



ESCUELA SUPERIOR DE INFORMÁTICA
UNIVERSIDAD DE CASTILLA – LA MANCHA

Práctica 1

Tres en raya definitivo

Javier Marchán Loro
Ángel Peralta López
Rubén Pérez Pascual
Antonio Ruedas García

Asignatura: Ingeniería del Software II
Titulación: Ingeniería Informática
Fecha: 6 de febrero de 2014

1. Introducción

Este documento describe el análisis, desarrollo e implementación llevados a cabo para obtener un juego **Tres en raya definitivo** ¹.

Se trata de una aplicación cliente-servidor para escritorio que se comunica a través de RMI. El lenguaje utilizado para la implementación ha sido Java.

Este documento pretende que el lector obtenga una idea clara de la composición del sistema, para ello se ha dividido en X secciones. En primer lugar los requisitos funcionales ayudan a entender el objetivo buscado. La arquitectura del sistema junto con los casos de uso son las secciones más importantes, ya que ayudarán al lector a hacerse una imagen mental, a alto nivel, del sistema desarrollado.

EL resto de secciones profundizan en el conocimiento del sistema.

¹Este documento no explica las reglas del juego

2. Requisitos funcionales

2.1. Registrar Usuario

Caso de uso	Registrar Usuario
Precondición	Ninguna
Escenario general	<ol style="list-style-type: none"> 1. Al servidor le llega una petición de registro por parte del cliente. 2. El servidor comprueba si los datos del usuario existen en la base de datos. 3. En caso de no existir los almacena en la base de datos y crea al jugador. 4. El servidor confirma al usuario que el jugador ha sido creado.
Poscondición	Un nuevo usuario existe en el sistema.

2.2. Iniciar sesión

Caso de uso	Iniciar sesión
Precondición	Usuario registrado
Escenario general	<ol style="list-style-type: none"> 1. Al servidor le llega una petición de inicio de sesión por parte del usuario. 2. El servidor comprueba la validez de los datos del usuario. 3. El servidor confirma al gestor de usuarios que el usuario es válido. 4. El gestor de usuarios crea una sesión.
Poscondición	Se ejecuta el caso de uso "Ver lista de partidas"
Escenario alternativo	Puede ocurrir que los datos de acceso no sean correctos, en cuyo caso se aborta la operación y se le comunica al usuario que los datos introducidos son erróneos.

2.3. Cerrar sesión

Caso de uso	Cerrar sesión
Precondición	Sesión iniciada
Escenario general	<ol style="list-style-type: none">1. Al servidor le llega una petición de cierre de sesión por parte del usuario.2. El servidor elimina al jugador de la lista de jugadores activos.3. Si el jugador que cierra sesión está jugando una partida, se eliminará al jugador de la partida.
Poscondición	El usuario deja de estar activo en el servidor y borrado de la lista de jugadores.

2.4. Crear Partida

Caso de uso	Crear Partida
Precondición	Ambos usuarios deben estar logeados y uno debe haber retado a otro
Escenario general	<ol style="list-style-type: none"> 1. El usuario selecciona de la ventana principal crear una nueva partida. 2. La ventana principal muestra la ventana de crear partida. 3. El usuario selecciona las características de la partida. 4. La ventana crear partida habilita el botón <i>crear</i> cuando los datos son correctos. 5. El usuario pulsa el botón crear. 6. La ventana crear partida notifica al gestor de partidas que un usuario desea crear una partida. 7. El gestor de partidas pide al servidor mediante el proxy que cree una partida. 8. El proxy le indica al gestor de partidas que se ha creado una nueva partida. 9. El gestor modifica la ventana principal con el nuevo estado de las partidas. 10. La ventana principal oculta la ventana de crear partidas. 11. La ventana principal comunica al jugador que se ha creado una nueva partida.
Poscondición	En el sistema hay una nueva partida y se ejecuta el caso de uso "Unirse a partida".
Escenario alternativo	<ol style="list-style-type: none"> 1. El usuario selecciona de la ventana principal crear una nueva partida. 2. La ventana principal muestra la ventana de crear partida. 3. El usuario selecciona las características de la partida. 4. La ventana crear partida habilita el botón <i>crear</i> cuando los datos son correctos. 5. El usuario pulsa el botón crear. 6. La ventana crear partida notifica al gestor de partidas que un usuario desea crear una partida. 7. El gestor de partidas pide al servidor mediante el proxy que cree una partida. 8. El proxy le indica al gestor de partidas que no se ha podido crear la partida. 9. La ventana principal oculta la ventana de crear partidas. 10. La ventana principal comunica al jugador que no se ha podido crear la partida.

2.5. Eliminar partida

Caso de uso	Eliminar partida
Precondición	El jugador debe haber iniciado sesión y haberse unido a la partida.
Escenario general	<ol style="list-style-type: none">1. Al servidor le llega una petición eliminar partida por parte del usuario.2. El servidor elimina la partida.3. EL servidor notifica a ambos jugadores la eliminación de la partida.
Poscondición	

2.6. Registrar Victorias

Caso de uso	Registrar Victorias
Precondición	Los usuarios deben estar conectados a una partida y la partida debe haber finalizado.
Escenario general	<ol style="list-style-type: none">1. El servidor detecta que una partida ha finalizado y cuál es el jugador ganador.2. El servidor almacena en la base de datos al ganador de la partida.3. Notifica al los usuarios que la partida ha sido finalizada y qué jugador ha sido el vencedor.
Poscondición	Se elimina la partida.

2.7. Ver la lista de partidas

Caso de uso	Ver la lista de partidas
Precondición	Iniciar sesión
Escenario general	<ol style="list-style-type: none">1. La ventana principal pide al gestor de partidas una lista con las partidas libres disponibles y una lista con las partidas que el usuario esta jugando.2. El gestor de partidas pide al servidor ambas listas.3. El servidor envía ambas listas de partidas al gestor de partidas.4. La ventana principal muestra la lista de partidas disponibles y la lista de partidas comenzadas.
Poscondición	La lista de partidas esta disponible para el usuario.

2.8. Enviar Petición de Reto

Caso de uso	Enviar petición de Reto
Precondición	Usuarios logeados en el sistema y petición de reto realizada.
Escenario general	<ol style="list-style-type: none">1. Al servidor le llega una petición de reto por parte de un usuario.2. El servidor envía al otro jugador la petición de reto.
Poscondición	

2.9. Registrar un movimiento

Caso de uso	Registrar un movimiento
Precondición	Iniciar sesión, Unirse a una partida y Conectarse a una partida
Escenario general	<ol style="list-style-type: none">1. Al servidor le llega el movimiento realizado por un usuario.2. El servidor almacena el movimiento en la base de datos.3. El servidor notifica al otro usuario que ya puede realizar su movimiento.
Poscondición	Se realiza el caso de uso: "Enviar actualización de la partida".

2.10. Responder a acción enemiga

Caso de uso	Responder a acción enemiga
Precondición	Iniciar sesión, Unirse a una partida y Conectarse a una partida
Escenario general	<ol style="list-style-type: none">1. El servidor envía al gestor de partidas la información del ataque del oponente.2. El gestor de partidas pasa la información a la ventana de juego.3. La ventana de juego pregunta al jugador que acción realizar (aceptar ataque, dar soldados o dar gallifantes).4. El usuario elige la acción en la ventana de juego.5. La ventana de juego envía la información al gestor de partidas.6. El gestor de partidas envía la respuesta al servidor por medio del proxy.
Poscondición	

2.11. Comprar refuerzos

Caso de uso	Comprar refuerzos
Precondición	El usuario debe haberse unido a una partida y estar conectado a ella.
Escenario general	<ol style="list-style-type: none">1. El usuario pulsa sobre el botón comprar.2. Se muestra al usuario una ventana con diferentes opciones.3. El usuario introduce los datos necesarios y confirma la operación.4. El gestor del juego recibe los datos y envía la información al proxy.5. El proxy notifica al gestor de juego que la compra se ha realizado con éxito.6. El gestor de juego actualiza los datos del juego.
Postcondición	Se ejecuta el caso de uso “Enviar actualización de la partida”.
Escenario alternativo	<ol style="list-style-type: none">1. El usuario pulsa sobre el botón comprar.2. Se muestra al usuario una ventana con diferentes opciones.3. El usuario introduce los datos necesarios y confirma la operación.4. El gestor del juego recibe los datos, comprueba que se cumple la condición de que el jugador tenga dinero suficiente, el jugador no tiene dinero suficiente.5. El gestor de juego pide a la ventana que muestre un mensaje con información sobre el error producido al realizar la acción de compra.

2.12. Enviar petición de alianza

Caso de uso	Enviar petición de alianza
Precondición	El usuario debe haberse unido a una partida
Escenario general	<ol style="list-style-type: none">1. El usuario pulsa sobre el botón alianza.2. Se muestra al usuario una ventana con diferentes jugadores.3. El usuario selecciona uno de ellos y confirma la operación.4. El gestor del juego recibe los datos y envía la información al proxy.5. El proxy notifica al gestor de juego que la alianza se ha realizado con éxito.6. El gestor de juego actualiza los datos del juego.
Postcondición	Mostrar correctamente los datos de juego una vez actualizado y se ejecuta el caso de uso “Enviar actualización de la partida”.
Escenario alternativo	<ol style="list-style-type: none">1. El usuario pulsa sobre el botón alianza.2. Se muestra al usuario una ventana con diferentes jugadores.3. El usuario selecciona uno de ellos y confirma la operación.4. El gestor del juego recibe los datos, comprueba que se cumple la condición que permite la alianza, pero el jugador seleccionado no es válido para una alianza.5. El gestor de juego pide a la ventana que muestre un mensaje con información sobre el error producido al realizar la alianza.

2.13. Responder a petición de alianza

Caso de uso	Responder a petición de alianza
Precondición	El usuario debe haberse unido a una partida
Escenario general	<ol style="list-style-type: none">1. El proxy comunica al gestor de juego que existe una petición de alianza.2. El gestor de juego pide a la ventana que muestre al jugador el mensaje pertinente.3. El usuario selecciona la acción oportuna.4. El gestor del juego recibe los datos y envía la información al proxy.5. El proxy notifica al gestor de juego que la alianza se ha realizado con éxito.6. El gestor de juego actualiza los datos del juego.
Postcondición	Mostrar correctamente los datos de juego una vez actualizado Se ejecuta el caso de uso “Enviar actualización de la partida”. .

2.14. Romper alianza

Caso de uso	Romper alianza
Precondición	El usuario debe haberse unido a una partida y tener, al menos, una alianza
Escenario general	<ol style="list-style-type: none">1. El usuario pulsa sobre el botón alianza.2. Se muestra al usuario una ventana con diferentes alianzas.3. El usuario selecciona una de ellas y confirma la operación.4. El gestor del juego recibe los datos y envía la información al proxy.5. El proxy notifica al gestor de juego que la alianza se ha roto con éxito.6. El gestor de juego actualiza los datos del juego.
Postcondición	Mostrar correctamente los datos de juego una vez actualizado. Se ejecuta el caso de uso “Enviar actualización de la partida”. .

2.15. Enviar actualización de la partida

Caso de uso	Enviar actualización de la partida
Precondición	El usuario debe haberse unido a una partida
Escenario general	<ol style="list-style-type: none">1. El usuario realiza una operación que modifica la partida.2. El gestor de juego recopila todos los datos modificados y envía la información al proxy.3. El proxy notifica al gestor de juego que la actualización se ha realizado con éxito.
Postcondición	Los datos del cliente y del servidor están sincronizados.

2.16. Recibir actualización de la partida

Caso de uso	Recibir actualización de la partida
Precondición	El usuario debe haberse unido a una partida
Escenario general	<ol style="list-style-type: none">1. El proxy comunica al gestor de juego que la partida ha cambiado.2. El gestor de juego recibe todos los datos modificados y pide a la ventana que se actualice.
Postcondición	Los datos del cliente y del servidor están sincronizados.

3. Otros requisitos

3.1. Requisitos de la interfaz de usuario

3.1.1. Ventana de inicio

Descripción: Da la bienvenida al usuario.
Debe permitir al usuario ejecutar el caso de uso “Iniciar sesión” e introducir los siguientes datos:

Dato	Tipo	Descripción
NombreUsuario	Texto	Nombre del usuario
Contraseña	Texto	Contraseña en el sistema

3.1.2. Ventana de registro

Descripción: Permite registrarse al usuario.
Es necesario que permita al usuario introducir los siguientes datos:

3 Otros requisitos

Dato	Tipo	Descripción
Nombre	Texto	Nombre del usuario
eMail	Texto	Dirección de correo electrónico
Contraseña	Texto	Contraseña en el sistema
Conf. Contraseña	Texto	Confirmación de contraseña

3.1.3. Ventana principal

Descripción: Mostrará al usuario las partidas disponibles y las partidas en juego. Es necesario que permita al usuario ejecutar los siguientes casos de uso:

1. Crear una partida.
2. Unirse a una partida.
3. Conectarse a una partida.
4. Ver lista de partidas.

3.1.4. Ventana Crear partida

Descripción: Permite al usuario crear una nueva partida. Es necesario que permita al usuario introducir los siguientes datos:

Dato	Tipo	Descripción
Nombre	Texto	Nombre de la partida
Días de juego	Fecha	Días en los que se jugará la partida
Hora de Inicio de Juego	Hora	Hora en la que empieza o continua la partida los días indicados
Hora de Fin de Juego	Hora	Hora en la que termina la partida los días indicados

3.1.5. Ventana de juego

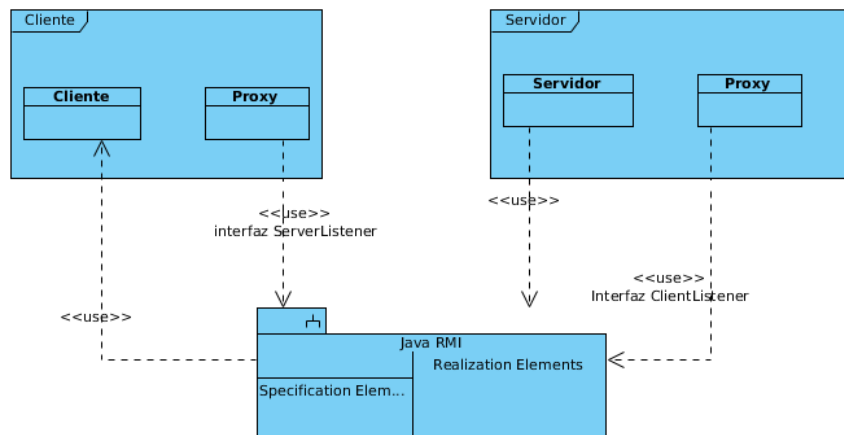
Descripción: Muestra al usuario todos los detalles de una partida y permite jugar en ella. Deberá mostrar el tablero de juego con la asignación de territorios.

Es necesario que permita al usuario ejecutar los siguientes casos de uso:

1. Desconectarse de una partida.
2. Realizar un movimiento.
3. Responder acción enemiga.
4. Comprar refuerzos.
5. Enviar petición de alianza.
6. Responder a petición de alianza.
7. Romper alianza.

4. Arquitectura del sistema

A continuación se muestra una visión de la arquitectura del sistema. Se trata de una arquitectura cliente-servidor a través de RMI. El cliente usa un proxy para comunicarse con el servidor y éste conoce un proxy de cada cliente con el que se comunica.



5. Casos de uso

Diagrama de casos de uso del cliente

5 Casos de uso

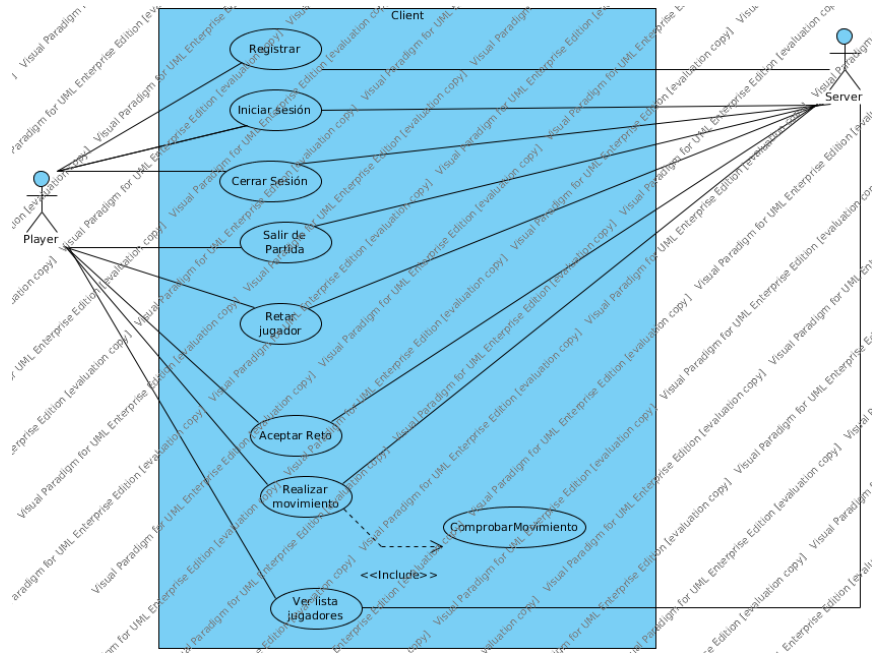
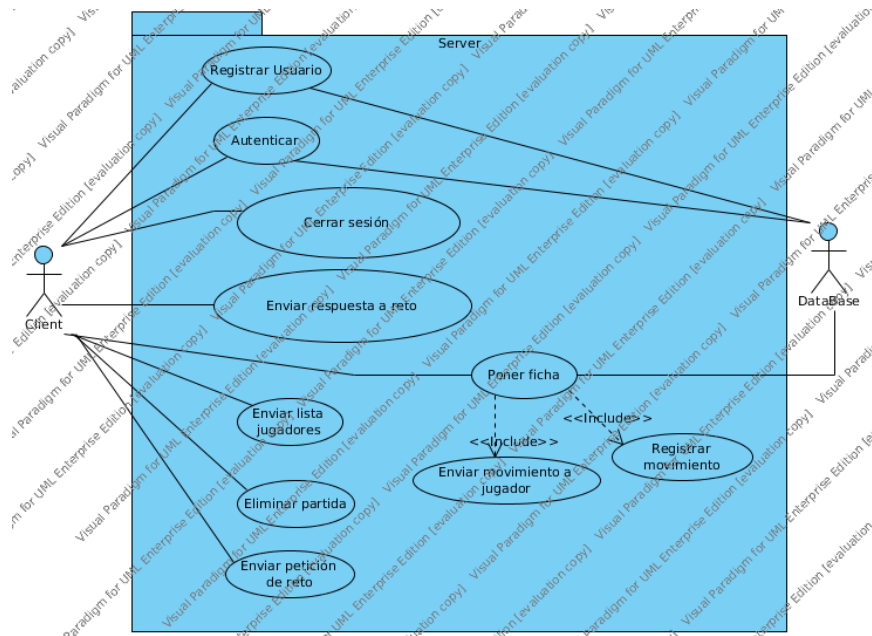


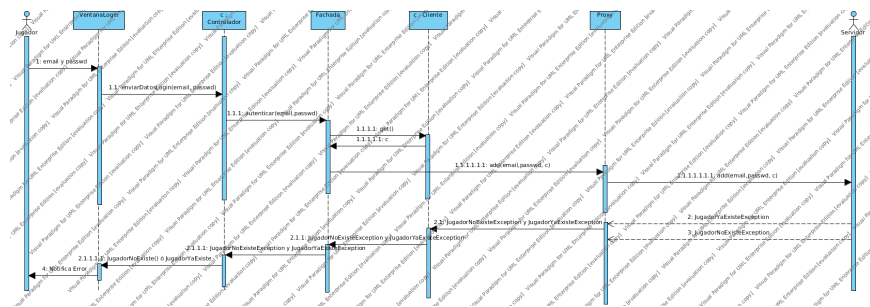
Diagrama de casos de uso del servidor



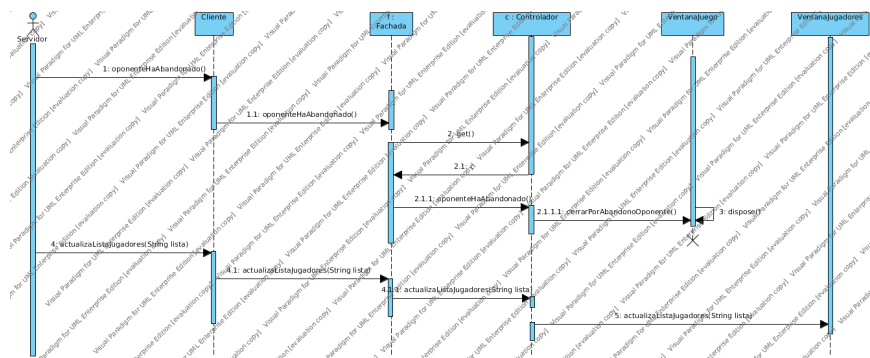
6. Diagramas de secuencia

Diagramas de secuencia del cliente:

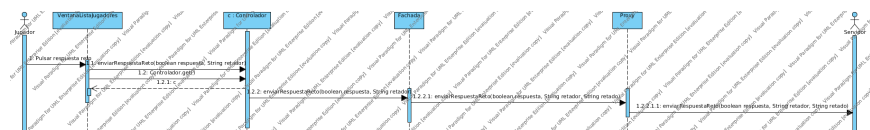
Autenticar



Cerrar sesión oponente

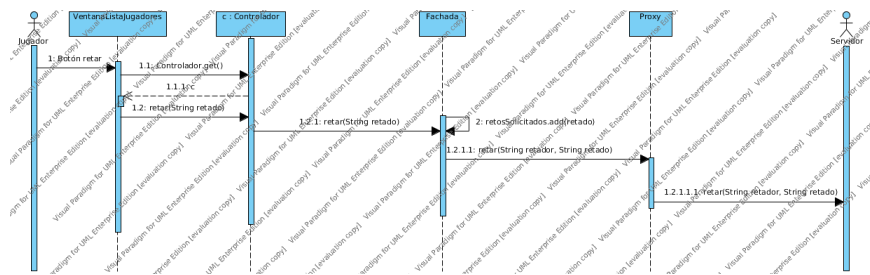


Enviar respuesta de reto

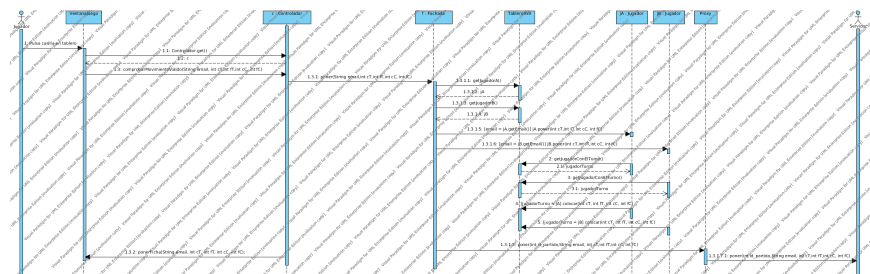


Enviar reto

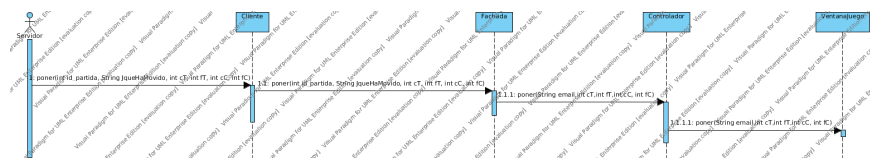
6 Diagramas de secuencia



Realizar movimiento

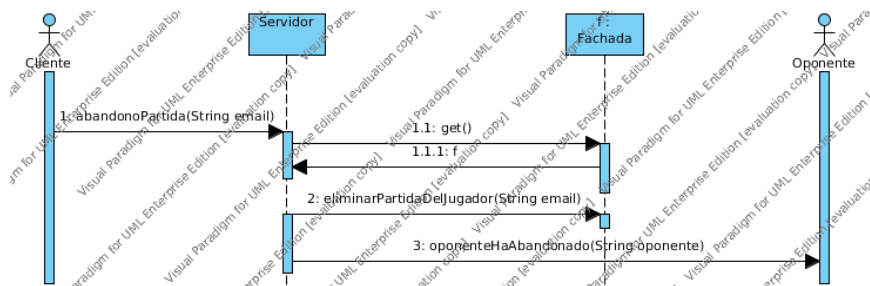


Recibir movimiento



Diagramas de secuencia del servidor:

Cerrar sesión



```

sequenceDiagram
    participant get as get
    participant getImpl as getImpl
    participant getImplImpl as getImplImpl

    get->>getImpl: get
    activate getImpl
    getImpl->>getImplImpl: get
    activate getImplImpl
    getImplImpl-->>getImpl: 
    deactivate getImplImpl
    getImpl-->>get: 
    deactivate getImpl
  
```

Sequence diagram for the `get` operation

The diagram illustrates the sequence of messages for the `get` operation. The `get` package sends a `get` message to the `getImpl` package. The `getImpl` package then sends a `get` message to the `getImplImpl` package. The `getImplImpl` package returns the result to the `getImpl` package, which then returns the result to the `get` package.

```

sequenceDiagram
    participant User
    participant System
    participant Database

    User->>System: get
    activate System
    System->>Database: get
    activate Database
    Database-->>System: result
    deactivate Database
    System-->>User: result
    deactivate System
  
```