

Universidade de Vigo

ESCOLA SUPERIOR DE ENXEÑARÍA INFORMÁTICA

Memoria do Traballo de Fin de Grao que presenta

D. Daniel Gómez Rodríguez

para a obtención do Título de Graduado en Enxeñaría Informática

Red de mensaxería instantánea e privada



Xullo, 2022

Traballo de Fin de Grao Nº: 20/21-80

Titor/a: Francisco José Ribadas Pena

Área de coñecemento: Ciencias da Computación e Intelixencia Artificial

Departamento: Informática

Red de mensaxería instantánea e privada

Dedicatoria

De min para a miña xente

Red de mensaxería instantánea e privada

Agradecementos

Agradecido aos meus pais pola oportunidade de estudar ata este nivel, aos meus profesores, a pesares de non haber tido tanto trato coma debería, aos meus compañeiros, tanto de carreira coma de celebracións, e moitas grazas a miña xente por ter tanta cachaza connmigo.

Indice

1	Introdución.....	1
2	Obxectivos.....	2
3	Resumo da solución.....	2
3.1	Servidor Web.....	2
3.2	Cliente.....	3
3.3	Base de Datos.....	3
3.4	Proceso de desenvolvemento.....	3
4	Planificación e Seguimento.....	5
4.1	Seguimento.....	6
5	Arquitectura.....	9
5.1	Arquitectura do servidor.....	10
5.2	Arquitectura da interface.....	10
6	Tecnoloxías, Ferramentas e Integracións de produtos de terceiros.....	10
6.1	Tecnoloxías Integradas no Proxecto.....	10
6.2	Ferramentas.....	11
7	Especificación e análise de requisitos.....	13
7.1	Actores.....	13
7.2	Requirimentos.....	13
7.2.1	Requirimentos Funcionais ^[18]	13
7.2.2	Requirimentos Non Funcionais ^[18]	14
7.3	Diagramas de casos de uso.....	15
7.4	Descrición de casos de uso.....	16
8	Deseño de software (estático e dinámico).....	22
8.2	Deseño estático.....	22
8.1.1	Modelo do dominio.....	22
8.1.2	Modelo do servidor.....	23
8.2	Deseño dinámico.....	27
9	Xestión de datos e información.....	31
9.1	Información do usuario.....	31
9.2	Información do sistema.....	32
10	Probas levadas a cabo.....	33
10.1	Probas automáticas.....	33
11	Manual de usuario.....	34
11.1	Manual do xestor da aplicación.....	34
	Requisitos Previos.....	34
	Instalación.....	34
	Configuración.....	35
	Cambiar a interface.....	36
11.2	Manual do usuario.....	36
12	Principais contribucións.....	41
13	Conclusións.....	42

13.1 Conclusións Técnicas.....	42
13.2 Conclusións Persoais.....	42
14 Vías de traballo futuro.....	43
15 Referencias.....	44
16 Anexos.....	46
Socket.IO.....	46
WebSocket.....	46
HTTP Long-Polling.....	46

Índice de Ilustracións

Ilustración 1: Representación de esforzo estimado en cada fase do proxecto....	4
Ilustración 2: Diagrama de Gantt do tempo estipulado.....	6
Ilustración 3: Diagrama de Gantt do tempo.....	8
Ilustración 4: Esquema xeral da arquitectura do sistema.....	9
Ilustración 5: Diagrama de casos de uso dos usuarios.....	15
Ilustración 6: Diagrama de casos de uso na xestión de grupos.....	15
Ilustración 7: Diagrama representando a arquitectura MVC no proxecto.....	25
Ilustración 8: Diagrama de clases representando os modelos da aplicación....	26
Ilustración 9: Diagrama de clases dos controladores da aplicación.....	26
Ilustración 10: Diagrama de clases dos datos persistentes.....	31
Ilustración 11: Diagrama de clases dos rexistros.....	32
Ilustración 12: Exemplo de erro ao conectar coa base de datos.....	35

Índice de tablas

Tabla 1: Tempos estimados de desenrolo do proxecto.....	6
Tabla 2: Tempos empregados reais.....	8
Táboa 3: Probas automatizadas levadas a cabo.....	33

1 Introducción

As novas tecnoloxías ofrecen ao ser humano a capacidade de transmitir información a distancia. Os avances necesarios para que isto fose posible son numerosos (medios de transmisión, criptoloxía, codificación, etc.), e o resultado que pódese observar na actualidade: **a posibilidade de comunicarse, virtualmente, con calquera persoa en calquera parte do mundo.**

Na actualidade, os métodos de comunicación a distancia máis comúns para o usuario medio son o correo electrónico, as chamadas telefónicas e os servizos de mensaxería instantánea. No sector de mensaxería, os máis coñecidos son Whatsapp^[1], Telegram (2.000 e 1.478 millóns de usuarios, respectivamente), aínda que tamén hai moitas máis dispoñibles con distintos enfoques, como por exemplo Signal^[2].

Debido á situación global provocada polo SARS-CoV-2, esta tecnoloxía mostra a utilidade de poder manter conectadas ás persoas aos seus ámbitos sociais e profesionais^[3], independentemente da súa situación. Sen embargo, ás veces os organismos que coordinan unha institución reutiliza unha ferramenta como ben pode ser Whatsapp para habilitar a capacidade de coordinar as operacións realizadas polos membros do grupo coma o facían en presencial, polo que o seu traballo pode verse entorpecido, ou no peor dos casos, comprometido.

Por outra parte, hai un incremento na preocupación dos usuarios de que os seus datos podan ser utilizados con fins daniños para o usuario, sexa estafas, chantaxe, ameazas ou calquera outro crime que poida ser cometido con esa información^[4].

Este proxecto nace coa necesidade de ofrecer aos usuarios una aplicación de mensaxería sinxela de configurar e activar nun entorno persoal para que teñan control da súas conversas. Tamén trata de cubrir os casos de uso básicos para apelar ao maior grupo posible de usuarios, e ofrecer a posibilidade de modificar o proxecto para especializar a utilidade do servizo.

2 Obxectivos

O obxectivo deste proxecto é crear unha **aplicación web** que permita aos usuarios comunicarse entre eles, sexa a través dun grupo de múltiples usuarios, así coma directamente entre eles. Só se poderá comunicar os usuarios que se haxan rexistrado como contactos entre eles, e só os usuarios dun grupo poden falar neste. As conversas deberán poder ser en tempo real.

Para poder utilizar a aplicación, **é necesario que o usuario se rexistre** no sistema. Deberá introducir o **nome** co que se identificará no servizo, un **correo electrónico** e o **contrasinal** que utilizará para acceder a súa conta.

Os **grupos constan** dun **creador**, o cal non pode ser alterado, un **grupo de moderadores**, os cales só poden ser escollidos polo creador do grupo, e **integrantes do grupo**, os cales só poden ser engadidos polo creador máis os moderadores.

Para levar a cabo esta proceso de rexistro, acceso e inclusión en contactos e grupos, requírese un servizo que permita conectar a todos os clientes. Para elo, utilizarase un servidor que ofrezca estas funcións, grazas a unha **interface de programación de aplicacións**^[5] (abreviado “**API**”).

A continuación enuméranse as funcionalidades que debe cumprir o sistema:

1. Ser capaz de engadir novos usuarios
2. Transmitir mensaxes entre usuarios
3. Permanecer activo o maior tempo posible
4. Informar ao usuario no caso de que non se poida conectar coa base de datos de que non poderá utilizar o servizo.

3 Resumo da solución

A solución proposta constará dunha API servida a través dun servidor que se comunica utilizando o **protocolo de transferencia de hipertexto**^[6] (abreviado **HTTP**), a cal responderá ás peticións realizadas polos usuarios a través do cliente. A continuación amplíanse os detalles sobre o servidor e o cliente:

3.1 Servidor Web

O servidor ofrecerá a posibilidade de recibir e responder a peticións a través de **HTTP**, grazas a una **API REST**^[7](do anglosaxón “**Representational State**

Transfer”). Esta elección basease na idea de que será máis sinxela a implantación, mantemento do código e compatibilidade con distintos tipos de clientes.

Para garantir a veracidade dos usuarios, utilizaranse tokens, os cales crearanse cando o usuario inicie sesión.

Na comunicación, o servidor abrirá una conexión a través de SocketIO^[8] para realizar unha comunicación a tempo real entre os interlocutores.

3.2 Cliente

Para conectarse co servidor, créase unha aplicación web, na cal o usuario poderá rexistrarse ou acceder á aplicación coas súas credenciais. Unha vez dentro, poderá xestionar a súa información e configuracións, os seus contactos e grupos, e comunicarse.

O cliente funciona independente do servidor, polo que só deberá realizar peticións HTTP ao servidor. Cando comeza unha sesión de chat, o cliente inicia unha conexión a través de SocketIO. Se o cliente permite a iniciación dun WebSocket^[9], utilízase este protocolo para a comunicación. Se o navegador non permite este tipo de comunicacións, o protocolo cambia a long-polling. Se ben este protocolo consume máis recursos do servidor, non incapacita ao usuario de poder utilizar a aplicación se o seu navegador ou dispositivo non permite unha conexión de WebSocket.

3.3 Base de Datos

Para o almacenaxe de información persistente, coma os datos dos usuarios, os grupos creados, as mensaxes que se desexen manter, etc., utilízase MongoDB^[10], unha base de datos baseada en documentos. Para distribuír a carga, utilízase o sistema Atlas, unha implementación de MongoDB na nube, aínda que o sistema pódese modificar para conectarse cunha instancia de MongoDB local.

3.4 Proceso de desenvolvemento

A metodoloxía aplicada é **RUP** (do anglosaxón “**Rational Unified Process**”) É un proceso de desenvolvemento de software desenvolto por Rational Software, e máis tarde adquirido por IBM. Ofrece unha visión para asignar, xestionar e escoller tarefas e responsabilidades nunha organización encargada de desenvolver software. O **obxectivo deste proceso é producir**, nunha marxe aceptable, **unha pauta, horario e presuposto** de software de alta calidade que cumpre as necesidades dos seus usuarios. Os principios desta son:

- **Adaptar o proceso** ás necesidades do cliente para o cal producimos o software,
- **Equilibrar as prioridades e estandarizar os requirimentos** dos participantes do proceso,
- O proxecto **entregase en etapas reiteradas**, onde vaise amosando valor iterativamente ao cliente,
- **Establecer unha comunicación fluída entre equipos**, aínda que sexan de departamentos distintos entrementes traballen no mesmo proxecto,

O proceso conta de dúas dimensións, representado na seguinte ilustración:

1. O eixo horizontal representa o tempo e amosa o ciclo de vida de cada aspecto do proceso a medida que avanza.
2. O eixo vertical representa as principais disciplinas dos procesos, nas cales agrúpanse actividades pola súa natureza.

As gráficas representadas amosan o tempo e recursos óptimos que deberían empregarse en cada fase da iteración.

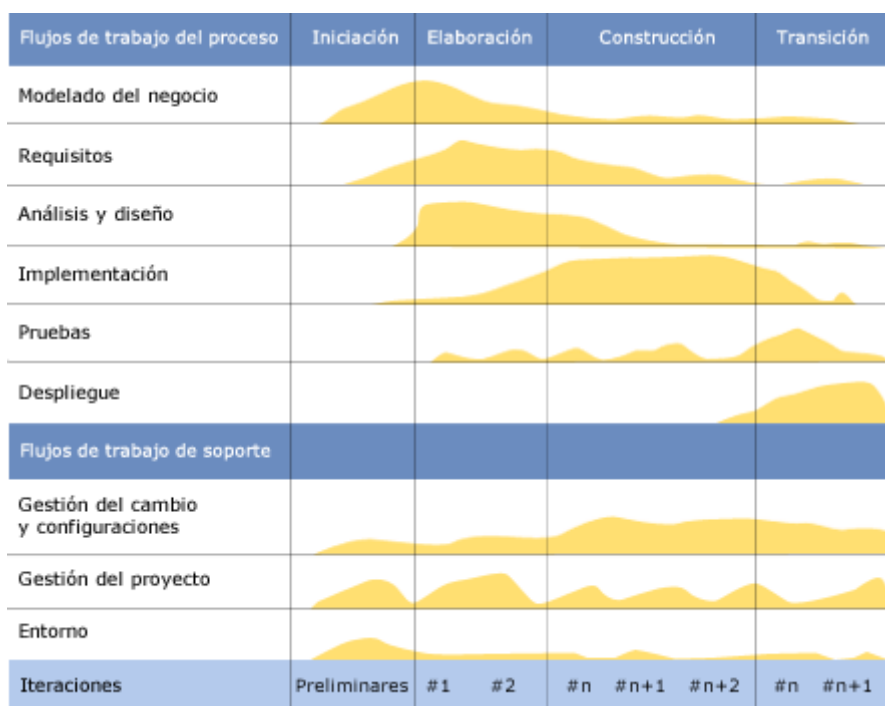


Ilustración 1: Representación de esfuerzo estimado en cada fase do proxecto

4 Planificación e Seguimento

O proxecto levarase a cabo seguindo una metodoloxía RUP^[11] (Proceso Racional Unificado), sendo máis específicos, seguindo a metodoloxía do Proceso Unificado.

No proxecto, aínda que sepáranse o servidor e o cliente, decídese avanzar tanto no servidor coma na interface ao mesmo tempo, para adecuar as necesidades do servidor respecto á idea inicial do proxecto.

A continuación lístanse as distintas iteracións planeadas:

1. Crear un sistema básico de servidor-cliente que poida enviar mensaxes.

Para o comezo do proxecto, necesitamos crear un entorno para poder cumprir os obxectivos do traballo. A idea será crear un servidor que reciba as conexións dos clientes, os cales, ao enviar unha mensaxe, mostraranse ao resto de clientes conectados ao servidor.

2. Crear unha interface de usuario básica

Despois de crear o sistema base, procederase co desenvolvemento dunha IU básica, coa que comprobaremos que, unha vez accedido ao servidor, visualizáranse ás mensaxes. Máis adiante, esta IU será ampliada para engadir as funcións a maiores de xestión e configuración.

3. Introducir un sistema de usuarios

Para asegurar que o sistema é privado, o sistema contará cun proceso de rexistro, no cal un cliente rexístrase cun usuario e contrasinal. Despois, tanto o servidor coma o cliente creará un par de claves privada e pública para incrementar a seguridade do servidor.

4. Crear salas de comunicación

Habilitarase a capacidade dos usuarios de crear salas privadas de chat. Tamén crearanse salas públicas no servidor.

En canto ao tempo dedicado ao proxecto, o obxectivo é dedicar entre 4 e 5 horas diarias ao proxecto, **sen contar os fins de semana**. O inicio do proxecto marcase no **luns, 23 de Agosto do 2021**. Dados os tempos previstos (na táboa nº 1 e na ilustración nº 2, na seguinte páxina), predícese un tempo total de traballo de **315 horas**.

A continuación amósase os tempos estimados en cada unha das iteracións da metodoloxía:

	Tarea	Duración	Comezo	Data de Fin (Estimada)
	TOTAL	70 días	23.08.21	05.11.21
1	Inicio	5 días	23.08.21	28.08.21
1.1	Documentación	1 día	23.08.21	24.08.21
1.2	Análise	2 días	24.08.21	26.08.21
1.3	Deseño	2 días	26.08.21	28.08.21
2	Elaboracion	20 días	30.08.21	19.09.21
2.1	Documentación	1 día	30.08.21	31.08.21
2.2	Análise	2 días	31.08.21	02.09.21
2.3	Deseño	5 días	02.09.21	07.09.21
2.4	Implementación	10 días	08.09.21	17.09.21
2.5	Probas	1 día	17.09.21	18.09.21
2.6	Documentación Resultados	1 día	18.09.21	19.09.21
3	Construción	40 días	20.09.21	30.10.21
3.1	Documentación	3 días	20.09.21	23.09.21
3.2	Análise	3 días	23.09.21	26.09.21
3.3	Deseño	5 días	26.09.21	01.10.21
3.4	Implementación	25 días	01.10.21	26.10.21
3.5	Probas	2 días	26.10.21	28.10.21
3.6	Documentación Resultados	2 días	28.10.21	30.10.21
4	Transición	4 días	01.11.21	05.11.21
4.1	Documentación	2 días	01.11.21	03.11.21
4.2	Probas	2 días	03.11.21	05.11.21

Tabla 1: Tempos estimados de desenrolo do proxecto

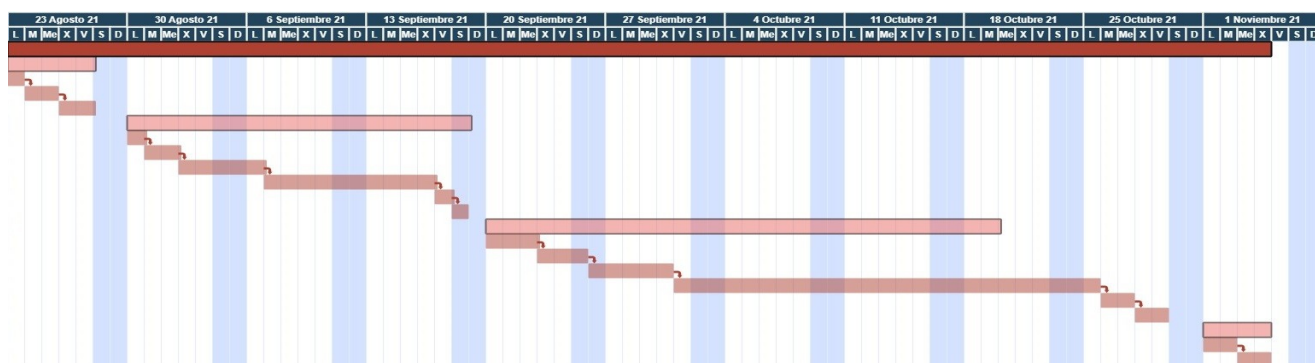


Ilustración 2: Diagrama de Gantt do tempo estipulado

4.1 Seguimento

A pesar de que os tempos previstos tiñan en conta a velocidade á que desenrolaba proxectos durante o curso, e os marxes eran máis amplos do previsto, o feito de que o proxecto abarcaba todo o entorno da aplicación (servidor, aplicación, bases...), os tempos reais non coincidiron co previsto.

O maior problema de seguir esta metodoloxía foi que, dado que está pensada para traballos en grupo, non soupen aplicalo correctamente na separación de accións. Tamén, o marxe de tempo de 4-5 horas de traballo por día foi moi optimista, polo que a cantidade de días empregados foi superior.

	Tarea	Duración	Comezo	Remate
	TOTAL	107 días	23.08.21	05.11.21
1	Inicio	12 días	23.08.21	03.09.21
1.1	Documentación	1 día	23.08.21	24.08.21
1.2	Análise	9 días	24.08.21	01.09.21
1.3	Deseño	2 días	01.09.21	03.09.21
2	Elaboración	37 días	06.09.21	21.10.21
2.1	Documentación	5 días	06.09.21	11.09.21
2.2	Análise	5 días	11.09.21	16.09.21
2.3	Deseño	14 días	16.09.21	08.10.21
2.4	Implementación	10 días	08.10.21	18.10.21
2.5	Probas	1 día	18.10.21	19.10.21
2.6	Documentación Resultados	2 días	19.10.21	21.10.21
3	Construción	54 días	21.10.21	15.12.21
3.1	Documentación	3 días	21.10.21	24.10.21
3.2	Análise	3 días	25.10.21	27.10.21
3.3	Deseño	5 días	28.10.21	01.11.21
3.4	Implementación	38 días	02.11.21	9.12.21
3.5	Probas	2 días	10.12.21	12.12.21
3.6	Documentación Resultados	2 días	13.12.21	15.12.21
4	Transición	4 días	15.12.21	18.12.21
4.1	Documentación	2 días	15.12.21	16.12.21
4.2	Probas	2 días	17.12.21	18.12.21

Tabla 2: Tempos empregados reais

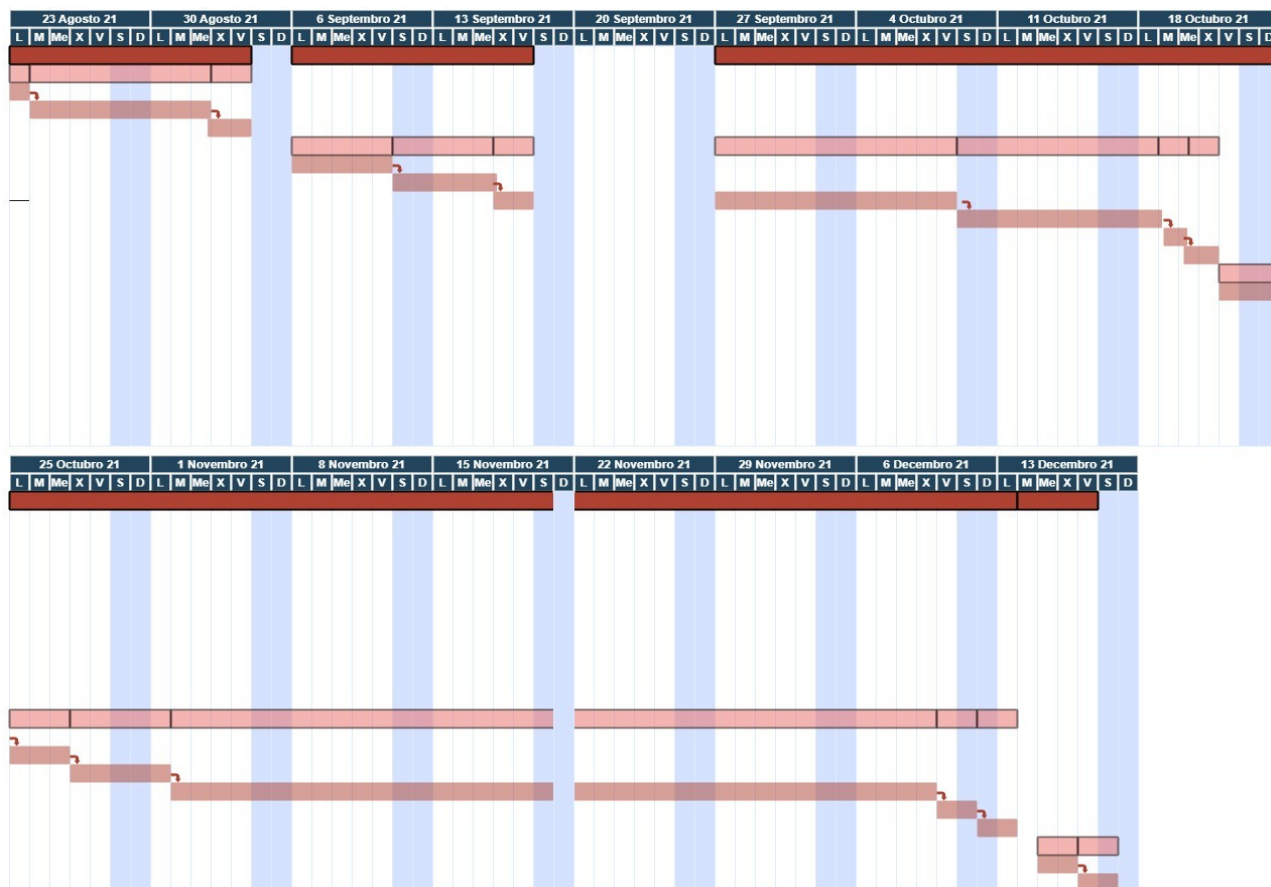


Ilustración 3: Diagrama de Gantt do tempo

5 Arquitectura

A aplicación constará dun servidor, encargado de acceder aos datos creados polos usuarios (mensaxes, grupos, contactos, etc.), e tamén encargado de prover ditos datos á interface. Será un servidor REST^[7], o cal pode recibir distintas peticións HTTP, ben para observar, actualizar, crear ou eliminar contido.

A interface creada para o servizo será ofrecida por outro servidor independente. O servidor pode estar almacenado no mesmo servidor que o provedor de información. A interface desenvolta ofrecese a través do navegador web, aínda que pode ser utilizado de forma local coma unha aplicación, se así o desexa o cliente.

Esta arquitectura permite separar as necesidades da interface coas capacidades da aplicación. Se ben neste proxecto a interface utilizará todas as

capacidades posibles desenvolvidas, isto capacita a distintas versións de interfaces de utilizar o sistema. Na seguinte ilustración amosase a arquitectura base do sistema.

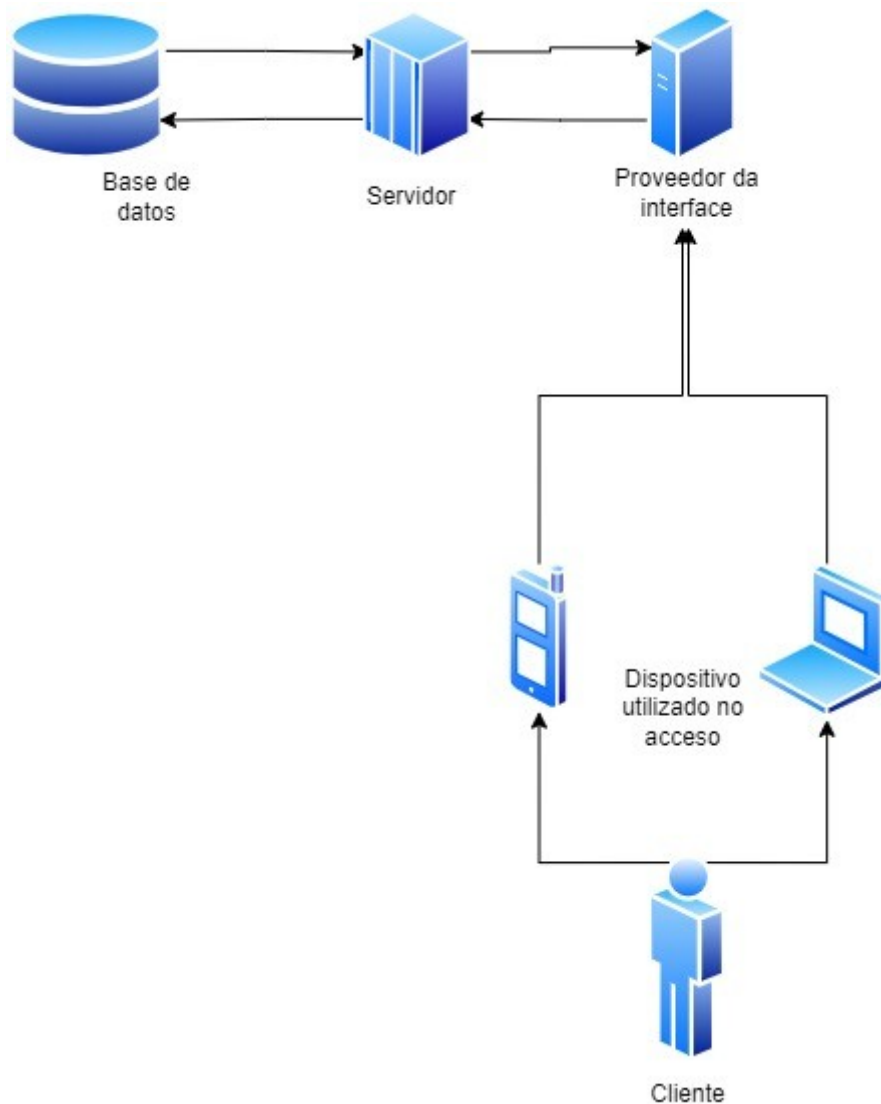


Ilustración 4: Esquema xeral da arquitectura do sistema

5.1 Arquitectura do servidor

Tal e como se describiu previamente, o servidor segue o patrón REST (Representational State Transfer en inglés). Fai referencia á comunicación establecida entre cliente e servidor, onde cada petición HTTP transmite a información necesaria para realizar unha operación.

5.2 Arquitectura da interface

Se ben neste documento refírese como interface, sería o equivalente a aplicación, web app, aplicación de escritorio, ou outra interface que desexa interactuar co servidor (por exemplo, chatbots), pero para este proxecto céntrase en realizar unha aplicación web sinxela para poder aproveitar a interoperabilidade que ofrece, sen comprometerse nun proxecto moi ambicioso.

6 Tecnoloxías, Ferramentas e Integracións de produtos de terceiros

6.1 Tecnoloxías Integradas no Proxecto

- **Node.js**^[12]

Node é un entorno de execución multiplataforma de JavaScript, deseñado para crear aplicacións de red escalables, coma por exemplo servidores web. O código execútase no servidor, a diferenza da maioría do código JavaScript, o cal está deseñado a executarse no navegador.

Node consiste no motor V8 de JavaScript, o que permite á máquina executar código JavaScript coma se fose unha aplicación propia. Esta aplicación executase nun proceso unicamente, sen crear novos fíos por cada petición de interrupción. Isto permite que Node sexa capaz de controlar conexións concorrentes cun único servidor sen introducir a dificultade engadida a xestionar concurrencia de fíos, o cal é un foco de bugs.

Tamén, grazas á súa licenza de open-source^[17], hai unha cantidade de paquetes e librerías de frameworks e ferramentas para engadir capacidades ao servidor creados pola comunidade, os cales son útiles para reducir o tempo empregado en crear código.

- **Express**^[13]

Express é o framework “de facto” para a infraestrutura dunha aplicación web en Node.js.

- **Bcryptjs**^[14]

Módulo para almacenar contrasinais hasheadas, por motivos de seguridade. Esta librería hashea os contrasinais do usuario dende o front e no back, para evitar que poida ser adquirida nun ataque de MITM.

- **Jsonwebtoken**^[15]

Módulo que implementa JSON Web Token en Node.

- **Mongoose**^[10]

Librería para Node, a cal permite escribir consultas para unha base de datos de MongoDB. Tamén conta con ferramentas adicionais para enriquecer a funcionalidade da base de datos. Algunhas destas características son:

- Validar os datos que se tratan de introducir na base
- construción de queries e middlewares para facilitar a reusabilidade de código
- Conversión de tipos de datos

- **Socket.IO**^[8]

É unha librería que activa unha comunicación en tempo real, bidireccional e baseada en eventos entre o navegador e o servidor.

- **Vue**^[16]

Framework de JavaScript para a construción de interfaces de usuario e aplicacións de páxina única, aínda que tamén é capaz de realizar interfaces de múltiples páxinas. Está deseñado para ser adoptable ás necesidades do usuario e en incrementos: está pensado para contar cunha librería central mínima, e coa capacidade de incrementar as súas funcións e capacidades con librerías externas.

6.2 Ferramentas

- **Visual Studio Code**

Editor de texto enriquecido. Debido á cantidade de extensións para facilitar o desenvolvemento de software, e tamén á popularidade do linguaxe JavaScript, é un bo entorno no que traballar no proxecto.

- **Postman**

Plataforma para a creación e utilización de APIs. Para este proxecto a aplicación utilizarase para probas.

- **MongoDB Atlas**

SaaS (Software as a Service), ofrece unha base de datos na nube para diminuír as competencias a manexar no proxecto. Coa versión de balde pódese crear unha base de datos cun límite de tráfico e capacidade.

- **GitHub**

Ferramenta para a xestión de versións do software.

- **Draw.io**

Ferramenta utilizada na creación dos modelos e esquemas do documento.

- **LibreOffice**

Editor de texto enriquecido, desenvolto como no proxecto Open Source^[17]. Utilizado para a elaboración da documentación do avance do proxecto.

7 Especificación e análise de requisitos

De acordo co Proceso Unificado, o análise de requisitos é o paso inicial do proxecto. Isto non quere dicir que todos os requirimentos teñen que ser amosados no análise inicial, xa que en futuras iteracións poden amosar novos casos. Neste documento amósanse todos os casos recollidos ata o final do proxecto, así coma os actores e diagramas realizados para representar toda esta información:

7.1 Actores

Consideraranse os seguintes actores:

- **Usuario:** O tipo xenérico de cliente. Pode agregar a outros usuarios como contactos para conversar con eles, así coma crear grupos para conversar con varios usuarios. Cando un usuario crea un grupo, convertese no actor **creador** dese grupo.
- **Creador:** Usuario propietario dun grupo de conversa. Pode xestionar os moderadores do grupo, e ten control total do grupo.
- **Moderador:** Usuario escollido polo **creador** do grupo para moderar o grupo de conversa. Pode realizar as accións de xestión no grupo, pero non poden escoller novos moderadores.

7.2 Requirimentos

Divídense os requirimentos en dous grupos distintos:

7.2.1 Requirimentos Funcionais^[18]

Son as declaracións dos servizos que prestará o sistema, así coma os que non debe realizar baixo ningún concepto. Descríbese os seguintes:

- **Xestión do usuario**
O usuario debe ter a opción de observar os seus datos de perfil, así coma modificalos.
- **Pescuda de usuarios**
O usuario debe poder buscar outros usuarios.
- **Enviar petición de contacto**
O usuario debe ter a opción de enviar unha petición para formar parte dos contactos de outro usuario.
- **Crear grupo de conversa**
O usuario debe poder crear un novo grupo de conversa.
- **Xestionar grupos de conversa**

Os donos dos grupos deben ter a capacidade de xestionar os moderadores e usuarios do grupo.

- **Agregar usuarios ao grupo**
Os donos dos grupos deben poder engadir novos integrantes no grupo.
- **Iniciar conversa**
Os usuarios deben poder, ou ben iniciar unha conversa, ou unirse a unha conversa en curso, sempre e cando teñan as capacidades de unirse a esta.

7.2.2 Requisitos Non Funcionais^[18]

Son aqueles requirimentos que definen as restricións do sistema como a capacidade dos dispositivos de entrada e saída, así coma a representación dos datos que se utilizan na interface do sistema. Algúns dos requirimentos non funcionais son:

- **Dispoñibilidade**
Tempo durante o cal o sistema estará en servizo ofrecendo a súa funcionalidade plena. Neste tempo englobase tamén o tempo empregado no mantemento do sistema, o cal forma parte do proceso de desenvolvemento.
- **Instalabilidade**
Refírese ao tempo e capacitación requirido para realizar as operacións pertinentes para poder comezar a utilizar o sistema.
- **Interoperabilidade**
Refírese á capacidade do sistema de intercambiar servizos con outros sistemas e a facilidade coa que se integra noutros dispositivos externos.
- **Robustez**
É a capacidade do sistema de continuar adecuadamente ante situacións anormais como fallas no hardware, datos non válidos, ataques ou condicións inesperadas.
- **Seguridade**
É un concepto moi amplo, pero pódese resumir en requisitos para mitigar o risco de ataques e prever ameazas.

7.3 Diagramas de casos de uso

Na seguinte figura amósase os casos de uso que pódense extraer co análise realizado:

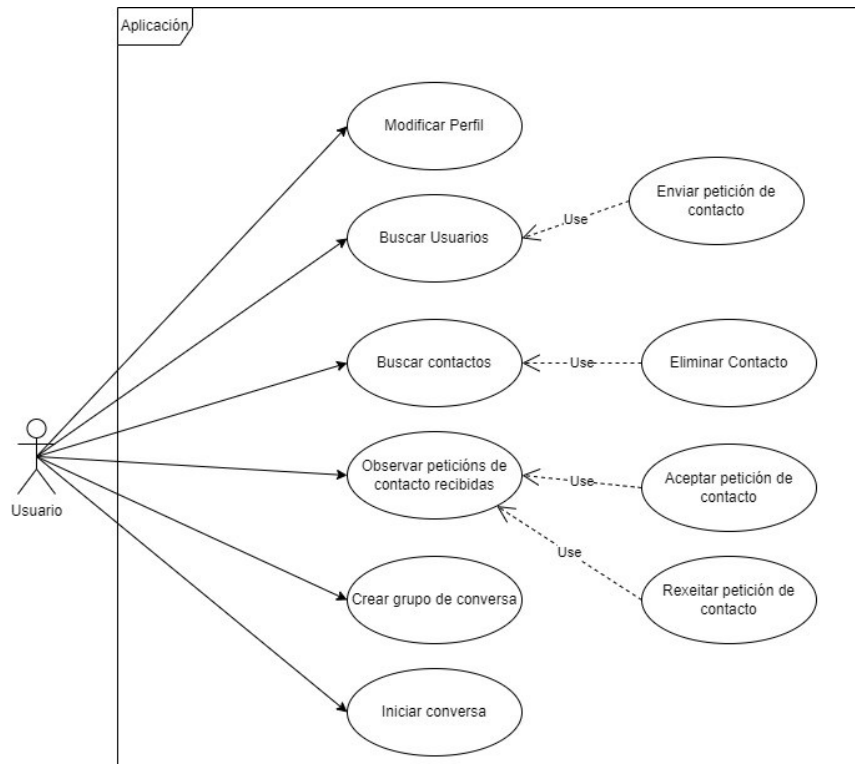


Ilustración 5: Diagrama de casos de uso dos usuarios

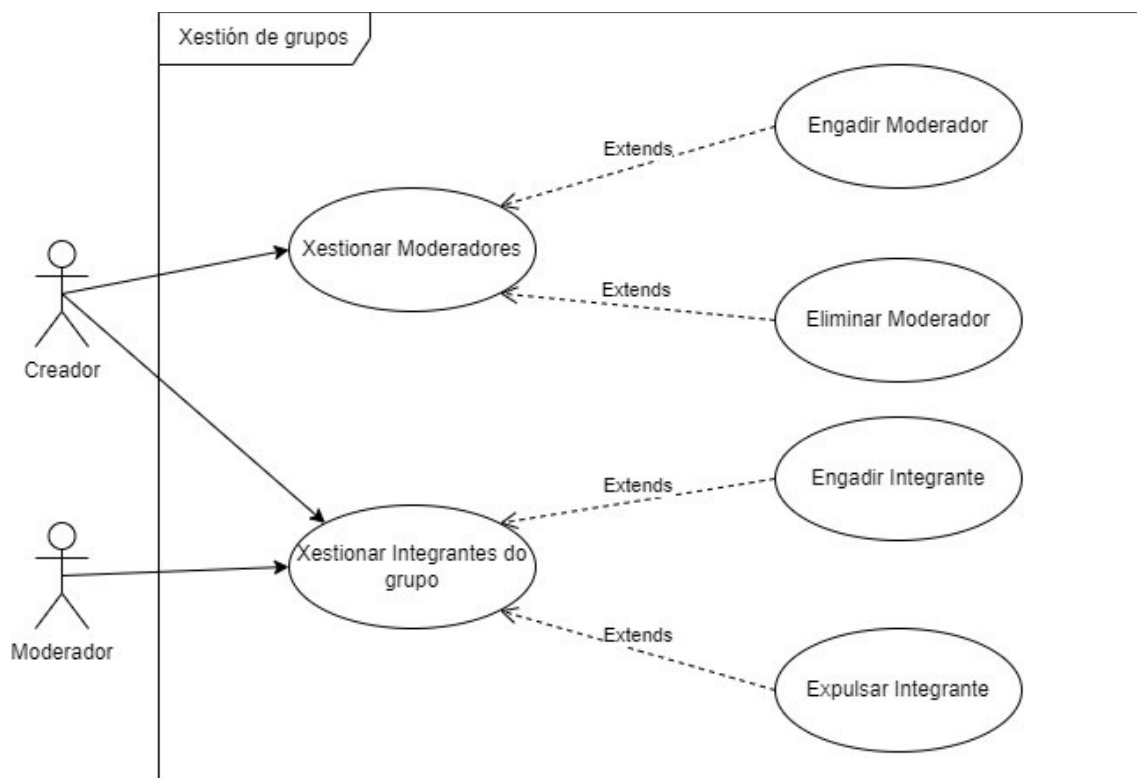


Ilustración 6: Diagrama de casos de uso na xestión de grupos

7.4 Descrición de casos de uso

Neste punto descríbense máis a fondo os casos de uso descritos previamente:

- **Casos de uso dos usuarios:**

Caso: Modificar Perfil		
Actor		Usuario
Descrición		Observar e modificar os datos do perfil do usuario en cuestión
Condicións Previas		O usuario ten que iniciar sesión na aplicación previamente
Condicións Posteriores		Os datos do usuario deben quedar rexistrados no sistema
Fluxo de eventos		
	Cliente	APLICACIÓN
1	Acceder á pantalla de “Perfil”	
2		Carga os datos do usuario e mostra un icono para acativar a modificación
3	Modifica os datos pertinentes	
4		Valida os datos
5		Informa ao usuario dos cambios

Caso: Buscar Usuarios		
Actor	Usuario	
Descrición	Buscar usuarios rexistrados na aplicación co nome de usuario	
Condicións Previas	O usuario ten que iniciar sesión na aplicación previamente e introducir o nome de usuario a buscar	
Condicións Posteriores	O sistema debe amosar os resultados da busca	
Fluxo de eventos		
	Cliente	APLICACIÓN
1	Accede á pantalla de “Comunicacións”	
2		Amosa a pantalla de seleccionar opción
3	selecciona a opción de “Usuarios”	
4		Amosa un campo para introducir o nome da búsqueda
5	introduce o nome do usuario a buscar, ou parte deste	Busca os datos na BD, e os mostra por pantalla.

Caso: Enviar Petición de Contacto		
Actor		Usuario
Descripción		Enviar unha petición para formar parte dos contactos do usuario.
Condicións Previas		O usuario ten que iniciar sesión na aplicación previamente e haber seleccionado un usuario na busca de usuarios
Condicións Posteriores		Debe quedar rexistrada unha nova petición de contacto no sistema
Fluxo de eventos		
	Cliente	APLICACIÓN
0	[Tra-lo final con éxito de “Buscar Usuarios”]	
1	Selecciona a opción de “Engadir Usuario”	
2		Comproba que non existe unha petición xa
3		Crea unha nova petición
4		Retorna ó usuario á pantalla de Usuarios

Nome: Buscar Contactos		
Actor	Usuario	
Descripción	Móstranse os contactos do usuario	
Condicións Previas	O usuario ten que iniciar sesión na aplicación previamente	
Condicións Posteriores	O sistema devolve os datos dos usuarios que forman parte dos contactos do usuario	
Fluxo de eventos		
	Cliente	APLICACIÓN

1	Accede á pantalla de “Comunicacións”	
2		Amosa a pantalla de seleccionar opción
3	O usuario selecciona “Contactos”	
4		Mostrase os usuarios que son contactos do usuario
Caso: Eliminar contacto		
Actor		Usuario
Descrición		Amosa os contactos do usuario
Condicións Previas		O usuario ten que iniciar sesión na aplicación previamente e seleccionar o contacto a eliminar
Condicións Posteriores		O sistema elimina o contacto da lista do usuario, así coma ao usuario da lista de contactos do eliminado.
Fluxo de eventos		
	Cliente	APLICACIÓN
0	[Tra-lo final con éxito de “Buscar Contactos”]	
1	Selecciona un contacto	
2		Mostrase información adicional do contacto e a opción de eliminar contacto
3	O usuario preme no botón de eliminar	
4		Bórrase o contacto da lista do usuario

Caso: Observar Peticións de Contacto Recibidas	
Actor	Usuario
Descrición	Móstranse as peticións de contacto recibidas cara o usuario
Condicións Previas	O usuario ten que iniciar sesión na aplicación previamente
Condicións Posteriores	O sistema devolve todas as peticións

		por aceptar ou rexeitar do usuario
Fluxo de eventos		
	Cliente	APLICACIÓN
1	Accede á pantalla de “Comunicacións”	
2		Amosa a pantalla de seleccionar opción
3	Selecciona “Peticións”	
4		Móstranse as peticións pendentes recibidas
Caso: Aceptar Petición de Contacto		
Actor		Usuario
Descrición		Acepta ao outro usuario como contacto
Condicións Previas		O usuario ten que iniciar sesión na aplicación previamente e seleccionar a petición en cuestión
Condicións Posteriores		Agregasen ambos usuarios como contactos e eliminase a petición de contacto
Fluxo de eventos		
	Cliente	APLICACIÓN
0	[Tra-lo final con éxito de “Observar Peticións de Contacto Recibidas”]	
1	Selecciona unha das peticións pendentes	
2		Mostra un formulario para aceptar ou declinar a petición
3	Preme na opción de “Aceptar”	
4		Agrega o contacto á lista de contactos do usuario, e na do contacto

Caso: Rexeitar Petición de Contacto	
Actor	Usuario
Descrición	Denega a petición de contacto de outro usuario
Condicións Previas	O usuario ten que iniciar sesión na

		aplicación previamente e debe acceder á petición en cuestión
Condicións Posteriores		A petición eliminase do sistema
Fluxo de eventos		
	Cliente	APLICACIÓN
0	[Tra-lo final con éxito de “Observar Peticións de Contacto Recibidas”]	
1	Selecciona unha das peticións pendentes	
2		Mostra un formulario para aceptar ou declinar a petición
3	Preme na opción de “Rexeitar”	
4		Elimina a petición de contacto
Caso: Crear Grupo de Conversa		
Actor		Usuario
Descrición		Crease un novo grupo de conversa
Condicións Previas		O usuario ten que iniciar sesión na aplicación previamente e debe introducir os datos do grupo
Condicións Posteriores		Crease o novo grupo no sistema
Fluxo de eventos		
	Cliente	APLICACIÓN
1	Accede á pantalla de “Comunicacións”	
2		Amosa a pantalla de seleccionar opción
3	Selecciona “Grupos”	
4		Mostra a pantalla de grupos aos que pertence, e a opción de crear un novo
5	Selecciona “Crear Grupo”	
6		Amosa un formulario para introducir os datos do grupo
7	Introduce os datos e envía	
8		Rexistra o grupo na BD

Caso: Iniciar Conversación	
Actor	Usuario
Descrición	O usuario accede a unha conversa
Condicións Previas	O usuario ten que iniciar sesión na aplicación previamente, seleccionar

		a conversa na que vai falar e ter os permisos para acceder a esta
Condicións Posteriores		O usuario accede á conversa
Fluxo de eventos		
	Cliente	APLICACIÓN
1	Accede á pantalla de “Comunicacións”	
2		Amosa a pantalla de seleccionar opción
3	Selecciona “Conversas”	
4		Mostra a pantalla de iniciar conversa en grupos e privado
5	Selecciona o grupo ou contacto co que conversar	
6		Accede á pantalla do chat seleccionado

- **Casos de uso da xestión de grupos**

Caso: Xestionar Moderadores		
Actor		Creador do grupo
Descrición		O creador controla quen é ou deixa de ser un moderador no grupo
Condicións Previas		O usuario ten que iniciar sesión na aplicación previamente e seleccionar o grupo do cal é creador
Condicións Posteriores		O sistema amosa os moderadores do grupo, máis a opción de engadir novos.
Fluxo de eventos		
	Cliente	APLICACIÓN
0	[Tra-lo final con éxito de “Seleccionar Grupo”]	
1	Se é creador do grupo, móstranse novos controles para engadir ou retirar moderadores	
2	[Tras seleccionar engadir ou retirar]	
3		Actualiza os datos do grupo na BD

Caso: Xestionar Integrantes	
Actor	Creador do grupo / Moderadores

Descrición		O creador controla quen é ou deixa de ser un moderador no grupo
Condicións Previas		O usuario ten que iniciar sesión na aplicación previamente e seleccionar o grupo do cal é creador
Condicións Posteriores		O sistema amosa os moderadores do grupo, máis a opción de engadir novos.
Fluxo de eventos		
	Cliente	APLICACIÓN
1	[Tra-lo final con éxito de "Seleccionar Grupo"]	
2	Se é moderador ou creador do grupo, móstranse novos controles para engadir ou retirar moderadores	
3	[Tras seleccionar engadir ou retirar]	
4		Actualiza os datos do grupo na BD

8 Deseño de software (estático e dinámico)

A continuación documéntase o deseño estático e dinámico do proxecto, describindo a estrutura de clases e obxectos no sistema e as interaccións entre eles mediante diagramas, así coma unha breve explicación, a cal será ampliada nos Anexos do documento.

8.1 Deseño estático

8.1.1 Modelo do dominio

Da definición do problema, deséñase un sistema que cumpra as seguintes normas:

- Debese permitir a entrada ao sistema a calquera usuario. Para identificarse conta cun nome de usuario e contrasinal.
- O usuario ten unha configuración propia. Nela, pode escoller como serán tratadas as mensaxes que envíe (se perduran ou non), se é un usuario público, etc.
- Os usuarios poden crear grupos, nos que teñen control total do grupo. Poden asignar moderadores a outros integrantes do grupo, nos que delega certas responsabilidades e capacidades no grupo.
- Os usuarios poden transmitir mensaxes, ben sexa directamente a outro usuario, ou ben a un grupo no que forme parte.

Na ilustración nº7 podemos ver o diagrama de clases que obtemos coa análise realizada do problema:

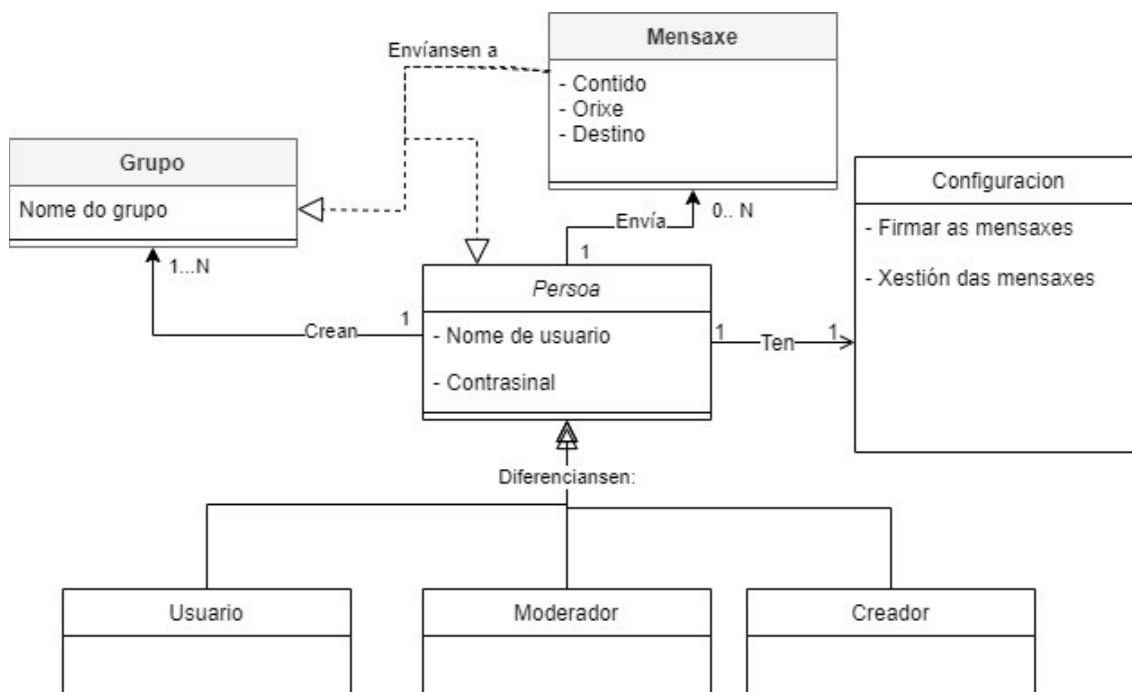


Ilustración 7: Diagrama de clases do sistema.

8.1.2 Modelo do sistema

O servidor seguirá unha arquitectura MVC^[20] (do inglés Model, View, Controller). Este modelo xurde coa necesidade de ter unha arquitectura que permita crear software máis robusto, cun ciclo de vida máis apropiado, é dicir, permitir a facilidade de mantemento, reutilización de código e separación de conceptos.

Fundaméntase na separación do código en tres capas :

- **Modelo**. Nela, trabállanse cós datos, polo que nesta capa recompílanse os esquemas nos que almacenaremos a información, e medios necesarios para almacenar devandita información.
- **Vista**. Conteñen o código que produce a visualización das interfaces de usuario.
- **Controladores**. Conteñen o código necesario para responder ás accións solicitadas á aplicación. Tamén é a capa que actúa como enlace entre o Modelo e as Vistas: os datos que o Modelo debe transmitir á vista para mostrar correctamente a interface, así coma as peticións de

almacenar, obter, modificar ou eliminar datos enviadas polos usuarios nas Vistas, son controlados nesta capa para permitir unha transmisión entre eles.

Nas ilustracións 8, 9 e 10 amósanse os diagramas representando os distintos compoñentes no sistema.

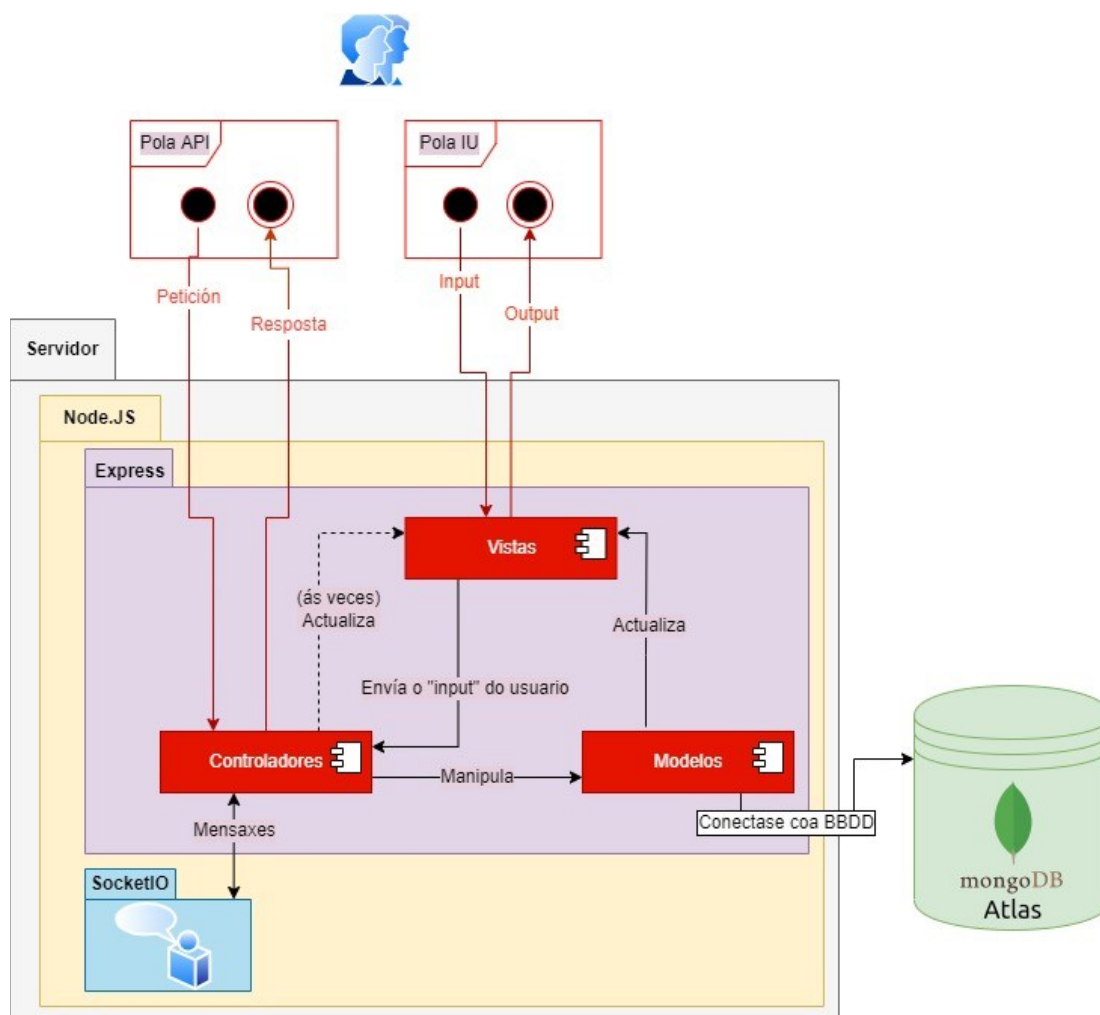


Ilustración 8: Diagrama representando a arquitectura MVC no proxecto

Red de mensajería instantánea e privada

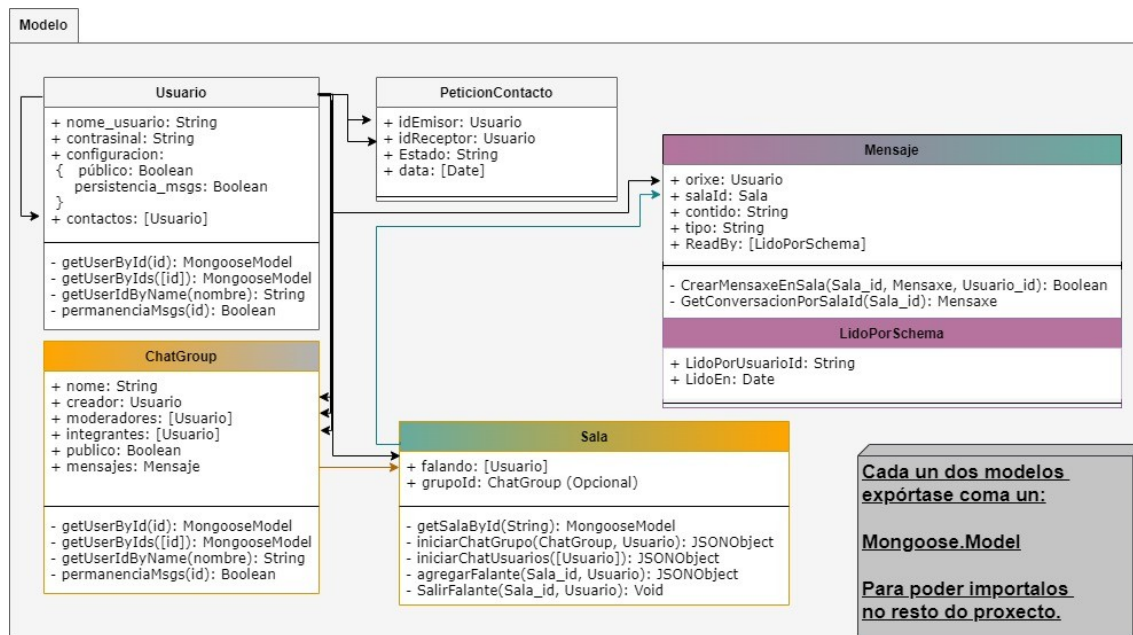


Ilustración 9: Diagrama de clases representando os modelos da aplicación

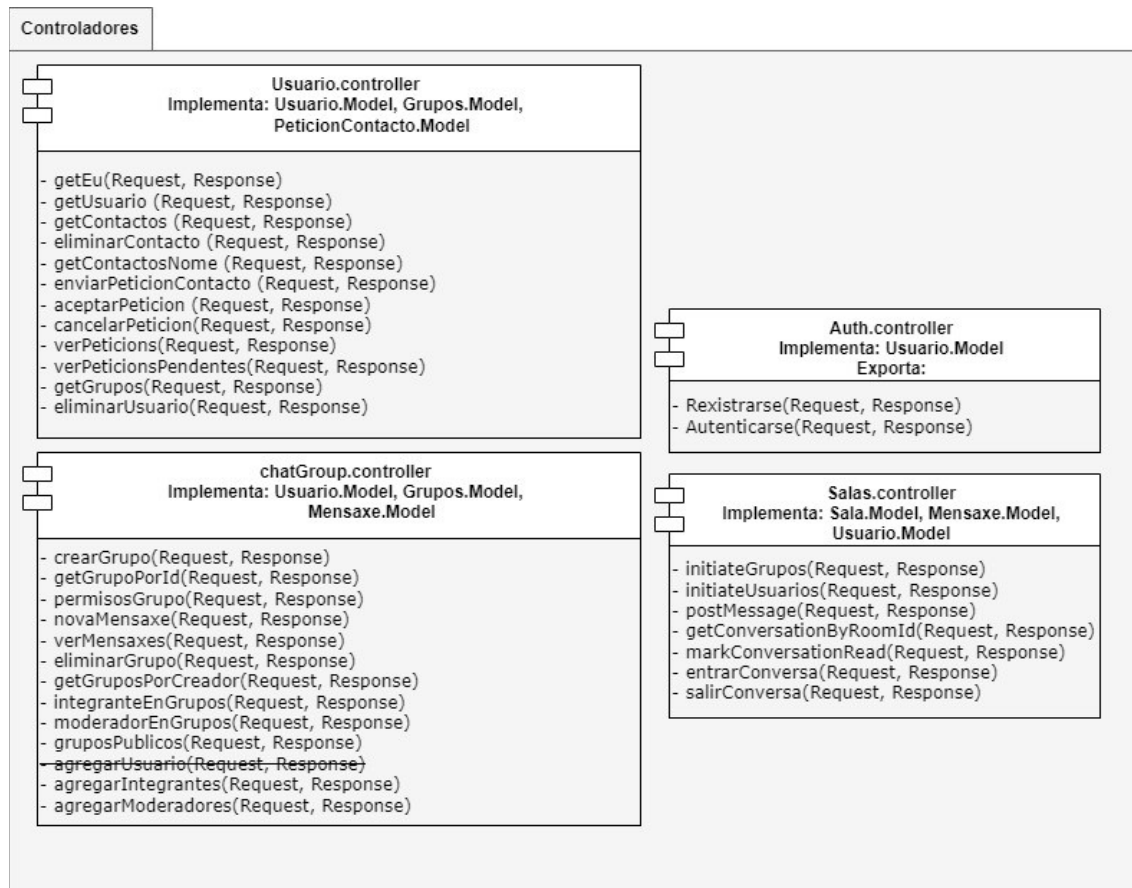


Ilustración 10: Diagrama de clases dos controladores da aplicación

8.2 Deseño dinámico

Nesta sección procédese a representar graficamente as secuencias de interaccións no sistema para as actividades dispoñibles na aplicación.

Dado que hai sistemas que teñen unha secuencia moi similar, modelaranse os aspectos máis relevantes da aplicación, que son:

- O rexistro e autenticación para o acceso á aplicación. (ilustración nº 11)
- As operacións CRUD cara a base de datos para acceder á información almacenada e manipulala (ilustración nº 12)
- As accións pertinentes na transmisión de mensaxes. (ilustración nº 13)

- **Rexistro, acceso á aplicación e aos recursos desta:**

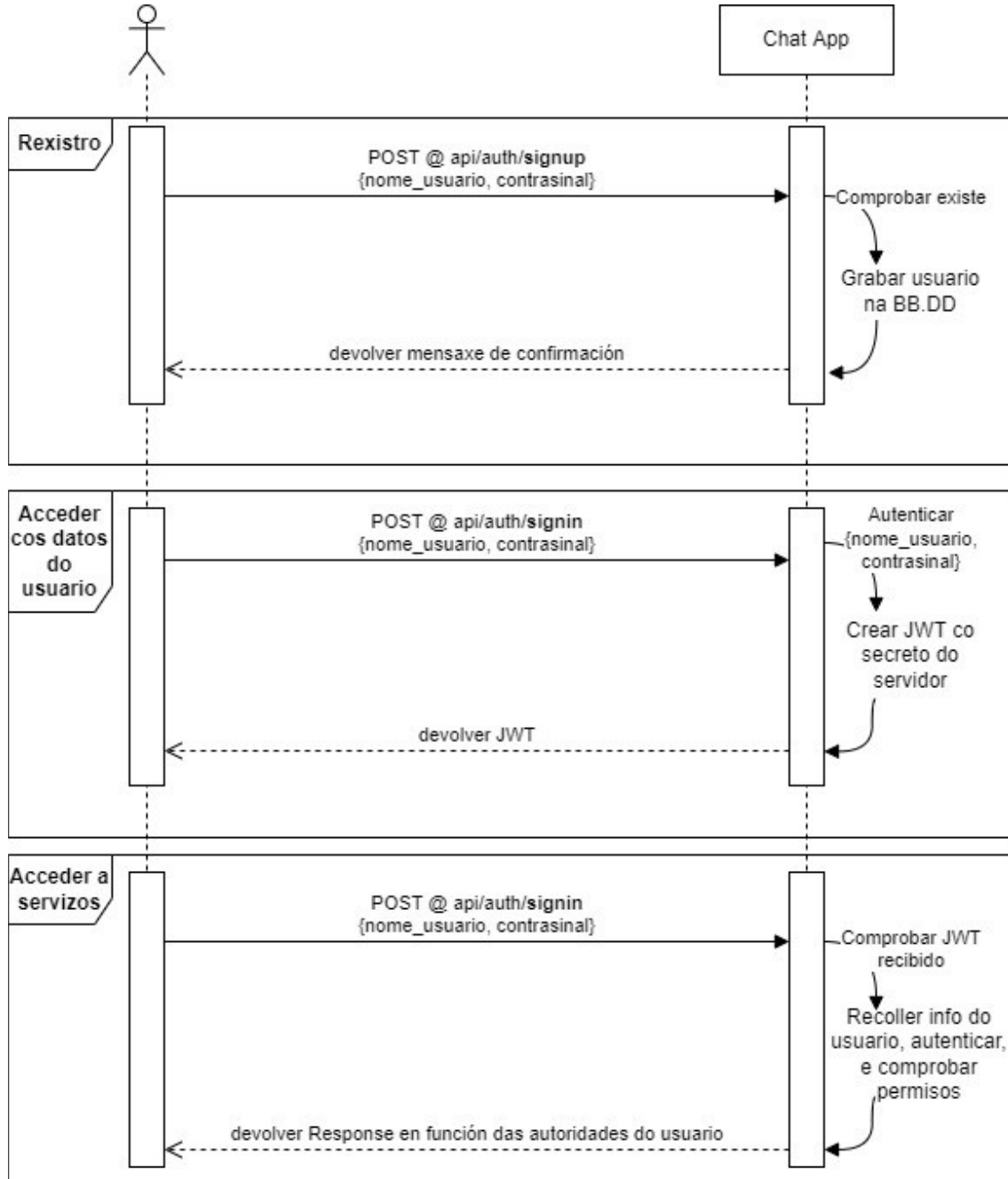


Ilustración 11: Diagramas de secuencia de rexistro e autenticación.

- **Operacións CRUD:**

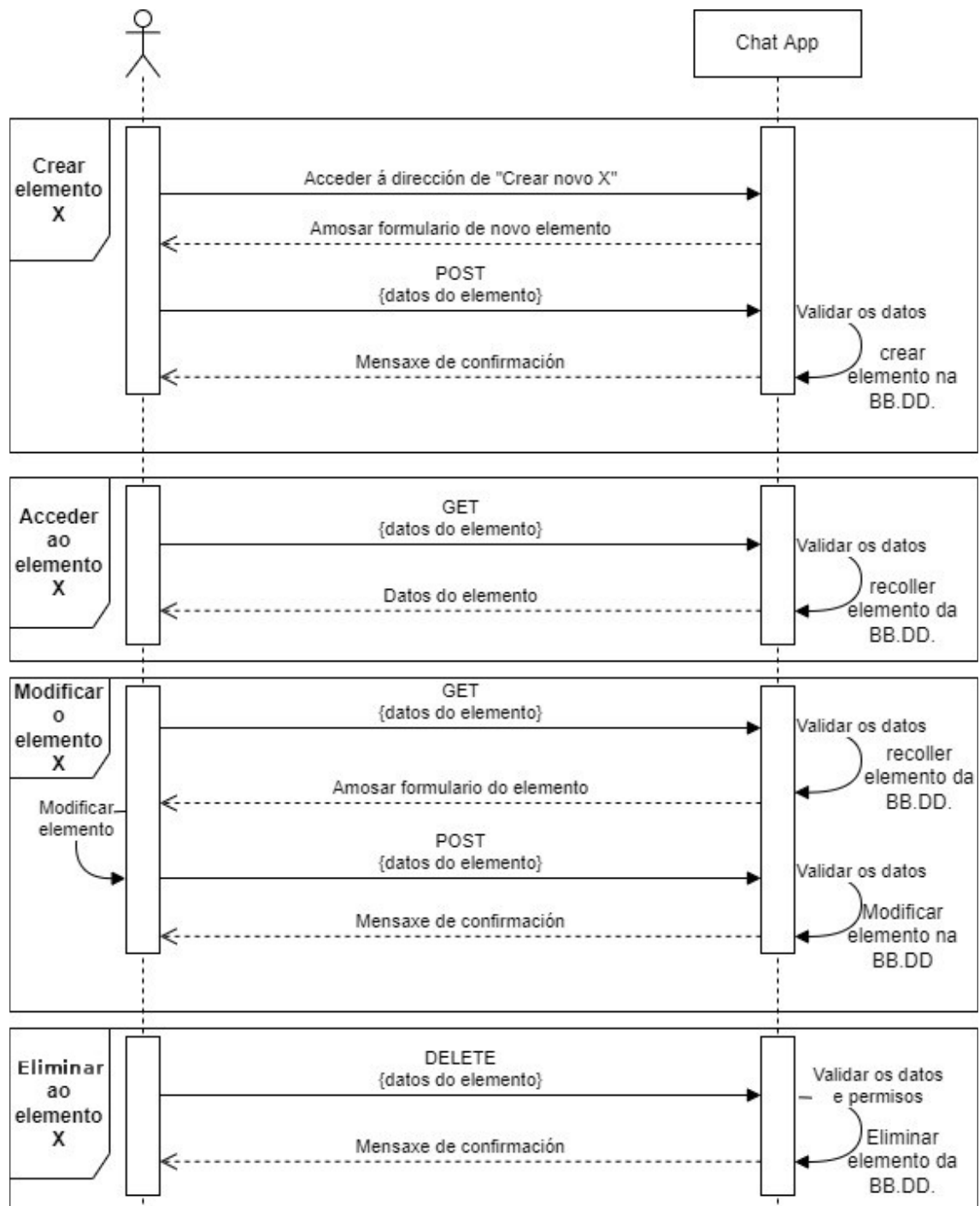


Ilustración 12: Diagramas de secuencia das accións comúns.

- **Accións na comunicación cos usuarios:**

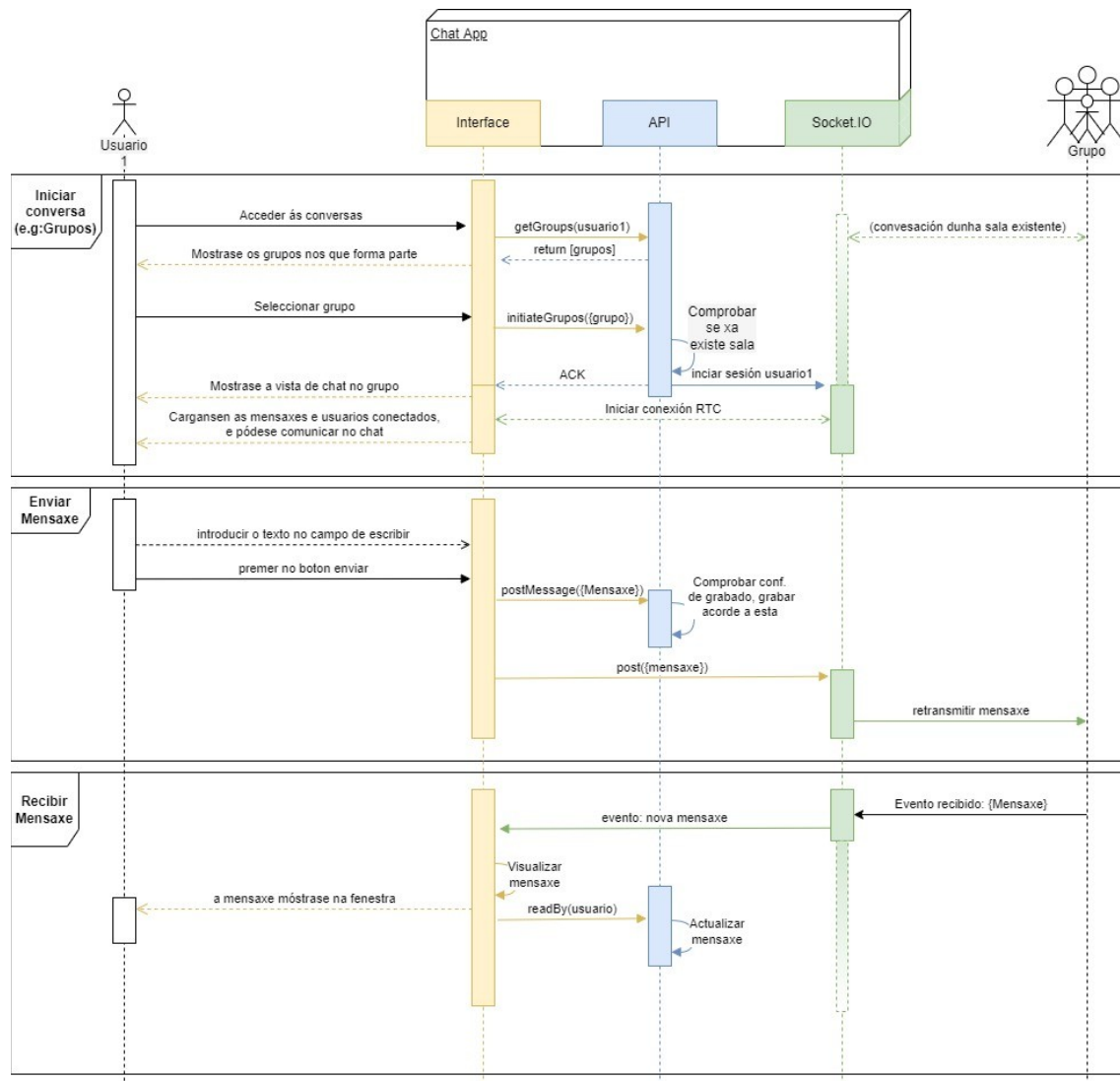


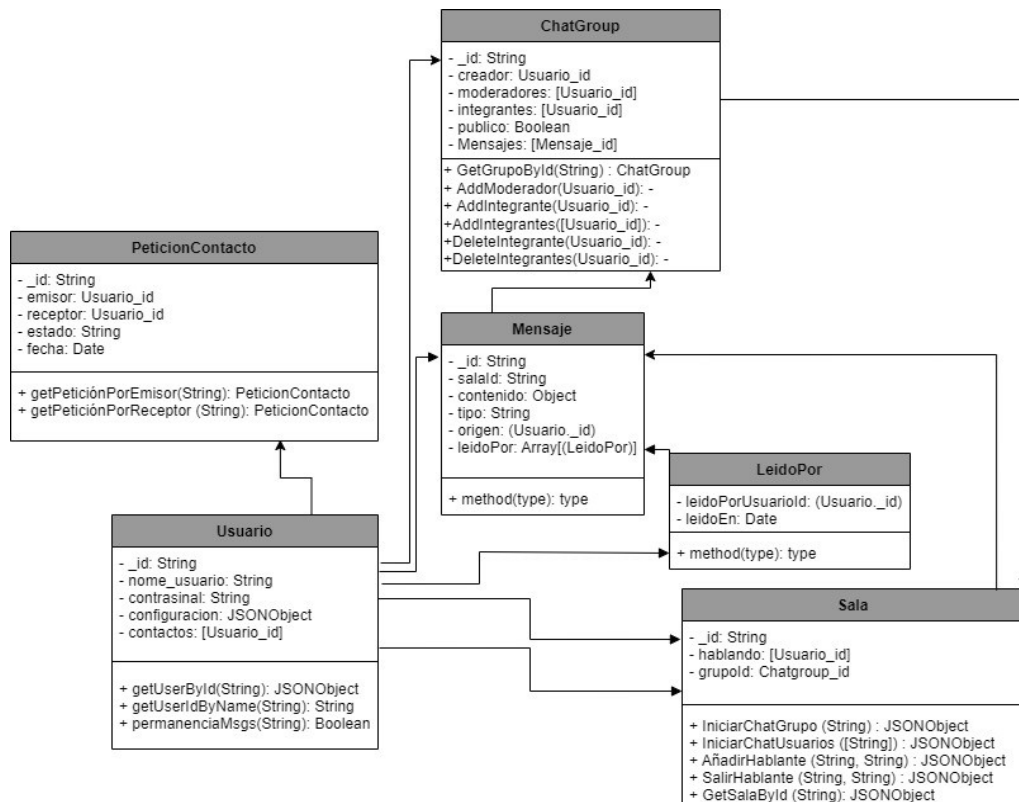
Ilustración 13: Diagrama de secuencia dunha conversa na aplicación.

9 Xestión de datos e información

9.1 Información do usuario

Respecto á información que o usuario introduce e recibe da aplicación, o sistema almacenará:

- **A información necesaria para identificar ao usuario no sistema**
 - Nome de usuario
 - Contraseñal
- **A información persoal do usuario**
 - A configuración establecida polo usuario
 - Os contactos do usuario.
- **As peticións de contacto entre usuarios e usuario-grupo**
 - Quén a fai
 - A quen vai dirixida
 - Aceptou, rexeitou , ou está sen confirmar a petición
- **As conversas realizadas no sistema**
 - Quén mandou a mensaxe
 - A quen estaba dirixida
 - Quén leeu a mensaxe
 - Cándo foi escrita



Para este proxecto, utilizarase MongoDB como xestor de información. Este programa utiliza un sistema de xestión de documentos, é dicir, os datos almacenados no sistema non son táboas coma nunha base de datos de, por exemplo, SQLite. Grazas a isto, podemos tratar a información en grupos, pero en cada documento pódese tomar certa liberdade co contido desta, o cal é moi útil para agrupar as mensaxes tanto privadas coma nos grupos nun mesmo esquema.

9.2 Información do sistema

Para as labores de mantemento, solución de problemas e seguridade, o sistema necesita almacenar:

- **Datos sobre as conexións recibidas**
 - Orixe da conexión
 - Data na que se fixo dita conexión
- **Datos sobre a información introducida na aplicación**
 - Que se fixo na conexión
 - Canto tempo se emprega na recepción e resposta
- **Datos sobre a información que o servidor devolve ao usuario**
 - Que fixo o servidor
- **Datos propios do servidor**
 - Rexistros dos erros que se producen
 - Cantas conexións realiza de media
 - Canto tarda de responder de media
 - Cantos recursos consume do servidor

Sen embargo, parte desta información supón unha intrusión na privacidade do usuario, polo que idéase encriptar previamente ao seu almacenaxe para que, no caso de que a información sexa comprometida, o impacto que lle supón ao usuario sexa o mínimo posible.

A continuación represéntanse os tipos de rexistros que se van almacenar:

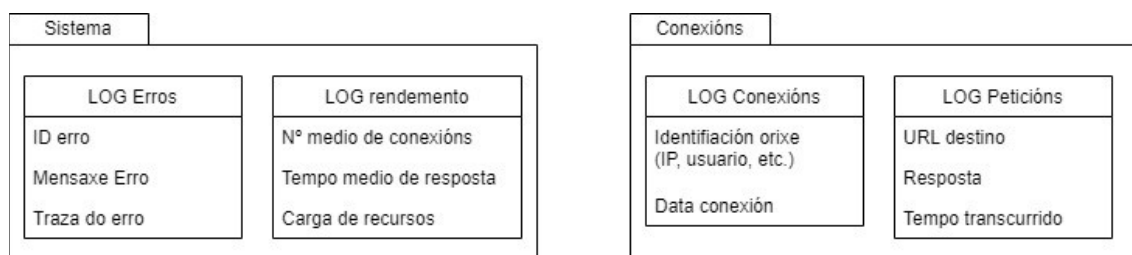


Ilustración 15: Diagrama de clases dos rexistros

10 Probas levadas a cabo

Para comprobar o correcto funcionamento dos compoñentes do sistema, así coma de comprobar se a experiencia de usuario é intuitiva, realizaranse tanto probas automatizadas coma manuais.

10.1 Probas automáticas

Son probas que, unha vez definidas, pódense realizar por si mesmas, é dicir, non é necesario que o programador interactúe co sistema mentres se realiza a proba.

Grazas a este sistema, formúlase tanto todos os casos de uso, así coma de as excepcións que poidan afectar ao correcto funcionamento do sistema. A continuación amósanse as probas realizadas:

	Afecta a	Descrición	Condición de éxito	Resultado obtido	Check
Acceso á Aplicación (OK)	Autenticación	Rexistro de usuario	Usuario agregado na base de datos	Usuario agregado na base de datos	OK
	Autenticación	Acceso á aplicación cun usuario existente	Retornase un JWT para o acceso ás funcións.	Retornase un JWT para o acceso ás funcións.	OK
	Autenticación	Cerrar sesión	Elimínase o token no navegador	Elimínase o token no navegador	OK
Acceso á Aplicación (Excepcións)	Auth – Rexistro	Algún dos campos obrigatorios está baleiro	Cancelar a execución no servidor e avisar ao usuario	Cancelar a execución no servidor e avisar ao usuario	OK
	Auth – Rexistro	Nome de usuario xa existente	Cancelar a execución no servidor e avisar ao usuario	Cancelar a execución no servidor e avisar ao usuario	OK
	Auth – Rexistro	Nome de usuario con caracteres non válidos	Cancelar a execución no servidor e avisar ao usuario	Cancelar a execución no servidor e avisar ao usuario	OK
	Auth – Rexistro	contrasinal non válida	Cancelar a execución no servidor e avisar ao usuario	Cancelar a execución no servidor e avisar ao usuario	OK
	Auth – Acceso	Intentar acceder sen contrasinal ou usuario	Avisar ao usuario	Avisar ao usuario	OK
	Auth – Saída	Cerrar o navegador sen pechar sesión	Eliminar o token do navegador	O token permanece no navegador	WARNING
Xestión do Usuario	Usuario	Enviar petición de contacto	Crease unha nova petición de contacto	Crease unha nova petición de contacto	OK
	Usuario	Modificar a configuración do usuario	Cámbiase a configuración do usuario	Cámbiase a configuración do usuario	OK
	Usuario	Eliminar a conta de usuario	Bórranse todos os rexistros do usuario nas BB.DD.	Bórranse todos os rexistros do usuario nas BB.DD.	OK
	Usuario	Aceptar petición de contacto	Agrégase o contacto á lista destes do usuario	Agrégase o contacto á lista destes do usuario	OK
	Usuario	Rexeitar petición de contacto	Elimínase a petición de contacto	Elimínase a petición de contacto	OK
Xestión dos Grupos	Grupo	Crear Grupo de conversa	Crease un novo grupo	Crease un novo grupo	OK
	Grupo	Engadir integrante ao grupo	Agrégase o contacto á lista de integrantes	Agrégase o contacto á lista de integrantes	OK
	Grupo	Eliminar integrante do grupo	Elimínase o contacto da lista de integrantes	Elimínase o contacto da lista de integrantes	OK
	Grupo	Engadir moderadores do grupo	Agregasen os contactos á lista de moderadores	Agregasen os contactos á lista de moderadores	OK
	Grupo	Eliminar moderadores do grupo	Elimínase os contactos da lista de moderadores	Elimínase os contactos da lista de moderadores	OK
	Grupo	Eliminar grupo	Bórranse todos os rexistros do grupo nas BB.DD.	Bórranse todos os rexistros do grupo nas BB.DD.	OK
Servizo de mensaxería	Sala	Iniciar chat privado	Crease unha sala se é necesario para o chat ou devolve os datos da sala do chat	Crease unha sala se é necesario para o chat ou devolve os datos da sala do chat	OK
	Sala	Iniciar chat de grupo	Crease unha sala se é necesario para o chat ou devolve os datos da sala do chat	Crease unha sala se é necesario para o chat ou devolve os datos da sala do chat	OK
	Sala	Conectarse a unha conversa	crea unha conexión RTC coa conversa	crea unha conexión RTC coa conversa	OK
	Sala	Salir dunha conversa	Pecha a conexión RTC coa conversa	Pecha a conexión RTC coa conversa	OK
	Mensaxe	Enviar mensaxe	Enviase a mensaxe aos integrantes do grupo	Enviase a mensaxe aos integrantes do grupo	OK
	Mensaxe	Mensaxe Recibida	O usuario recibe unha mensaxe da conversa	O usuario recibe unha mensaxe da conversa	OK
	Mensaxe	Eliminar Mensaxe	Elimínase unha mensaxe da conversa	Elimínase unha mensaxe da conversa	OK

Táboa 3: Probas automatizadas levadas a cabo

11 Manual de usuario

Hay dous manuais, dirixidos para:

1. Calquera que desexe despregar a aplicación nun entorno propio para servir a app a unha comunidade
2. Os usuarios finais que utilizarán a aplicación

11.1 Manual do xestor da aplicación

Requisitos Previos

- **S.O.:** calquera capaz de executar Node.
- **Software adicional:** Node, v.14.x.x ou superior
- **Hardware e similares:**
 - Conexión a unha rede LAN (para uso local), ou acceso a internet (para uso global)
 - Computadora, sexa específica para servidores ou persoal, para manter a execución da aplicación
- **Outros:** Permisos de administrador no dispositivo

Instalación

- Obter a última versión do proxecto^[23]
- Se quere facer a instalación manual:
 - Abrir unha terminal nova (powershell ou cmd en Windows, bash en Linux/MacOS)
 - no código fonte, acceder á carpeta do servidor (server), e executar o seguinte comando:
(.../server)> npm install
 - para executar o servidor, lanzar a seguinte instrución:
(.../server)> npm start
 - Se quere que o servidor se execute en segundo plano, proceder cós mecanismos do SO para dita función, por exemplo:
 - En Linux:
 - (opc.1)“&” ao final da liña en Linux e acto seguido executar:
(.../server)> disown -h %1
//Este comando fai que o proceso xa non forme parte dos traballos do shell no que se executou, para que aínda que peche a terminal, o proceso non se termine con este peche.
 - (opc.2)Utilizando nohup:
(.../server)> nohup npm start &

//Este comando permite a execución da orde aínda que peche a terminal se execute de forma independente da sesión. A saída (output) da execución almacenase nun ficheiro chamado nohup.out, que créase, por defecto, na ruta onde nos atopemos ao executar o comando

Configuración

Cando execute a aplicación por primeira vez, é moi probable que reciba unha mensaxe de erro similar á seguinte:

```
No se ha podido conectar con la base de datos: MongoParseError: Invalid scheme, expected connection string to start with "mongodb://" or "mongodb+srv://"
at new ConnectionString (C:\Users\dani\Desktop\TFG-090622\Proyecto\Software\V1.0.1\server\node_modules\mongodb-connection-string-url\lib\index.js:86:19)
at parseOptions (C:\Users\dani\Desktop\TFG-090622\Proyecto\Software\V1.0.1\server\node_modules\mongodb\lib\connection_string.js:213:17)
at new MongoClient (C:\Users\dani\Desktop\TFG-090622\Proyecto\Software\V1.0.1\server\node_modules\mongodb\lib\mongo_client.js:62:63)
at C:\Users\dani\Desktop\TFG-090622\Proyecto\Software\V1.0.1\server\node_modules\mongoose\lib\connection.js:784:16
at new Promise (<anonymous>)
at NativeConnection.Connection.openUri (C:\Users\dani\Desktop\TFG-090622\Proyecto\Software\V1.0.1\server\node_modules\mongoose\lib\connection.js:781:19)
at C:\Users\dani\Desktop\TFG-090622\Proyecto\Software\V1.0.1\server\node_modules\mongoose\lib\index.js:342:10
at C:\Users\dani\Desktop\TFG-090622\Proyecto\Software\V1.0.1\server\node_modules\mongoose\lib\helpers\promiseOrCallback.js:32:5
at new Promise (<anonymous>)
```

Ilustración 16: Exemplo de erro ao conectar coa base de datos

Este erro débese a que non hai unha configuración válida da base de datos. Isto é completamente normal: a aplicación non conta cun servidor centralizado, nin dunha red descentralizada para manter unha serie de bases estandarizadas, polo que recae a responsabilidade de habilitar dita conexión ao xestor do servizo.

Debido a isto, é necesario que o xestor da aplicación teña, ou ben contratado ou ben creado por el, unha base de datos MongoDB. Unha vez creada, o xestor debe crear un ficheiro chamado “.env” na carpeta do servidor, onde se almacenan as seguintes variables (o formato de declaración é: NOME_VARIABLE=valor , unha variable por liña):

- **SECRET:** unha secuencia de caracteres, os cales utilízanse para as comunicacións servidor-cliente para encriptar os contrasinais.
- **DB_HOST:** Identificador do dispositivo onde se atopa a BB.DD.
- **DB_USER:** Usuario con permisos de escritura e lectura á BB.DD.
- **DB_PASS:** Contraseñal do usuario
- **DB_NAME:** Nome da BB.DD.
- **DB_URL:** (Opcional) Dirección dunha BB.DD secundaria, para que no caso de non poder entaboar conexión coa DB_URL_ATLAS ter unha segunda opción
- **DB_URL_ATLAS:** Dirección principal da BB.DD
- **PORT:** porto polo que vai escoitar a aplicación (se non se especifica, por defecto 8080). Se escolle un porto reservado (80), é necesario configurar ó dispositivo para permitir devandita conexión.

Outra capacidade que pode ofrecer o servidor é o servizo de API exclusivamente. Isto habilita a capacidade de deseñar unha interface nativa a calquera dispositivo, comunicando co servidor a través da API. Se a interface procede doutra fonte, é necesario configurar correctamente CORS (do inglés “Cross Origin Resource Sharing”) para poder servir os datos á interface.

A URL para poder acceder á aplicación, por defecto, será a dirección “localhost”. Para servir públicamente o servidor, é necesario configurar a máquina que aloxa o programa para transmitir ao programa a dirección IP do dispositivo.

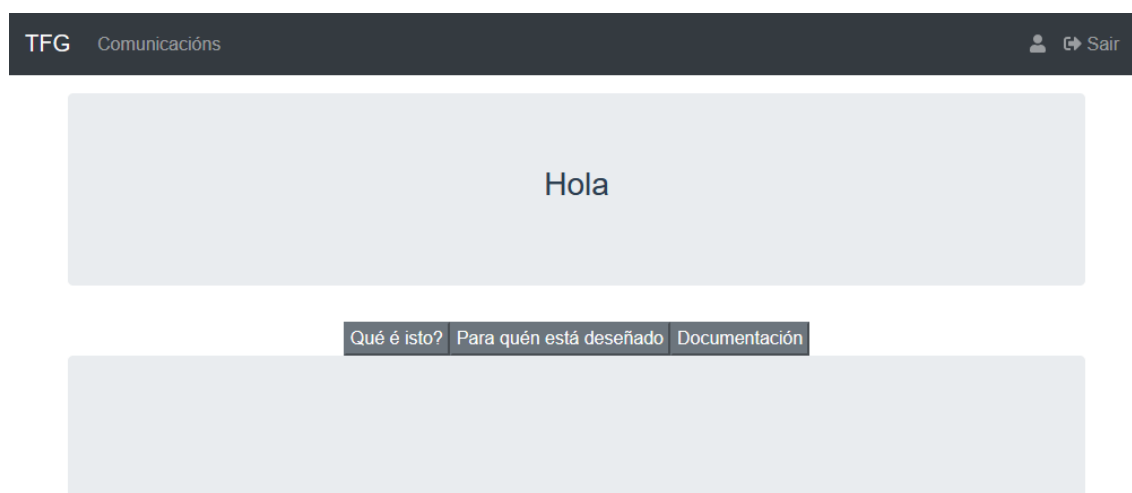
Cambiar a interface

Se o desexa, pode modificar a interface da aplicación, xa que a actual é moi minimalista. Para isto, é necesario cambiar o contido do cartafol “public”, e axustar as rutas URL do servidor no caso de ter unha interface de múltiples páxinas, para servir devanditas páxinas.

11.2 Manual do usuario

Benvido ao manual de usuario desta aplicación. Aquí explicaremos como navegar pola aplicación, e que pode facer nesta.

Cando se conecte á aplicación, amosarase a páxina de inicio da aplicación:



Se trata de acceder a un punto da páxina que require autenticaci3n sen esta, será redirixido á paxina de autenticaci3n de usuario. Nótese que, se ao pasar por un campo do formulario, informará no caso de que os datos introducidos sexan válidos:

Red de mensajería instantánea e privada

TFG Rxistrarse Acceder



Nome de usuario

CAMPO OBRIGATORIO

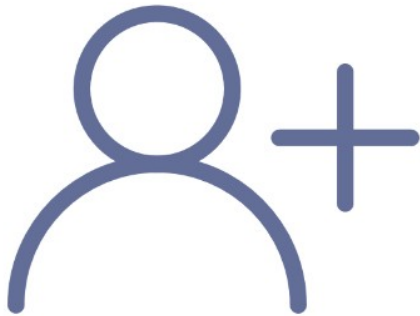
Contrasinal

CAMPO OBRIGATORIO

Acceder

Se todavía non conta cunha conta, pode rexistrarse na aplicación facendo click no menú superior, en “Rexistrarse”:

TFG Rxistrarse Acceder



Nome de usuario

Contrasinal

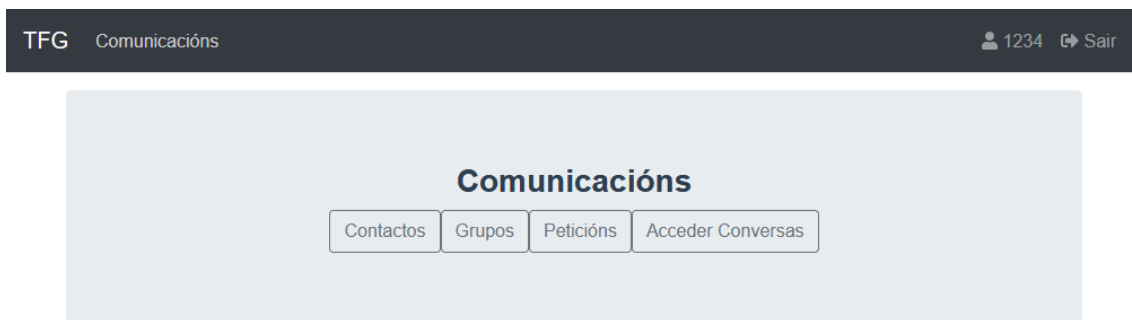
Rxistrarse

Unha vez rexistrado, e ao acceder coa súa conta, pode observar que cambia os elementos na barra de navegación:



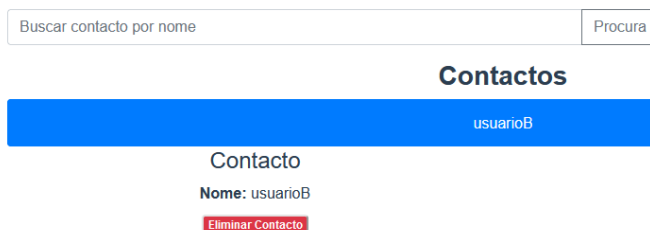
- Á **esquerda**, no apartado **“Comunicacións”** podemos acceder ás funcións da aplicación.
- Á **dereita**, no nome do usuario e o icono de persoa, podemos acceder ao **perfil do usuario** co que entramos á aplicación, ou **pechar a sesión** no botón “Sair”, á dereita deste.

Para acceder ás funcionalidades, clickamos en “Comunicacións”, onde se amosa a seguinte vista:



Dende aquí podemos, facendo click nos botóns:

- Acceder á lista de contactos agregados, onde podemos tamén eliminar contactos se así o desexa o usuario



Red de mensajería instantánea e privada

- Acceder á lista de grupos dos que se forma parte, acceder a estes para ver máis detalles e, se é o dono do grupo, convertir a moderadores aos integrantes que desexe.

Grupos

Crear Grupo

grupo1

Grupo

Nombre: grupo1

Integrantes:

Editar

BORRAR GRUPO

Creador

usuarioA

Moderadores

Añadir Moderadores

Integrantes

Añadir Usuarios

usuarioA

- Acceder á lista de peticións recibidas, ou para realizar unha petición de contacto (ou grupo).

Peticións de Contacto

usuarioB

Petición

Nome: usuarioB

Data: 2022-06-30T19:41:43.297Z

Aceptar Denegar

- Acceder ao chat, ben sexa cun usuario ou grupo.

Red de mensajería instantánea e privada

Conversas

Grupos

Usuarios

grupo1

Unha vez dentro dunha conversa, accederá á vista de chat:

Chat

Conectados:

• Yo

Yo

Ola

12 Principais contribucións

Este proxecto ofrece unha plataforma sinxela de executar, coa capacidade de habilitar unha canle de comunicación en tempo real. Sen embargo, o obxectivo do proxecto non é ofrecer o servizo; o obxectivo principal deste proxecto é involucrar, tanto aos usuarios do servizo coma ao xestor deste, no desenvolvemento dun servizo de comunicación para adaptalo ás necesidades de dita comunidade.

A continuación lístanse as principais achegas da aplicación:

- **Intuitivo**
A interface está pensada para ser o máis sinxela e accesible posible para os usuarios con pouca experiencia coas novas tecnoloxías.
- **Educativo**
Outro obxectivo do proxecto é involucrar ao xestor do servizo no proceso de implantación, pero só no estritamente necesario para incentivar a curiosidade deste, e incentivalo a traballar nun sistema propio.
- **Cooperativismo**
Dado que o sistema require de usuarios que se comuniquen no sistema para manter a razón de existir da aplicación, o xestor debe comunicarse cos seus usuarios para manter a estes, así como de adecuar a aplicación ás necesidades dos usuarios, e non só ás necesidades que considera este.
- **Competitividade**
Como o proxecto é aberto ao público, calquera pode crear a súa propia instancia, se o proxecto alcanza un grao de visibilidade notorio, os usuarios terán múltiples instancias, cada unha máis personificada ou adecuada de forma distinta. Os sistemas que se mostren máis atractivos aos usuarios serán os que acheguen máis usuarios, polo que motivará aos xestores a modificar a aplicación acorde ás costumes do conxunto de usuarios.

13 Conclusións

O resultado final do proxecto é unha aplicación sinxela e relativamente pequena (< 30 MB) que habilita a comunicación entre usuarios a tempo real en forma de texto plano. No apartado de funcionalidades, atópase algo limitado, pero grazas á elección dos paquetes e frameworks, actualizar calquera dos compoñentes da aplicación é moito máis sinxelo, polo que pódese invertir máis tempo en planificar .

13.1 Conclusións Técnicas

A decisión de linguaxe de programación (JavaScript) fíxose en base á sinxeleza deste, para aprender en máis profundidade as capacidades que ten, e o feito de que, xunto a Node.js e Vue, pódese realizar o stack completo da aplicación coa mesma linguaxe.

A facilidade coa que se pode replicar o proxecto nun equipo distinto é tamén un factor importante, aínda que moitas xa contan cun sistema para empaquetar o proxecto, listo para usalo en distintos sistemas operativos.

Tamén cabe destacar o papel da librería Socket.IO, coa cal simplificouse o desenrolo das funcións respecto ás comunicacións inmediatas, e tamén á capacidade de MongoDB de permitir que os datos non teñan un esquema estricto; isto axudou a adecuar os modelos, sen a necesidade de ter que resetear a BB.DD. reiteradas veces, facilitando a labor de desenrolo do proxecto.

13.2 Conclusións Persoais

O que pretendía lograr con este proxecto era aprender a desenvolver un proxecto pleno, e ao mesmo tempo adquirir novos coñecementos doutras linguaxes de programación, entorno e prácticas.

Comprendín a necesidade de planificación e correcta xestión do traballo para asegurar un correcto desenrolo do proxecto. Unha preparación previa asegura a capacidade de poder seguir unha pauta establecida, coa capacidade de poder reverter a unha versión previa no proceso de desenrolo en caso de problemas e probar unha nova pauta.

Tamén, aínda que non se aplicou correctamente o proceso RUP, o feito de ter pequenos incrementos axuda a crear unha sucesión de metas máis alcanzables, axudando a facer predicións de tempo empregado en cada sección.

14 Vías de traballo futuro

A continuación listasen as posibles ampliacións deste proxecto, ordeadas segundo a viabilidade inmediata:

1. Engadir interoperabilidade entre instancias

Poderíase contemplar a posibilidade de permitir conectar os servidores entre eles para habilitar a capacidade de comunicarse entre distintos grupos definidos en distintos sistemas, sen a necesidade de ter que cambiar de conexión. Isto ofrece unha mellora de accesibilidade, así de permitir a varios xestores coexistir sen ser competencia, exclusivamente.

2. Engadir a posibilidade de comunicarse por voz ou vídeo

A capacidade de comunicarse de máis formas que escrita atrae a un grupo moito máis amplo que só coa capacidade de comunicarse a través de texto. Tamén pódese engadir a capacidade de compartir documentos ou ficheiros, pero sería requirido buscar un procedemento no cal asegurase que os documentos non son almacenados, pero de todas formas é posible acceder a eles (unha opción sería habilitar o acceso ao documento directamente do usuario, pero este método implica moitas vulnerabilidades)

3. Transformar o proxecto para convertelo nunha integración

Dado que o mercado para aplicacións de comunicación xa conta con varias solucións que ofrecen todo o ofrecido por esta, unha utilidade que permitiría esta aplicación sería integrala noutra aplicación distinta (coma un videoxogo ou na aplicación dunha empresa para habilitar un medio de comunicación rápido para os empregados).

Outra vía similar sería habilitar a aplicación para integrar varias aplicacións de mensaxería, e transformar a aplicación nun xestor de comunicacións privado.

15 Referencias

1. Estadísticas de uso de Whatsapp - <https://www.whatsapp.com/about/> [Última visita: 30/10/2021]
2. página oficial de Signal - <https://signal.org/es/> [Última visita: 30/10/2021]
3. “Las tecnologías de la información y las comunicaciones como apoyo a las actividades internacionales y al aprendizaje a distancia en las universidades “, Jorge Luis Quintero Barrizote, Universidad de Cienfuegos
(http://scielo.sld.cu/scielo.php?pid=S2218-36202020000100366&script=sci_arttext&lng=pt)
[Última visita: 02/02/2022]
4. “La inversión en ciberseguridad crecerá hasta los 250 mil millones de dólares en 2023” – NoticiasVigo.es – 07/11/2018 [Última visita: 30/10/2021]
5. Qué es una api?”, red Hat, 31/10/2017
<https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces> [Última visita: 20/10/2021]
6. información sobre servidores web:
https://developer.mozilla.org/es/docs/Learn/Common_questions/What_is_a_web_server , 11/02/2021
7. Descripción de API de REST , red hat, 08/05/2020
<https://www.redhat.com/es/topics/api/what-is-a-rest-api> [Última visita: 30/10/2021]
8. Documentación de socket.io: <https://socket.io/docs/v4/> [Última visita: 30/10/2021]
9.
 1. “WebSockets vs long polling”, Kieran Kilbride-Singh, 27/10/2021 - <https://ably.com/blog/websockets-vs-long-polling> [Última visita: 30/10/2021]
 2. ”The WebSocket Protocol”, A. Melnikov, 2011
<https://datatracker.ietf.org/doc/html/rfc6455> [Última visita: 30/10/2021]
 3. “Main goal of WebSocket protocol”, IETF
<https://trac.ietf.org/trac/hybi/wiki/FAQ> [Última visita: 30/10/2021]

10. "Comparative Study of database modeling approaches" , Mohammed Aljarallah, Department of Computer Science, Algoma University, April 2014 (Pag. 36 - 40)
11. ["The Rational Unified Process An Introduction" 3ª edición, philippe kruchten, isbn 0-321-19770-4, library of congress cataloging-in-publication data, kruchten, philippe](#)
12. <https://books.google.es/books?hl=es&lr=&id=RYCMx6o47pMC&oi=fnd&pg=PR13&dq=rational+unified+process+papers&ots=h3fpxQKbSh&sig=ImcioxXyYX2k3opVXyCv9Ldz9l0#v=onepage&q=rational+unified+process+papers&f=false>
[Última visita: 30/10/2021]
13. Página oficial de Node: <https://nodejs.org/es/> [Última visita: 30/10/2021]
14. Documentacion de express.js <http://expressjs.com/> [Última visita: 1/11/2021]
15. Paquete Bcrypt - <https://www.npmjs.com/package/bcrypt> [Última visita: 30/10/2021]
16. "JSON Web Token", ISSN: 2070-1721
<https://datatracker.ietf.org/doc/html/rfc7519> , [Última visita: 20/12/2021]
17. Página oficial de Vue - <https://vuejs.org/> [Última visita: 20/12/2021]
18. Iniciativa Open Source, página oficial: <https://opensource.org/> [Última visita: 20/12/2021]
19. Dabbagh, Mohammad & Lee, Sai & Parizi, Reza. (2015). Functional and non-functional requirements prioritization: empirical evaluation of IPA, AHP-based, and HAM-based approaches. Soft Computing. 20. 10.1007/s00500-015-1760-z. [Última visita: 20/12/2021]
20. Página oficial do motor v8: <https://v8.dev/> [Última visita: 20/12/2021]
21. Declaración do modelo MVC,
<https://folk.universitetetioslo.no/trygver/themes/mvc/mvc-index.html>
[Última visita: 20/12/2021]
22. Jarallah AlGhandi and Joseph Urban. "Comparing and assessing programming languages: Basis for a qualitative methodology". In Proc. 1993 Software Applications Conference [Última visita: 20/12/2021]

23. Tesis “Performance comparison of XHR polling, Long polling, Server sent events and Websockets” , Rasmus Appelqvist & Oliver Örnmyr, Faculty of Computing Blekinge Institute of Technology SE-371 79 Karlskrona Sweden [Última visita: 20/12/2021]
24. Dirección do proxecto : <https://github.com/dangr400/chat-TFG> [Última visita: 20/5/2022]

16 Anexos

Socket.IO

É unha librería que habilita unha canle de comunicación bidireccional, baseada en eventos e de baixa latencia. Utiliza un protocolo con base en WebSocket e provee funcionalidades adicionais, coma:

- **retroceso a Long-Polling** no caso de non poder utilizar devandito protocolo,
- **reconexión automática.**
- **Buffering de paquetes**, para reenvialos cando volva a conexión.
- **Difusión de mensaxes**

Sen embargo, hai que remarcar que esta librería utiliza a tecnoloxía de WebSocket, **non utiliza o protocolo WebSocket** “per se”, senon que os paquetes conteñen metadata adicional, polo que non se pode interconectar cun sistema de websockets.

WebSocket

Protocolo que habilita unha conexión bidireccional utilizando unha única conexión TCP para aforrar o custo de ancho de banda e de procesamento ao utilizar métodos tradicionais no traspaso de información. Consta de dúas partes:

1. **Apertón de mans (handshake)**, encargado de iniciar a conexión entre o cliente e o servidor.
2. **Transferenza de información**, onde iniciase a transmisión de paquetes entre os clientes e os servidores.

Este protocolo busca eliminar o exceso de cabeceiras que son necesarias no protocolo HTTP para poder manter a mesma capacidade de comunicación en tempo real.

HTTP Long-Polling

É un método que procura minimizar a latencia e uso de recursos de procesamento e rede, mantendo unha rede de baixa latencia. respondendo únicamente a unha petición HTTP cando un evento, status, ou un tempo morto. O método pódese resumir nos seguintes pasos:

1. O cliente realiza unha **petición inicial** e **agarda** pola resposta.
2. O servidor **apraza a resposta** ata que haxa unha actualización dispoñible, ou ata que ten lugar un tempo morto ou status.

3. Cando haxa unha actualización, o servidor **envia unha resposta** completa ao cliente.
4. O cliente **envía unha nova petición** de long poll, ou ben inmediatamente ao recibir a resposta ou despois dunha pausa para permitir un período de latenza razoable.