Tecnologías para la Construcción de Software

16/12/21

Integrantes

Gustavo Flores Romero  
Andrea Soto Portilla  
Carlos Miguel Pérez Pérez

Especificación de requerimientos

CONECTA 4

Contenido

[1. Introducción 2](#_Toc90551526)

[2. Requisitos 2](#_Toc90551527)

[3. Modelo de dominio 4](#_Toc90551528)

[4. Casos de Uso 5](#_Toc90551529)

[a. Diagrama de Casos de Uso 5](#_Toc90551530)

[b. Descripciones de Caso de Uso 6](#_Toc90551531)

[i. CU01 – Iniciar Sesión 6](#_Toc90551532)

[ii. CU02 – Registrar Jugador 7](#_Toc90551533)

[iii. CU03 – Enviar Invitación 9](#_Toc90551534)

[iv. CU04 – Unirse al juego 10](#_Toc90551535)

[v. CU05 – Consultar Tabla de Puntajes 11](#_Toc90551536)

[vi. CU06 – Monitorear Presencia 12](#_Toc90551537)

[vii. CU07 – Mandar Mensaje en Tiempo Real (Chat) 13](#_Toc90551538)

[viii. CU08 – Partida 14](#_Toc90551539)

[ix. CU09 – Cambiar Idioma 16](#_Toc90551540)

[x. CU10 – Desactivar Sonidos 16](#_Toc90551541)

[5. Framework de desarrollo 17](#_Toc90551542)

[6. Análisis estático de código 18](#_Toc90551543)

[7. Resultado de las pruebas 20](#_Toc90551544)

[8. Conclusiones 21](#_Toc90551545)

# Introducción

La siguiente especificación de requerimientos está enfocada en el juego desarrollado “Conecta 4”. El juego consiste en un tablero de 7 columnas por 6 filas, donde participan dos jugadores donde el objetivo de cada uno es conectar 4 fichas de su color (fichas azules) antes de que su oponente lo haga (fichas rojas). La conexión de las 4 fichas puede ser en cualquier dirección, ya sea de forma horizontal, vertical o en diagonal. El jugador que cumpla con el objetivo gana y se le otorgan 100 puntos; en caso de que el tablero se llene, la partida terminará en un empate, y ningún jugador recibirá puntos.

# Requisitos

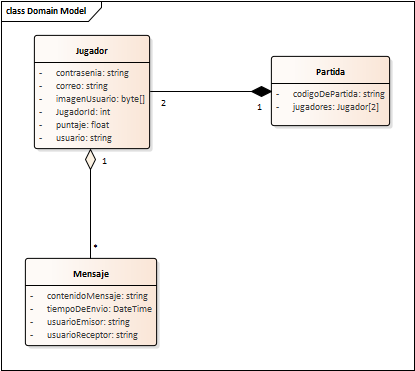
Los requisitos que fueron planteados para el desarrollo del juego son los siguientes:

1. REQ01 – Confirmación de registro de usuario vía email
   * El sistema debe enviar un email con un código de confirmación de registro al email ingresado por el usuario. El registro será exitoso si el código ingresado por el usuario coincide con el código enviado al email ingresado anteriormente
2. REQ02 – Inicio de sesión
   * Para poder ingresar al juego, se requiere una cuenta registrada previamente. El usuario debe ingresar un usuario y una contraseña válidos para que el sistema permita el inicio de sesión
3. REQ03 – Envío de invitaciones
   * El jugador puede crear una partida y el sistema debe ser capaz de mandar una invitación por email al usuario ingresado por el usuario que crea la partida
4. REQ04 – Unirse al juego
   * Un jugador puede unirse a una partida creada anteriormente ingresando el código de la partida recibido en el correo de invitación
5. REQ05 – Tablas de puntaje
   * El jugador puede consultar los 10 mejores puntajes del juego
6. REQ06 – Monitoreo de presencia
   * El sistema debe monitorear el tiempo de inactividad de cada jugador con el tiempo de inactividad establecido (30 segundos)
7. REQ07 – Comunicación en tiempo real
   * El sistema debe permitir la comunicación en tiempo real entre jugadores; debe existir un chat global y un chat durante la partida

También se definieron los siguientes atributos de calidad con los que debe contar el sistema:

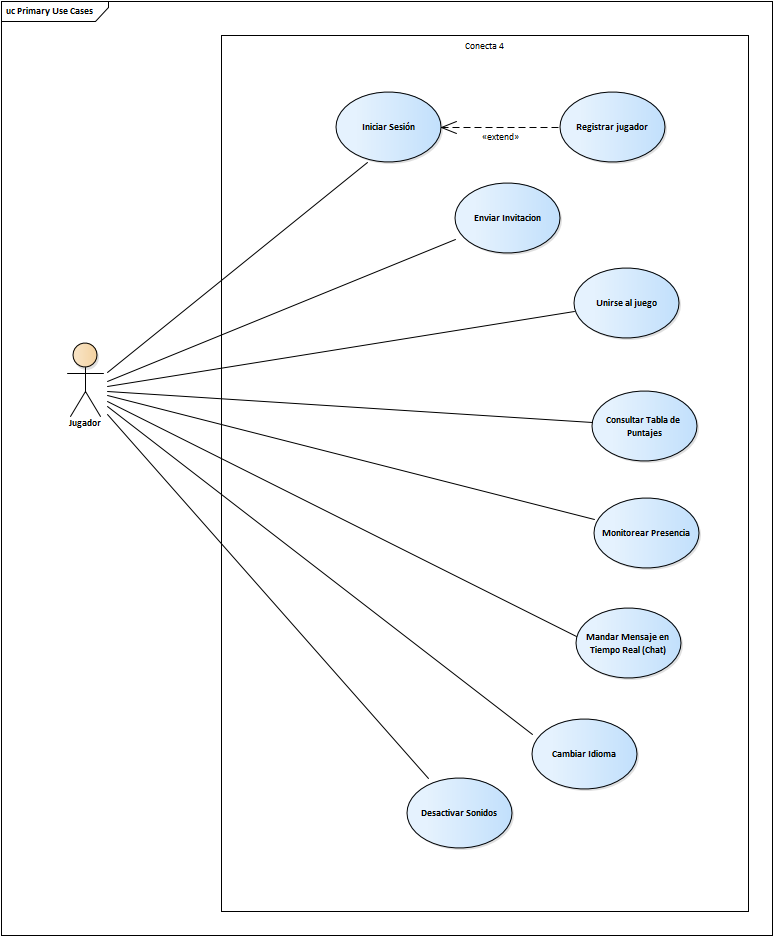
1. AC01 – Seguridad (Confidencialidad)
   1. La contraseña del usuario registrado se debe encontrar protegida utilizando algún método de encriptación
2. AC02 – Usabilidad (Capacidad de aprendizaje)
   1. Los jugadores son capaces de utilizar el juego en su totalidad sin dificultades; esto quiere decir que la interfaz de usuario debe ser amigable para jugadores nuevos

# Modelo de dominio



# Casos de Uso

# Diagrama de Casos de Uso



# Descripciones de Caso de Uso

# CU01 – Iniciar Sesión

|  |  |
| --- | --- |
| **ID:** | CU-01 |
| **Nombre:** | Iniciar Sesión |
| **Descripción:** | Se desea iniciar sesión en el juego |
| **Actor(es):** | Jugador |
| **Disparador:** | El **jugador** ejecuta el juego “Conecta 4” |
| **Precondiciones:** | PRE01-El usuario tiene una cuenta existente |
| **Flujo Normal:** | 1. El juego muestra la ventana GUI “IniciarSesion”, con los campos de usuario y contraseña, junto con los botones de “Iniciar sesión” y “Registrarse” 2. El **jugador** ingresa su usuario y contraseña y da clic en el botón “Iniciar sesión” (FA-01) 3. El juego valida los campos e inicia sesión del usuario verificando las credenciales con la base de datos (FA-02, FA-03 y EX-01) 4. El juego abre la GUI “Menú Principal” 5. Finaliza el Caso de Uso |
| **Flujos Alternos:** | FA-01: Registrarse   1. El **jugador** en el botón “Registrarse” 2. Se incluye el CU-02: “Registro de jugador” 3. Finaliza el caso de uso   FA-02: Usuario y/o contraseña incorrecta(s).   1. El juego muestra el mensaje de contrasñas incorrectas 2. El **jugador** da clic en “Aceptar” 3. Se regresa al paso 2 del flujo normal   FA-03: Campos vacíos necesarios   1. El juego muestra el mensaje de campos vacíos 2. El **jugador** da clic en “Aceptar” 3. Se regresa al paso 2 del flujo normal |
| **Excepciones:** | EX-01 No hay conexión con la base de datos   1. El juego muestra el mensaje de error en la conexión a la base de datos 2. El **jugador** da clic en “Aceptar” 3. Se regresa al paso 2 del flujo normal |
| **Postcondiciones:** | POST-01 Se ha iniciado la sesión con éxito |

# CU02 – Registrar Jugador

|  |  |
| --- | --- |
| **ID:** | CU02 |
| **Nombre:** | Registro de jugador |
| **Descripción:** | Se desea registrar una cuenta en el juego |
| **Actor(es):** | **Jugador** |
| **Disparador:** | El **jugador** da clic en el botón de “Registrarse” |
| **Precondiciones:** | PRE-01: El **jugador** no tiene una cuenta registrada |
| **Flujo Normal:** | 1. El juego muestra la ventana GUI “Registro de jugador” con los campos correo, usuario, contraseña y validación de contraseña, junto con el botón “Registrarse” y “Cancelar” 2. El **Jugador** ingresa el correo, el usuario, su contraseña y la validación de la contraseña y da clic en el botón “Registrarme” (FA-01) 3. El juego valida los campos (FA-02, FA-03, FA-04, FA-05 y FA-06) 4. El juego manda un correo para la validación del registro (EX-01 y EX-02) 5. El jugador ingresa el código recibido en la GUI “RegistroDeJugadorInputDeCodigo” y da clic en “Aceptar” (FA-01) 6. El sistema valida el código ingresado con el código enviado y registra el jugador en la base de datos (EX01 y EX02) 7. Finaliza caso de uso |
| **Flujos Alternos:** | FA-01 Clic botón cancelar   1. Termina caso de uso 2. Se regresa al CU-01   FA-02 Campos vacíos necesarios   1. El juego muestra el mensaje de campos vaciós 2. Se regresa al paso 2 del flujo normal   FA-03 Correo inválido   1. El juego muestra el mensaje de correo inválido 2. El **jugador** da clic en “Aceptar” 3. Se regresa al paso 2 del flujo normal   FA-04 Correo en uso   1. El juego muestra el mensaje de correo en uso 2. El **jugador** da clic en “Aceptar” 3. Se regresa al paso 2 del flujo normal   FA-05 Usuario en uso   1. El juego muestra el mensaje de usuario en uso 2. El **jugador** da clic en “Aceptar” 3. Se regresa al paso 2 del flujo normal   FA-06 Contraseñas no coinciden   1. El juego muestra el mensaje de contraseñas no coinciden 2. El **jugador** da clic en “Aceptar” 3. Se regresa al paso 2 del flujo normal |
| **Excepciones:** | EX-01 No hay conexión con la base de datos   1. El juego muestra el mensaje de error de conexión en la base de datos 2. El **jugador** da clic en “Aceptar” 3. Se regresa al paso 2 del flujo normal   EX-02 No hay conexión de internet   1. El juego muestra el mensaje de error de conexión a internet 2. El **jugador** da clic en “Aceptar” 3. Finaliza el caso de uso |
| **Postcondiciones:** | POST-01 Se ha registrado el jugador exitosamente |

# CU03 – Enviar Invitación

|  |  |
| --- | --- |
| **ID:** | CU-03 |
| **Nombre:** | Envío de invitación |
| **Descripción:** | Se desea enviar una invitación a otro jugador |
| **Actor(es):** | **Jugador** |
| **Disparador:** | El **jugador** da clic en el botón “Crear partida” de la GUI “Menú principal” |
| **Precondiciones:** | PRE01-El usuario tiene una sesión iniciada en el juego |
| **Flujo Normal:** | 1. El juego muestra la ventana GUI “EnviarInvitación”, con el campo del usuario al que se quiere invitar y el botón “Invitar” 2. El **jugador** ingresa el usuario del jugador que quiera invitar y da clic en el botón “Invitar” 3. El juego valida el usuario y envía la INVITACIÓN (código) al usuario proporcionado (FA-01, FA-02, EX-01 y EX-02) 4. El juego abre la GUI “Ventana del juego” y se queda en espera del otro **jugador** 5. Finaliza el caso de uso |
| **Flujos Alternos:** | FA-01 Usuario inválido   1. El juego muestra el mansaje de usuario no encontrado 2. El **jugador** da clic en “Aceptar” 3. Se regresa al paso 2 del flujo normal   FA-02 Campos vacíos necesarios   1. El juego muestra el mansaje de campos vacíos 2. El **jugador** da clic en “Aceptar” 3. Se regresa al paso 2 del flujo normal |
| **Excepciones:** | EX-01 No hay conexión con la base de datos   1. El juego muestra el mensaje de error de conexión a la base de datos 2. El **jugador** da clic en “Aceptar” 3. Se regresa al paso 2 del flujo normal   EX-02 No hay conexión de internet   1. El juego muestra el mensaje de error de conexión a internet 2. El **jugador** da clic en “Aceptar”   Finaliza el caso de uso |
| **Postcondiciones:** | POST-01 Se ha mandado la INIVTACIÓN exitosamente |

# CU04 – Unirse al juego

|  |  |
| --- | --- |
| **ID:** | CU-04 |
| **Nombre:** | Unirse al juego |
| **Descripción:** | Se desea ingresar a una PARTIDA con un código recibido |
| **Actor(es):** | Jugador |
| **Disparador:** | El **jugador** da clic en el botón “UnirseApartida” de la GUI-XX “Menú principal” |
| **Precondiciones:** | PRE01-El **jugador** que se quiere unir debe estar registrado en el juego  PRE02-El **jugador** debe haber iniciado sesión en el juego  PRE03-El **jugador** debe tener un código de invitación a una partida |
| **Flujo Normal:** | 1. El juego muestra la GUI “UnirseAPartida” con el campo de código de INVITACIÓN y el botón “Ingresar” 2. El **jugador** ingresa el código que recibió en el correo electrónico 3. El **jugador** da clic en “Ingresar” 4. El juego valida el código ingresado (FA-01, FA-02, EX-01) 5. El juego muestra la GUI “VentanaDeJuego” 6. Finaliza el caso de uso |
| **Flujos Alternos:** | FA-01 Código inválido   1. El juego muestra el mensaje de código inválido 2. El **jugador** da clic en “Aceptar” 3. Se regresa al paso 2 del flujo normal   FA-02 Campos vacíos necesarios   1. El juego muestra el mensaje de campos vacíos 2. El **jugador** da clic en “Aceptar”   Se regresa al paso 2 del flujo normal  FA-03 Partida llena   1. El juego muestra el mensaje de partida llena 2. El **jugador** da clic en “Aceptar” 3. Finaliza el caso de uso |
| **Excepciones:** | EX-01 No hay conexión de internet   1. El juego muestra el mensaje de error de conexión a internet” 2. Finaliza el caso de uso |
| **Postcondiciones:** | POST-01 Se ha unido la PARTIDA de manera exitosa |

# CU05 – Consultar Tabla de Puntajes

|  |  |
| --- | --- |
| **ID:** | CU-05 |
| **Nombre:** | Tablas de puntajes |
| **Descripción:** | El **jugador** puede consultar los puntajes de los mejores jugadores en forma de tabla |
| **Actor(es):** | Jugador |
| **Disparador:** | El **jugador** da clic en el botón “Tabla de puntajes” de la GUI-XX “Menú principal” |
| **Precondiciones:** | PRE01-El **jugador** debe tener una sesión iniciada |
| **Flujo Normal:** | 1. El juego muestra la ventana GUI “TablaDePuntaje” 2. El juego recupera de la base de datos los primeros 20 JUGADOR con la puntuación global más alta y los muestra en una tabla (EX-01 y EX-02) 3. El **jugador** da clic en el botón “X” 4. Finaliza el caso de uso |
| **Flujos Alternos:** |  |
| **Excepciones:** | EX-01 No hay conexión de internet   1. El juego muestra el mensaje de error de conexión a internet 2. Finaliza el caso de uso   EX-01 No hay conexión con la base de datos   1. El juego muestra el mensaje de error de conexión a la base de datos 2. Finaliza el caso de uso |
| **Postcondiciones:** | POST-01 El jugador ha consultado la TABLA DE PUNTAJES exitosamente |

# CU06 – Monitorear Presencia

|  |  |
| --- | --- |
| **ID:** | CU-06 |
| **Nombre:** | Monitoreo de presencia |
| **Descripción:** | El juego detecta cuando el **jugador** se ausenta de la PARTIDA en curso |
| **Actor(es):** | Jugador |
| **Disparador:** | El **jugador** se ausenta de la PARTIDA en curso por más de 30 segundos |
| **Precondiciones:** | PRE01-El **jugador** debe estar en una PARTIDA en curso |
| **Flujo Normal:** | 1. El **jugador** deja de jugar por más de 30 segundos 2. El juego muestra la GUI-XX “ConfirmacionDePresencia” con el botón “Sigo aquí” 3. El **jugador** da clic en el botón “Sigo aquí” de la GUI “JugadorAusente” (FA-01) 4. El juego continúa 5. Finaliza el caso de uso |
| **Flujos Alternos:** | FA-01 Tiempo de espera de confirmación agotado (1 minuto)   1. El juego finaliza la PARTIDA y muestra la GUI “MenuPrincipal” 2. Finaliza el caso de uso |
| **Excepciones:** | EX-01 No hay conexión de internet   1. El juego muestra el mensaje de error de conexión a internet” 2. El **jugador** da clic en “Aceptar” 3. Finaliza el caso de uso |
| **Postcondiciones:** |  |
|  |  |

# CU07 – Mandar Mensaje en Tiempo Real (Chat)

|  |  |
| --- | --- |
| **ID:** | CU-07 |
| **Nombre:** | Mandar Mensaje en Tiempo Real (Chat) |
| **Descripción:** | El **jugador** puede comunicarse con el otro JUGADOR desde el chat. |
| **Actor(es):** | Jugador |
| **Disparador:** | El **jugador** da clic en el botón del chat dentro de una partida de la GUI “VentanaDeJuego” |
| **Precondiciones:** | PRE01-El **jugador** debe estar en una partida iniciada |
| **Flujo Normal:** | 1. El juego muestra la ventana GUI “Chat” con una tabla donde se muestra la conversación, un TextBox y un botón “Enviar” 2. El **jugador** escribe un mensaje en el TextBox y da clic en el botón “Enviar” (EX-01) 3. El juego envía el mensaje 4. Finaliza el caso de uso |
| **Flujos Alternos:** |  |
| **Excepciones:** | EX-01 No hay conexión de internet   1. El juego muestra el mensaje de error de conexión a internet 2. Finaliza el caso de uso |
| **Postcondiciones:** | POST-01 Se envió el mensaje exitosamente |

# CU08 – Partida

|  |  |
| --- | --- |
| **ID:** | CU-08 |
| **Nombre:** | Partida |
| **Descripción:** | El **jugador** puede jugar la partida con otro **jugador** |
| **Actor(es):** | **2 jugadores** |
| **Disparador:** | El **jugador** da clic en el botón Crear partida de la GUI “MenúPrincipal” e invita al otro **jugador;** El segundo **jugador** da clic en el botón “UnirseAPartida" e introduce el código de partida. |
| **Precondiciones:** | PRE01-Ambos **jugadores** deben haber iniciado sesión |
| **Flujo Normal:** | 1. El juego muestra la ventana GUI “VentanaDeJuego” con el TABLERO de juego, el botón salir y el botón de CHAT 2. El primer **jugador** da clic en la columna donde quiere poner la ficha. (FA-01, FA-02, FA-03 y FA-04) 3. El juego inserta la FICHA donde se seleccionó. 4. El segundo **jugador** da clic en la columna donde quiere poner la FICHA (FA-01, FA-02, FA-03 y FA-04) 5. El juego inserta la FICHA donde se seleccionó 6. Cuando un **jugador** cumpla la regla del ganador, se muestra la GUI-XX “Ganador” y se agregan puntos al puntaje global del jugador; al **jugador** que pierde se le muestra el mensaje “¡Has perdido!” (EX01 y EX02) 7. Finaliza el caso de uso |
| **Flujos Alternos:** | FA01 Inactividad de jugador   1. El juego detecta que uno de los JUGADOR no ha tenido actividad en 30 segundos 2. Se extiende al CU-06   FA02 – Tablero lleno   1. Se muestra el mensaje “Empate” 2. Finaliza partida 3. Finaliza el caso de uso   FA03 – Clic en salir   1. Se regresa al menú principal 2. Finaliza caso de uso   FA04 – Clic en chat   1. Se extiende al CU-07 |
| **Excepciones:** | EX-01 No hay conexión de internet   1. El juego muestra el mensaje de error de conexión a internet 2. Finaliza el caso de uso   EX-02 No hay conexión con la base de datos   1. El juego muestra el mensaje de error de conexión de base de datos 2. Finaliza el caso de uso |
| **Postcondiciones:** | POST-01 El **jugador** ganador aumentó su PUNTAJE global |

# CU09 – Cambiar Idioma

|  |  |
| --- | --- |
| **ID:** | CU-09 |
| **Nombre:** | Cambiar idioma |
| **Descripción:** | El **jugador** puede cambiar el idioma del juego |
| **Actor(es):** | **Jugador** |
| **Disparador:** | El **jugador** da clic en el botón Ajustes de la GUI “MenuPrincipal” |
| **Precondiciones:** | PRE01-El **jugador** debe tener una sesión iniciada |
| **Flujo Normal:** | 1. El juego muestra la ventana GUI “Configuracion” con las opciones de los idiomas Español, Inglés, Francés y Portugués 2. El **jugador** selecciona el idioma que quiere aplicar 3. El juego cambia el idioma de las ventanas 4. Finaliza el caso de uso |
| **Flujos Alternos:** |  |
| **Excepciones:** |  |
| **Postcondiciones:** | POST-01 El **jugador** cambia el idioma del juego |

# CU10 – Desactivar Sonidos

|  |  |
| --- | --- |
| **ID:** | CU-10 |
| **Nombre:** | Desactivar sonidos |
| **Descripción:** | El **jugador** puede desactivar los sonidos y la música del juego |
| **Actor(es):** | **Jugador** |
| **Disparador:** | El **jugador** da clic en el botón Ajustes de la GUI “MenuPrincipal” |
| **Precondiciones:** | PRE01-El **jugador** debe tener una sesión iniciada |
| **Flujo Normal:** | 1. El juego muestra la ventana GUI “Configuracion” con las opciones sonido y música 2. El **jugador** activa o desactiva el sonido y/o la música 3. El juego activa o desactiva el sonido y/o la música 4. Finaliza el caso de uso |
| **Flujos Alternos:** |  |
| **Excepciones:** |  |
| **Postcondiciones:** | POST-01 El **sistema** activa o desactiva los sonidos y/o la música |

# Framework de desarrollo

Para el desarrollo del juego Conecta 4, utilizamos el framework “Entity Framework” en conjunto con las tecnologías de WPF y WCF.

**Entity Framework**

Entity framework es un framework de Mapeo Relacional de Objetos (ORM) que permite la interacción entre aplicaciones .NET y bases de datos relacionales.

Este framework ofrece un mecanismo automatizado a los desarrolladores para almacenar y acceder a los datos en la base de datos; es decir, simplifica el mapeo entre objetos en un sistema de software a las columnas y tablas de una base de datos relación.

Lo utilizamos en nuestro juego ya que ofrece un mecanismo automatizado bastante eficiente y fácil de usar para poder crear las tablas necesarias en la base de datos; en nuestro caso, creamos el modelo del Jugador, permitiéndonos acceder y modificar la tabla sin complicaciones durante la ejecución del juego.

**WPF**

Se trata de un framework de Interfaz de Usuario que crea aplicaciones de cliente de escritorio. WPF admite un conjunto de características para incluir en estas aplicaciones como controles, gráficos, diseños, enlace de datos, documentos y seguridad. WPF es un subconjunto de .NET Framework, algo similares a Windows Forms.

WPF es prácticamente la base de nuestro proyecto, pues es la interfaz principal para la comunicación entre el cliente y el servidor. Otro aspecto que nos ayudó muchísimo es su relación con el uso de XAML, pues la manera de trabajo entre interfaces y códigos nos facilitó funciones como la internacionalización.

**WCF**

Es utilizado para aplicaciones Orientadas a Servicios. Además, está basado en el concepto de comunicación basada en mensajes. El WCF consiste de tres componentes:

* Servicio WFC
* Servicio Host WCF
* Servicio Cliente WCF

Utilizarlo puede traer varias ventajas como la interoperabilidad, ofrecer mayor confiabilidad y seguridad; tiene un mecanismo de seguridad por default que es bastante robusto.

Dentro de nuestro sistema es utilizado para la comunicación entre jugadores, nos facilitó bastante manejar el intercambio de mensajes entre jugadores. Así como la comunicación entre el jugador y el sevidor con las acciones en las que fuera necesario, como el inicio de sesión o el desconectarse.

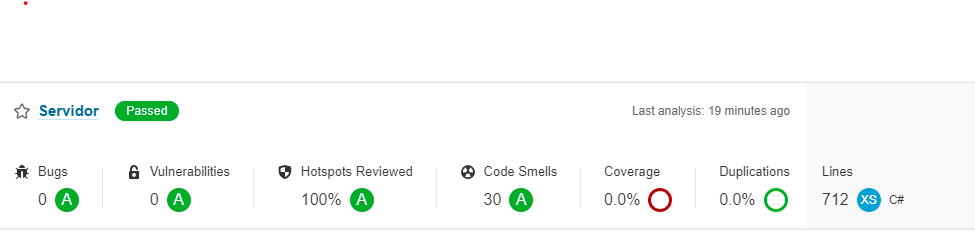
# Análisis estático de código

El análisis de código estático fue realizado utilizando el programa SonarQube. Este programa de código abierto nos permite realizar un análisis de todo el código del proyecto para detectar bugs, code smells, duplicaciones de código y posibles vulnerabilidades de seguridad.

Al ejecutarlo, obtuvimos los siguientes resultados:

**Servidor**

Antes

****

Después

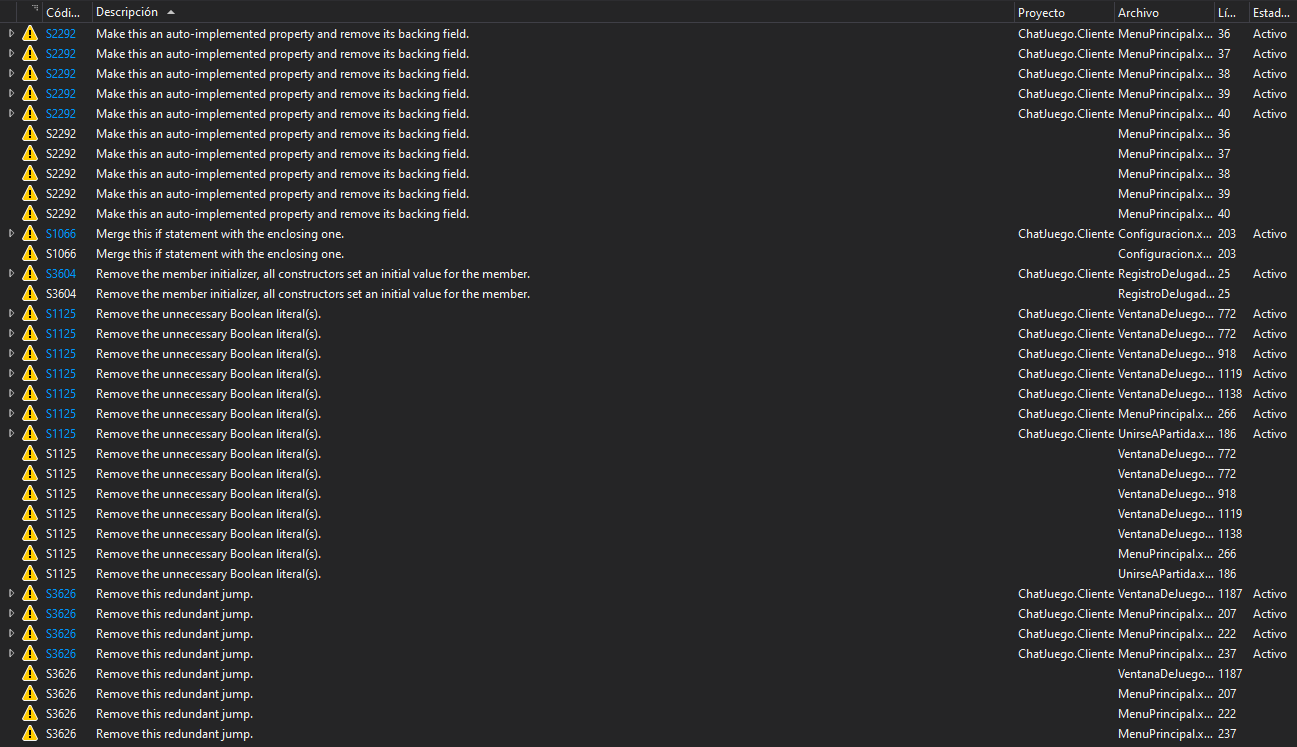
**Cliente**

Para el cliente, instalamos el paquete de SonarAnalyzer en la solución del Cliente para poder realizar el análisis de código estático.

Texto

Descripción generada automáticamente

Antes



Después

# Resultado de las pruebas

Se diseñaron 30 pruebas en total para verificar que el servidor funcione correctamente (se probaron tanto casos de éxito como casos en los que debería responder en caso de ser incorrecta la información proporcionada o si ocurre un problema de otro tipo). Las pruebas fueron divididas por la interfaz a la que corresponde, quedando de la siguiente forma:

* Pruebas del Servicio del Chat
* Pruebas del Servidor
* Pruebas del Servicio de Correo
* Pruebas del Servicio de la Tabla de Puntajes
* Pruebas del CallBack del Jugador

Los resultados fueron los siguientes:

Captura de pantalla de computadora

Descripción generada automáticamente

# Conclusiones

Consideramos que con este proyecto hemos generado bastante conocimiento sobre múltiples ámbitos en la construcción de software, ya sea la vinculación de un programa con correos electrónicos reales, funcionalidad un poco más avanzada de Visual Studio, la utilización de framework, el diseño con WPF, etc.

Durante el desarrollo de este proyecto, así como en el transcurso de la experiencia educativa, fuimos capaces de aprender muchas cosas nuevas y ponerlas en práctica. Consideramos que este proyecto es bastante relevante ya que, desde nuestro punto de vista, adquirimos conocimientos sobre cosas que van a ser clave para proyectos y materias próximos.