificar Equipo	
Descripción: Se quiere modificar un equipo del sistema.	
Actores: Administrador	
Pre-condición:	
Curso normal:	
Usuario	Sistema
1- El administrador selecciona el equipo a modificar y presiona el botón de modificar	
	2- El sistema muestra el User Control de modificar al usuario.
3- El administrador ingresa: el nombre del equipo, una breve descripción de sus tareas, selecciona los miembros del equipo y la cantidad máxima de usuarios que pueden pertenecer al mismo.	
	4- El sistema modifica el equipo seleccionado existente en el sistema.
Curso Alternativo:	
1.1- El administrador no selecciona a ningún equipo. El sistema muestra un mensaje de error. 3.1- El administrador no ingresa nombre del equipo, una breve	

descripción, no agrega a ningún usuario y/o cantidad máxima de usuarios.

El sistema muestra un mensaje de error.

3.2- El administrador ingresa un nombre de equipo ya creado.

El sistema muestra un mensaje de error.

3.3- El administrador ingresa valores que no sean numéricos en la cantidad máxima de usuarios.

El sistema muestra un mensaje de error.

Post-condición: El equipo es modificado correctamente en el sistema.

Caso de uso 6: Eliminar Equipo

ID: CU6	
Nombre: Eliminar Equipo	
Descripción: Se quiere eliminar un equipo.	
Actores: Administrador	
Pre-condición:	
Curso normal:	
Administrador	Sistema
1- El usuario selecciona el equipo y clickea en eliminar.	
	2- El sistema elimina el equipo indicado
Curso Alternativo:	
1.1- El usuario no selecciona un equipo para eliminar. El sistema muestra un mensaje de error.	
Post-condición: El Equipo es eliminado del sistema.	

Caso de uso 7: Informe Pizarrones creados por equipo

ID: CU7	
Nombre: Informe Pizarrones creados por equipo	
Descripción: Listar pizarrones creados por equipo con filtros	
Actores: Administrador	
Pre-condición:	
Curso normal:	
Administrador	Sistema
1- El Administrador selecciona el equipo	
	2- El sistema muestra todos los pizarrones creados por ese equipo.
3- El Administrador selecciona el filtro deseado.	
	4- El sistema muestra todos los pizarrones creados por ese equipo con el filtro adecuado.
Curso Alternativo:	
1.1- No se selecciona ningún equipo. El sistema muestra un mensaje de error. 3.1- Se selecciona un filtro de una fecha del futuro. El sistema muestra un mensaje de error.	
Post-condición: Se muestra los pizarrones creados por equipo, con el filtro especificado.	

Caso de uso 8: Comentarios Resueltos por Usuario

ID: CU8	
Nombre: Comentarios Resueltos por Usuario	
Descripción: visualizar los comentarios de un usuario.	
Actores: Usuario	
Pre-condición: Hay equipos usuarios con comentarios creados.	
Curso normal:	
Usuario	Sistema
1- El usuario selecciona un usuario de la lista.	
	2- El sistema muestra todos los comentarios resueltos de ese usuario, indicando el comentario, usuario creador y si fue resuelto
3- El usuario aplica un filtro.	
	4- El sistema muestra todos los comentarios resueltos de ese usuario que respeten el filtro aplicado, indicando su fecha de creación, usuario creador, y si fue resuelto.
Curso Alternativo:	

- 1.1- No se selecciona ningún usuario.
El sistema muestra un mensaje de error.
- 3.1- A la hora de aplicar el filtro de la fecha de creación, se elige una fecha del futuro.
El sistema muestra un mensaje de error.
- 3.2- A la hora de aplicar el filtro de la fecha de modificación, se elige una fecha del futuro.
El sistema muestra un mensaje de error.

Post-condición: Se muestran los comentarios del usuario indicado.

Caso de uso 9: Alta de Pizarrón

ID: CU9	
Nombre: Alta de Pizarrón	
Descripción: Se quiere crear un pizarrón	
Actores: Usuario	
Pre-condición: Hay usuarios y equipos creados.	
Curso normal:	
Usuario	Sistema
1- El usuario selecciona un equipo y selecciona crear pizarrón..	
	3- El sistema muestra el User Control de alta del pizarrón.
3- El usuario indica nombre del pizarrón, equipo al que pertenece, breve descripción, su ancho y alto.	
	2- El sistema procede a dar de alta el pizarrón deseado.
Curso Alternativo:	
1.1- No se seleccionó ningún equipo. El sistema muestra un mensaje de error. 1.2- Alguno de los campos está incompleto El sistema muestra un mensaje de error. 1.3- Los valores de ancho o alto son inválidos. El sistema muestra un mensaje de error.	
Post-condición: Se creo el pizarrón deseado.	

Caso de uso 10: Modificar Pizarrón

ID: CU10	
Nombre: Modificar Pizarrón	
Descripción: Se quiere modificar un pizarrón	
Actores: Usuario	
Pre-condición: Hay al menos un pizarrón creado.	
Curso normal:	
Usuario	Sistema
	1- El sistema muestra todos los pizarrones modificables por el usuario actual.
2- El usuario selecciona el pizarrón y clickea en modificar pizarrón deseado e ingresa los nuevos datos del mismo.	
	3- El sistema procede a modificar el pizarrón con los nuevos datos ingresados.
Curso Alternativo:	
1.1- No se selecciona ningún pizarrón. El sistema muestra un mensaje de error. 1.2- Alguno de los campos está incompleto El sistema muestra un mensaje de error. 1.3- Los valores de ancho o alto son inválidos. El sistema muestra un mensaje de error.	
Post-condición: Se modificó el pizarrón deseado.	

Caso de uso 11: Baja de Pizarrón

ID: CU11	
Nombre: Baja de Pizarrón	
Descripción: Se quiere eliminar un pizarrón	
Actores: Usuario	
Pre-condición: Hay al menos un pizarrón creado.	
Curso normal:	
Usuario	Sistema
	1- El sistema muestra todos los pizarrones en los que el usuario actual pertenece al equipo.
2- El usuario selecciona el pizarrón que desea eliminar.	
	3- El sistema procede a eliminar el pizarrón.
Curso Alternativo:	
1.1- No se selecciona ningún pizarrón. El sistema muestra un mensaje de error. 1.2- El usuario no es administrador ni creador del pizarrón seleccionado. El sistema muestra un mensaje de error.	
Post-condición: Se eliminó el pizarrón deseado.	

Caso de uso 12: Visualizar un Pizarrón

ID: CU12	
Nombre: Visualizar un Pizarrón	
Descripción: Se quiere ver un pizarrón	
Actores: Usuario	
Pre-condición: Hay al menos un pizarrón creado.	
Curso normal:	
Usuario	Sistema
	1- El sistema muestra todos los equipos a los que el usuario pertenece.
2- El usuario selecciona el equipo del cual quiere visualizar un pizarrón.	
	3- El sistema muestra todos los pizarrones de el equipo seleccionado.
2- El usuario selecciona el pizarrón que desea visualizar.	
	3- El sistema procede a mostrar el pizarrón.
Curso Alternativo:	
1.1- No se selecciona ningún pizarrón. El sistema muestra un mensaje de error. 1.2- No se selecciona ningún equipo. El sistema muestra un mensaje de error.	
Post-condición: Se mostró el pizarrón deseado.	

Caso de uso 13: Agregar un elemento al pizarrón

ID: CU13	
Nombre: Agregar un elemento al pizarrón.	
Descripción: Se quiere agregar un elemento al pizarrón	
Actores: Usuario	
Pre-condición: Se está visualizando el pizarrón.	
Curso normal:	
Usuario	Sistema
1-El usuario se encuentra visualizando un pizarrón y clickea agregar elemento.	
	2-El sistema verifica que el usuario esté habilitado para editar.
3- El usuario cargar una imagen o crea un cuadro de texto.	
	4- El sistema procede a agregar el elemento al panel en la esquina superior izquierda.
Curso Alternativo:	
1.1- El usuario no selecciona una foto y crea una imagen. El sistema muestra un mensaje de error. 1.1- El usuario deja algún campo vacío y crea un texto o imagen. El sistema muestra un mensaje de error.	
Post-condición: Se agregó el elemento al pizarrón.	

Caso de uso 14: Modificar un elemento de un pizarrón

ID: CU14	
Nombre: Modificar un elemento de un pizarrón.	
Descripción: Se quiere modificar un elemento del pizarrón	
Actores: Usuario	
Pre-condición: Se está visualizando el pizarrón.	
Curso normal:	
Usuario	Sistema
1-El usuario selecciona un equipo al que pertenece.	
	2-El sistema muestra todos los pizarrones del equipo seleccionado.
3- El usuario selecciona modificar pizarrón y modificar el pizarrón.	
	4- El sistema procede a mostrar el contenido de modificar el pizarrón que se le indicó.
5- El usuario ingresa un nombre, descripción o cambia los valores de la dimensión del pizarrón	
	6-El sistema procede a modificar los datos ingresados del pizarrón indicado.
Curso Alternativo:	

- | |
|---|
| <p>1.1- El usuario no selecciona un equipo.
El sistema muestra un mensaje de error.</p> <p>1.2- El usuario no selecciona un pizarrón.
El sistema muestra un mensaje de error.</p> |
| <p>Post-condición: Se agregó el elemento al pizarrón.</p> |

Caso de uso 15: Agregar un comentario a un elemento

ID: CU15	
Nombre: Agregar un comentario a un elemento.	
Descripción: Se quiere Agregar un comentario a un elemento.	
Actores: Usuario	
Pre-condición: Se está visualizando el pizarrón.	
Curso normal:	
Usuario	Sistema
1-El usuario selecciona Gestionar comentario.	
	2- El sistema muestra una lista de los elemento indicando su descripción.
3- El usuario selecciona el elemento a comentar.	
	4- El sistema muestra el recuadro para ingresar un comentario.
5- El usuario escribe su comentario y selecciona aceptar.	
	6- El sistema procede a agregar el comentario al elemento.
Curso Alternativo:	
1.1- El usuario no selecciona un elemento de la lista. El sistema muestra un mensaje de error. 1.2- El usuario ingresa un comentario demasiado corto. El sistema muestra un mensaje de error.	

Post-condición: Se agregó el comentario al elemento.

Caso de uso 16: Marcar comentario como realizado.

ID: CU16	
Nombre: Marcar comentario como realizado.	
Descripción: Se quiere indicar que un comentario ya ha sido tomado en cuenta	
Actores: Usuario	
Pre-condición: Se está visualizando el pizarrón.	
Curso normal:	
Usuario	Sistema
1-El usuario selecciona Gestionar comentario.	
	2- El sistema muestra una lista de los elemento indicando su descripción
3- El usuario selecciona el elemento del que desea resolver un comentario.	
	4- El sistema procede a mostrar todos los comentarios del elemento.
5- El usuario selecciona el comentario y clickea en el botón resolver.	
	6- El sistema marca ese comentario como realizado.
Curso Alternativo:	
1.1- El usuario no selecciona un elemento. El sistema muestra un mensaje de error. 1.2- El usuario no selecciona un comentario.	

<p>El sistema muestra un mensaje de error.</p> <p>1.3- El comentario seleccionado ya está resuelto.</p> <p>El sistema muestra un mensaje de error.</p>
<p>Post-condición: Se marcó el comentario como realizado.</p>

Caso de uso 17: Imprimir a pdf.

ID: CU17	
Nombre: Imprimir a pdf.	
Descripción: Se quiere imprimir un pizarrón a un archivo pdf	
Actores: Usuario	
Pre-condición: Se está en Menú principal	
Curso normal:	
Usuario	Sistema
1-El usuario selecciona la opción cerrar sesión.	
	2- El sistema muestra la ventana de inicio de se
Curso Alternativo:	
Post-condición: Se imprimó el pizarrón.	

Caso de uso 18: Eliminar elemento del pizarrón.

ID: CU18	
Nombre: Eliminar elemento del pizarrón.	
Descripción: Se quiere indicar que un comentario ya ha sido tomado en cuenta	
Actores: Usuario	
Pre-condición: Se está visualizando el pizarrón.	
Curso normal:	
Usuario	Sistema
1-El usuario selecciona Eliminar elemento.	
	2- El sistema muestra una lista de los elemento indicando su descripción
3- El usuario selecciona el elemento que desea eliminar.	
	4- El sistema procede a eliminar el elemento seleccionado.
Curso Alternativo:	
1.1- El usuario no selecciona un elemento. El sistema muestra un mensaje de error.	
Post-condición: Se eliminó el elemento seleccionado de la lista de elementos del pizarrón.	

Caso de uso 19: Iniciar Sesión.

ID: CU19	
Nombre: Iniciar Sesión.	
Descripción: Se quiere iniciar sesión	
Actores: Usuario	
Pre-condición: se está en la ventana de inicio de sesión..	
Curso normal:	
Usuario	Sistema
1-El usuario selecciona un usuario e ingresa la contraseña.	
	2- El sistema chequea que la contraseña sea correcta y procede a mostrar el menú principal.
Curso Alternativo:	
1.1- El usuario ingresa una contraseña que no es correcta. El sistema muestra un mensaje de error.	
Post-condición: Se logeó el usuario correctamente y se encuentra en el menú principal.	

Caso de uso 20: Cerrar Sesión.

ID: CU20	
Nombre: Cerrar Sesión.	
Descripción: Se quiere iniciar sesión	
Actores: Usuario	
Pre-condición: se está en el menú principal.	
Curso normal:	
Usuario	Sistema
1-El usuario selecciona Cerrar sesión.	
	2- El sistema muestra la ventana de iniciar sesión.
Curso Alternativo:	
Post-condición: Se está en la ventana de iniciar sesión	

Caso de uso 21: Generar una contraseña.

ID: CU21	
Nombre: Generar una contraseña.	
Descripción: Se está modificando un usuario	
Actores: Usuario	
Pre-condición: se está en el menú principal.	
Curso normal:	
Usuario	Sistema
1-El usuario selecciona Generar contraseña.	
	2- El sistema genera una contraseña aleatoria.
Curso Alternativo:	
Post-condición: Se inserta en el campo contraseña y repetir contraseña la misma contraseña generada	

Caso de uso 22: Reiniciar puntajes.

ID: CU22	
Nombre: Reiniciar Puntajes	
Descripción: Se está modificando los puntajes de equipo	
Actores: Administrador	
Pre-condición: se está en el menú de equipos.	
Curso normal:	
Usuario	Sistema
1-El administrador selecciona un equipo y presiona el botón “reiniciar puntaje”.	
	2- El sistema desplegará 5 campos para que el administrador rellene los puntajes nuevos.
3-El administrador completa los campos y presiona el botón de “guardar cambios”	
	4- El sistema guarda los cambios
Curso Alternativo:	
1.1- El usuario no selecciona ningún equipo y presiona el botón “reiniciar puntaje”. El sistema muestra un mensaje de error. 3.1- El usuario no ingresa nada en los campos seleccionados o ingresa valores no válidos.	
Post-condición: Se reinician los valores de puntaje para el equipo seleccionado	

Caso de uso 23: Ver puntajes.

ID: CU23	
Nombre: Ver Puntajes	
Descripción: Se observan los puntajes de equipo	
Actores: Usuario	
Pre-condición: se está parado en el menú de equipos.	
Curso normal:	
Usuario	Sistema
1-El usuario selecciona un equipo y presiona el botón “ver puntajes”.	
	2- El sistema desplegará el puntaje del equipo seleccionado.
Curso Alternativo:	
1.1- El usuario no selecciona ningún equipo y presiona el botón “ver puntajes”. El sistema muestra un mensaje de error.	
Post-condición: Se despliega en pantalla el puntaje del equipo seleccionado	

Caso de uso 24: Crear conexiones

ID: CU24	
Nombre: Crear conexiones	
Descripción: Se crean conexiones entre elementos	
Actores: Usuario	
Pre-condición: Se está parado en el menú visualizar pizarrón de un pizarrón que tenga al menos 2 elementos.	
Curso normal:	
Usuario	Sistema
1-El usuario selecciona un elemento y presiona botón derecho y selecciona la opción de crear conexión.	
	2- El sistema desplegara un listado de todos los elementos que pertenezcan en el mismo pizarrón y se selecciona el tipo de conexión.
3- El usuario selecciona la conexión deseada.	
	4- El sistema genera la conexión.
Curso Alternativo:	
1.1- El usuario no selecciona ningún equipo y presiona el botón “ver puntajes”. El sistema muestra un mensaje de error.	
Post-condición: Se despliega en pantalla el puntaje del equipo seleccionado	

Diagrama de paquetes

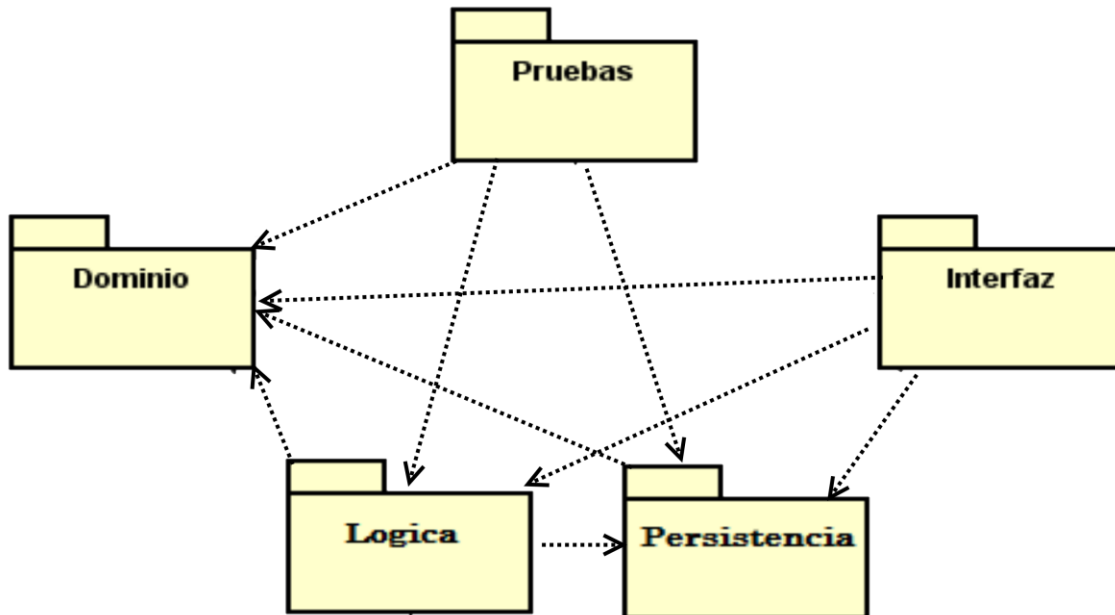
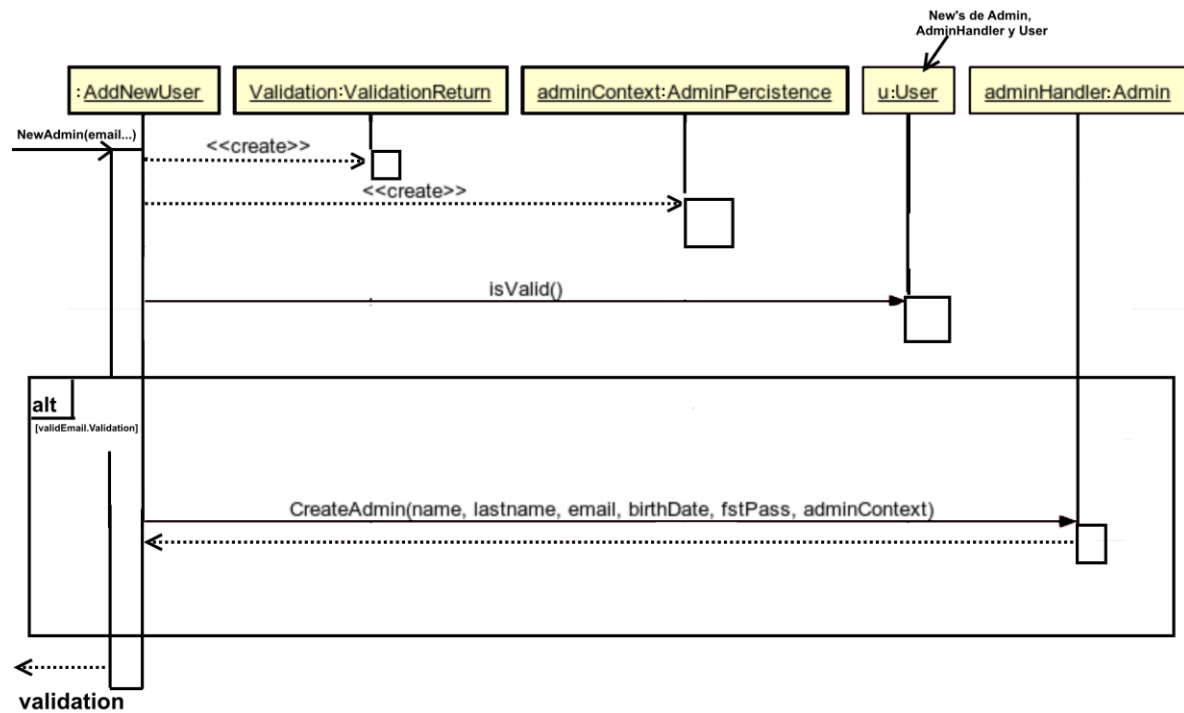


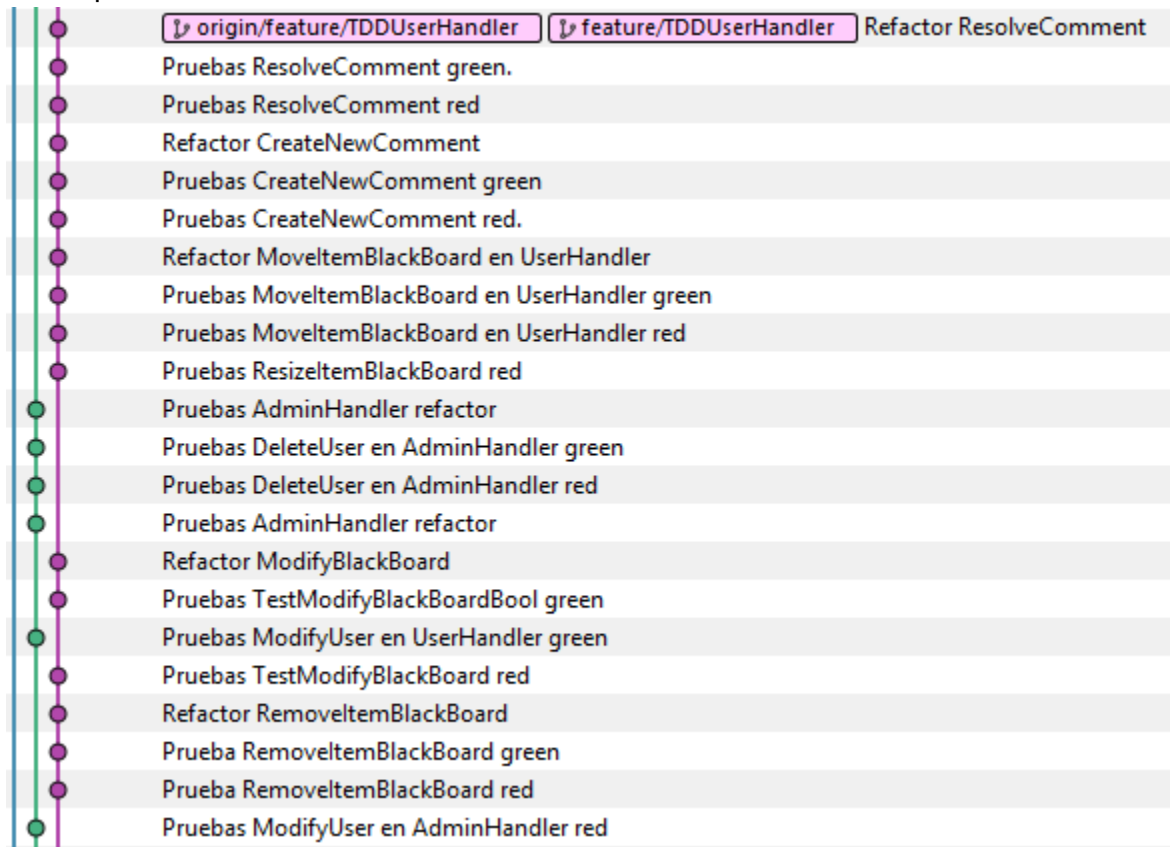
Diagrama de secuencia



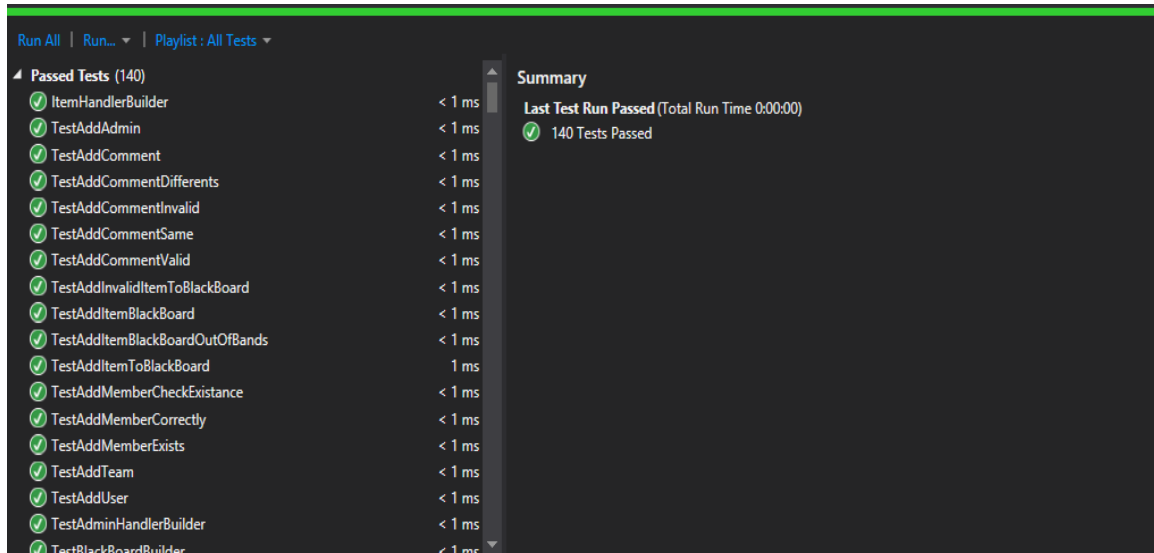
Decidimos realizar el diagrama de secuencia del Alta de Usuario ya que es un método importante para el sistema.

Evidencia TDD:

En el repositorio se puede ver la evidencia del uso de TDD. Se ven los 4 pasos que utilizamos. Primero crear la prueba, lo que generará que la solución no compile. Luego, en la fase roja, hacemos que compile el proyecto pero no implementamos nada más, lo que dará la prueba en roja, ya que ese método no estará implementado. Una vez esto, implementamos lo mínimo necesario para que la prueba pase, lo que llamamos la fase verde. Por último, hacemos refactoring, en el caso que sea necesario, para utilizar las buenas prácticas de Clean Code.



Copiamos una imagen del repositorio en la cual se puede observar la metodología red,green refactor. Red es que no compila, green compila y las pruebas corren correctamente, y refactor, se aplica clean code en el código.



Clean Code

Practicas aplicadas

Para el desarrollo de la solución, se aplicaron varias de las prácticas descritas en el libro de Clean Code que serán enumeradas a continuación:

1. **Nombres nemotécnicos:** Se buscó utilizar nombres lo más representativos posible para cada uno de los atributos, métodos, clases y paquetes.
2. **Uso de comentarios:** Se evitó totalmente el uso de comentarios, creemos que esto ayuda a que el código se mantenga más prolijo.
3. **Consistencia de nombres:** Se buscó utilizar siempre la misma palabra para describir el mismo concepto a lo largo del código.
4. **Formateo vertical:** Se buscó que los métodos estén agrupados de forma lógica y que el código se pueda leer de arriba hacia abajo. Las clases se organizan primero constantes, luego variables y finalmente métodos.
5. **SRP:** Se aplicó el principio de responsabilidad única para así conseguir un mayor nivel de cohesión en las clases.

Evidencia de Clean Code

```
1 reference | 0 changes | 0 authors, 0 changes
private bool validName()
{
    return (this.name.Length >= 0);
}
1 reference | 0 changes | 0 authors, 0 changes
private bool validDescription()
{
    return (this.description.Length >= 0);
}
1 reference | 0 changes | 0 authors, 0 changes
private bool validMaxUsers()
{
    return (this.maxUsers > 0);
}
1 reference | 0 changes | 0 authors, 0 changes
private bool validMembers()
{
    return (this.members.Count > 0);
}
1 reference | 0 changes | 0 authors, 0 changes
private bool validCantOfMembers()
{
    return (this.members.Count <= this.maxUsers);
}
```

Vamos a utilizar el ejemplo de la clase de Equipo para mostrar algunos conceptos de Clean code. Primero, como se puede apreciar, ninguno de los métodos es de más de una pantalla de largo, o recibe más de tres parámetros (salvo el constructor).

Extrajimos las validaciones que realiza a cinco métodos auxiliares “validName”, “validDescription”, “validMaxUsers”, “validMembers” y “validCantOfMembers” que se ocupan de chequear que los valores sean los correctos.

Luego “IsValid” el cual se encarga de llamar a los métodos e indicar si cumple con las restricciones y devolviendo un mensaje de error si es necesario. Esto brinda facilidad a la hora de cambiar el comportamiento, dado que si luego surge alguna otra restricción, únicamente se tendrá que modificar el código de los métodos auxiliares o agregar algún método auxiliar más.

Se puede apreciar que todos los nombres utilizados, tanto para métodos como para variables son nemotécnicos, y se utilizó un único idioma que fue el inglés.

Además para el manejo de errores creamos una clase llamada ValidationReturn con el principal objetivo de disminuir las responsabilidades de la interfaz como se nos fue indicado en la entrega anterior.

```
public class Dimension
{
    private int width;
    private int height;
    47 references | 29/29 passing | FABIANG-PC\Fabian, 35 days ago | 1 author, 1 change
    public Dimension(int aWidth, int aHeight)
    {
        this.width = aWidth;
        this.height = aHeight;
    }
}

public class Coordinate
{
    private int xAxis;
    private int yAxis;
    8 references | 5/5 passing | Federico Cetraro, 35 days ago | 1 author, 1 change
    public Coordinate()
    {
        this.xAxis = 0;
        this.yAxis = 0;
    }
}
```

Por último, queríamos destacar que generamos clases tales como Dimension, Coordinate y Score para minimizar parámetros de algunas clases.

Supuestos:

- Cuando se borra un usuario, si algún equipo se queda sin miembros, dicho equipo es eliminado automáticamente.
- Las conexiones de los elementos es 1 a 1.

Funcionalidades no implementadas:

Por motivos de tiempo decidimos priorizar en el la persistencia de la Base de Datos sobre las funcionalidades nuevas que fueron pensadas y se comenzaron a implementar pero no se llegaron a terminar.

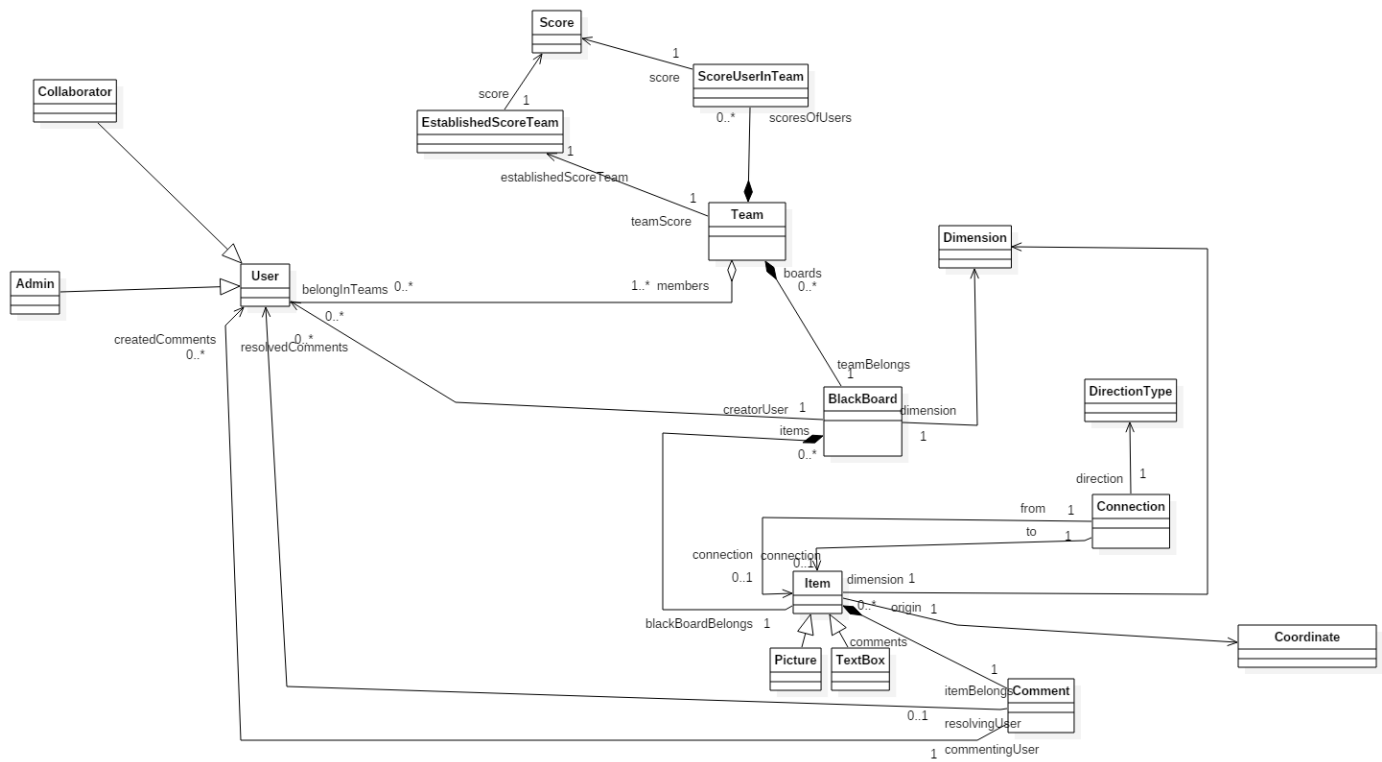
Por un lado está los puntajes de los equipos: para quisimos decidimos implementarlo con 3 clases más:

1. Score: clase que simplemente guarda los 5 puntajes que son pedidos en la letra para tener en cada Team.
2. ScoreUsersInTeam: clase que relaciona los usuarios con los equipos y el Score generado (suma 1 por cada vez que realiza algo de los 5 puntajes).
3. El EstablishedScoreTeam funcionaria como “multiplicador” de los ScoreUsersInTeam de manera que se calcula el puntaje de cada una de las tareas dependiendo de cómo este establecido en cada equipo.

Por otro lado, las conexiones las supusimos como relaciones 1 a 1, que cada elemento puede estar conectado a 0 o a 1 elemento distinto de sí mismo. Teníamos un a clase Connection la cual tenía Item to y Item from. En la base de datos, guardaba el ID_To y ID_From.

UML

Diagrama de clases



Manual de instalación

1. Pegar el contenido de la carpeta release en cualquier directorio del pc.
2. Modificar el connection string que se encuentra en exe.config para que utilice la base de datos local
3. La base de datos se autogenera.
4. Correr el programa.
5. Por defecto, en el sistema cuando se genera vacío viene cargado un usuario administrador con las siguientes credenciales:
 - Email "admin@admin"
 - Contraseña "admin"