

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

**Título do Traballo de Fin de Grado**



Novembro, 2021

**Traballo de Fin de Grao N°:**

**Titor/a:** Francisco José Ribadas Pena

**Área de coñecemento:** Linguaxes e Sistemas Informáticos

**Departamento:** Informática

## Dedicatoria

## Agradecimentos

## Índice

1	Introdución.....	6
1.2	Uso e aplicación dos servizos de mensaxería.....	6
2	Obxectivos.....	7
2.1	Resumo da solución.....	7
2.1.1	Servidor Web.....	7
2.1.2	Cliente.....	7
2.1.3	Base de Datos.....	8
3	Planificación e Seguimento.....	9
4	Arquitectura.....	11
4.1	Arquitectura do servidor.....	12
5	Tecnoloxías, Ferramentas e Integracións de produtos de terceiros.....	12
5.1	Tecnoloxías Integradas no Proxecto.....	12
5.2	Ferramentas.....	13
6	Especificación e análise de requisitos.....	14
6.1	Actores.....	14
6.2	Requirimentos.....	14
6.2.1	Requirimentos Funcionais.....	14
6.2.2	Requirimentos Non Funcionais.....	15
6.3	Diagramas de casos de uso.....	15
6.4	Descrición de casos de uso.....	16
7	Deseño de software (estático e dinámico).....	20
8	Xestión de datos e información.....	21
9	Probas levadas a cabo.....	22
10	Manual de usuario.....	23
11	Principais contribucións.....	24
12	Conclusións.....	25
13	Vías de traballo futuro.....	26
14	Referencias.....	27
15	Anexos.....	28
	Socket.IO.....	28
	WebSockets.....	28
	HTTP Long-Polling.....	28
	RUP (Rational Unified Process).....	28

## Índice de Ilustracións

Ilustración 1: Diagrama de Gantt do tempo estipulado.....	10
Ilustración 2: Esquema xeral da arquitectura do sistema.....	11

## Índice de tablas

Tabla 1: Tempos estimados de desenrolo do proxecto.....	10
---	----



# 1 Introducción

Un dos desenvolvementos máis importantes do ser humano é a capacidade de conversar con outro individuo, sexa a distancia que sexa, o máis rápido posible. Hoxe en día xa temos interiorizado que é normal poder falar con quen sexa, cando sexa, pero os avances necesarios para que isto fose posible son numerosos (medios de transmisión, criptoloxía, codificación, etc.)

Na actualidade, os métodos de comunicación a distancia máis comúns son o correo electrónico, as chamadas telefónicas e os servizos de mensaxería instantánea. No sector de mensaxería, os máis coñecidos son Whatsapp<sup>[1]</sup>, Telegram (400 millóns de usuarios). Se ben estas aplicacións ofrecen outros métodos de comunicación (chamadas, tanto de voz coma de vídeo), o seu auxe comezou cós mensaxes escritos.

A pesar de que o servizo sexa de comunicar os usuarios entre eles, hai un incremento na preocupación dos usuarios de que os seus datos sexan utilizados con fins alleos, ben sexa para crear un perfil do usuario, recopilar datos ou, nos casos máis extremos, controlar as conversas da xente.

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.

## 1.2 Uso e aplicación dos servizos de mensaxería

## 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 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 **API** (Application Programming Interface).

### 2.1 Resumo da solución

A solución proposta constará dunha API servida a través dun servidor HTTP, a cal responderá ás peticións realizadas polos usuarios a través do cliente. A continuación ampliáanse os detalles sobre o servidor e o cliente:

#### 2.1.1 Servidor Web

O servidor ofrecerá a posibilidade de recibir e responder a peticións a través de HTTP, grazas a una API REST. 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 para realizar unha comunicación a tempo real entre os interlocutores.

#### 2.1.2 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, utilizarase 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.

### **2.1.3 Base de Datos**

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



### 3 Planificación e Seguimento

O proxecto levarase a cabo seguindo una metodoloxía RUP (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 desenrolo dunha IU básica, coa que comprobaremos que, unha vez accedido ao servidor, visualizaranse 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 rexistrárase 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.

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

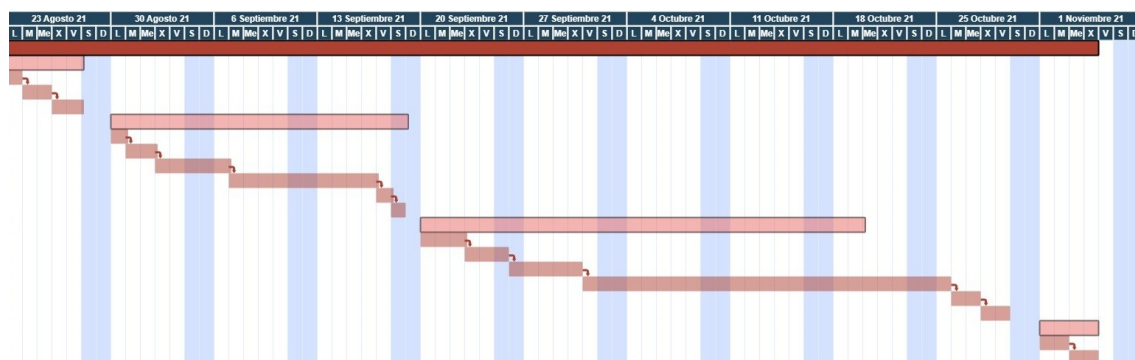


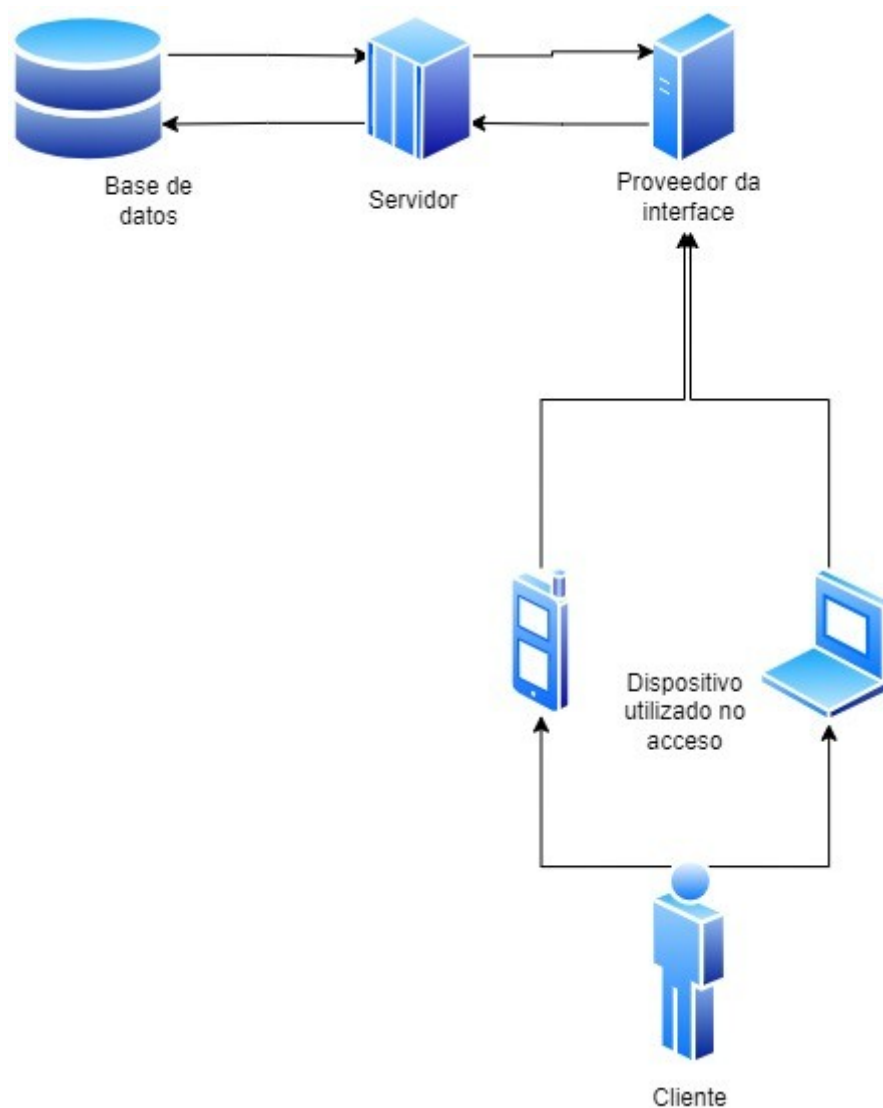
Ilustración 1: Diagrama de Gantt do tempo estipulado

## 4 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, 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 desenvolvida ofrécese 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 2: Esquema xeral da arquitectura do sistema*

## 4.1 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 Tecnoloxías, Ferramentas e Integracións de produtos de terceiros

## 5.1 Tecnoloxías Integradas no Proxecto

- **Node.js**

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.

- **Express**

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

- **Bcryptjs**

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**

Módulo que implementa JSON Web Token en Node.

- **Mongoose**

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**

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

- **Vue**

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.

## 5.2 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**

Base de datos situada na nube, para almacenar os datos persistentes da aplicación.

- **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, para a documentación do avance do proxecto.

## 6 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:

### 6.1 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.

### 6.2 Requirimentos

Divídense os requirimentos en dous grupos distintos:

#### 6.2.1 Requirimentos Funcionais

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 modificálos.

- **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.

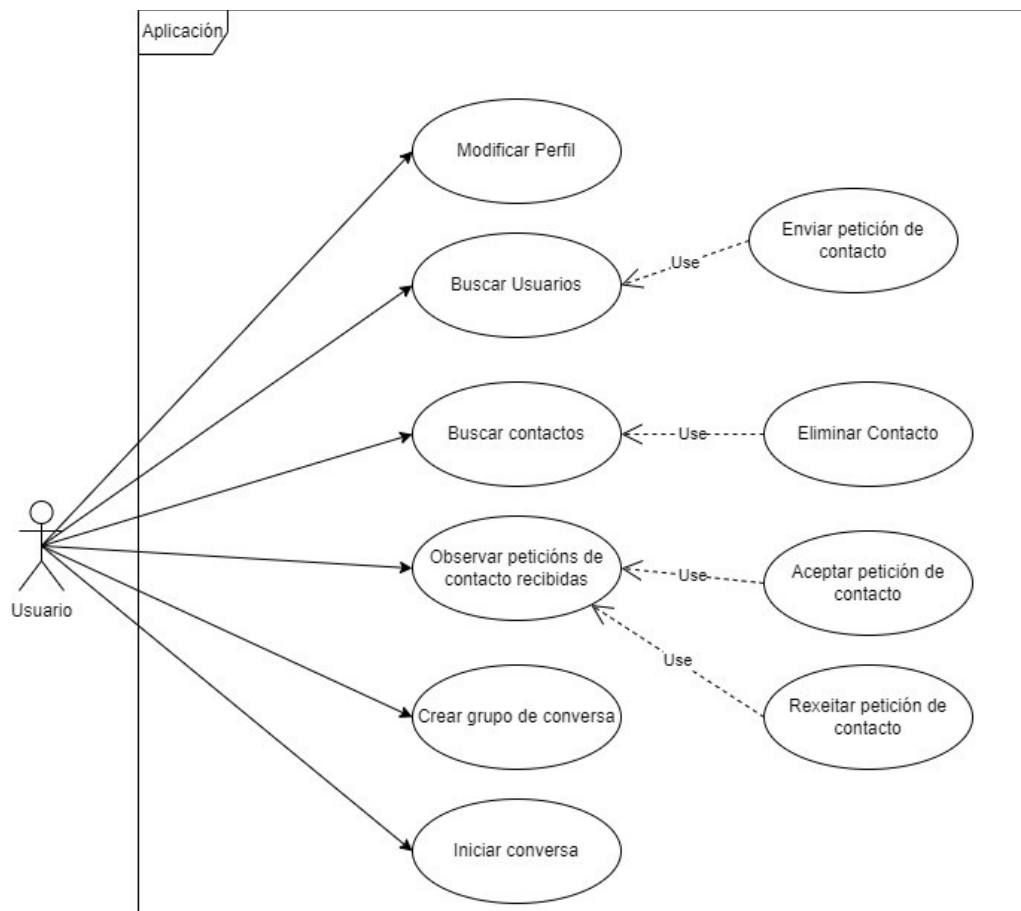
## 6.2.2 Requisitos Non Funcionais

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. Os requirimentos non funcionais son:

- **Disponibilidade**
- **Compatibilidade**
- **Robustez**
- **Seguridade**

## 6.3 Diagramas de casos de uso

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



*Ilustración 3: Diagrama de casos de uso dos usuarios*

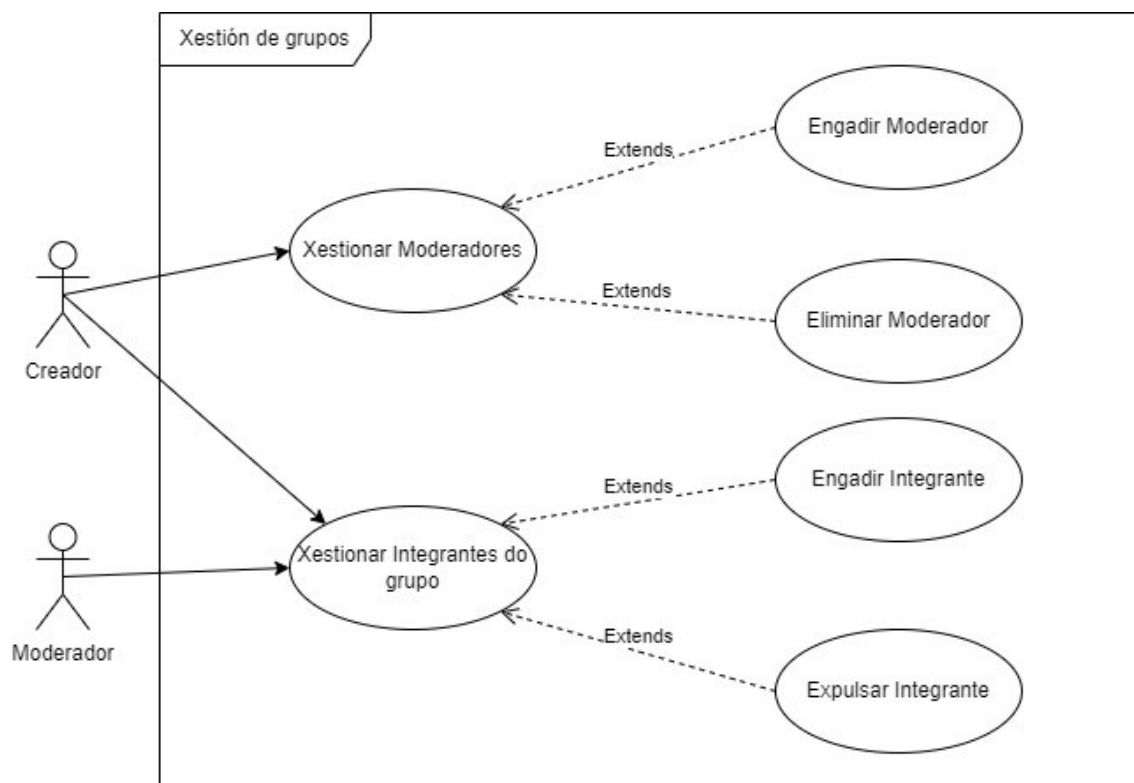


Ilustración 4: Diagrama de casos de uso na xestión de grupos

## 6.4 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
Descrición		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	
Descrición	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

## **7 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 mediante diagramas de clases , e as interaccións entre eles, con diagramas de secuencia.

## **8 Xestión de datos e información**

*DESCRIBIRANSE OS MÉTODOS OU TÉCNICAS EMPREGADAS PARA XESTIONAR TANTO OS DATOS  
COMA O RESTO DE INFORMACIÓN RELEVANTE*



## **9 Probas levadas a cabo**

*DESCRIBIRANSE AS PROBAS REALIZADAS AOS DISTINTOS NIVEIS PARA GARANTIR O CORRECTO  
FUNCIONAMENTO DO SISTEMA*

## **10 Manual de usuario**

*DEBE INCLUIR REQUISITOS MÍNIMOS, MANUAL DE INSTALACIÓN E DE UTILIZACIÓN*

## **11 Principais contribucións**

*DEBERANSE DESTACAR AS ACHEGAS IMPORTANTES DO TRABALLO REALIZADO, TENDO EN  
CONTA OS OBXECTIVOS FIXADOS*

## 12 Conclusións

*INCLUIRANSE TODAS AS CONCLUSIÓN DE TIPO TÉCNICO E PERSOAL*

## **13 Vías de traballo futuro**

*PRESENTARANSE POSIBLE AMPLIACIÓNS E TRABALLOS RELACIONADOS POR FACER*

## 14 Referencias

DEBERANSE CITAR TODAS AS FONTES DE INFORMACIÓN EMPREGADAS PARA A REALIZACIÓN DO TRABALLO. DEBERASE EMPREGAR UN ESTILO UNIFORME PARA TODAS ELAS E APORTARASE, EN CADA CASO:

1. AUTOR/A/ES/AS.
2. TÍTULO DO ARTIGO, LIBRO, MONOGRAFÍA,...
3. EDITORIAL OU NOME DA REVISTA.
4. NÚMERO DA REVISTA, VOLUME E PÁXINAS (SÓ PARA REVISTAS).
5. ANO DE PUBLICACIÓN.
6. DIRECCIÓN E DATA DE CONSULTA (SÓ PARA URL).

RECOMÉNDASE EMPREGAR PARA REFERENCIAS DE ARTIGOS, REVISTAS, E OUTRAS FONTES DE REFERENCIA, O FORMATO APA, ISO 690, IEEE OU SIMILARES, E UNIFORMES AO LONGO DE TODA A SECCIÓN.

1. ESTADÍSTICAS WHATSAPP - [HTTPS://WWW.WHATSAPP.COM/ABOUT/](https://www.whatsapp.com/about/)

X. “¿QUÉ ES UNA API?”, RED HAT, 31/10/2017 –  
[HTTPS://WWW.REDHAT.COM/ES/TOPICS/API/WHAT-ARE-APPLICATION-PROGRAMMING-INTERFACES](https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces) [ÚLTIMA VISITA: 20/10/2021]

X. “WEBSOCKETS VS LONG POLLING”, KIERAN KILBRIDE-SINGH, 27/10/2021 -  
[HTTPS://ABLY.COM/BLOG/WEBSOCKETS-VS-LONG-POLLING](https://ably.com/blog/websockets-vs-long-polling) [ÚLTIMA VISITA: 30/10/2021]

X. “JSON WEB TOKEN”, ISSN: 2070-1721  
[HTTPS://DATATRACKER.IETF.ORG/DOC/HTML/RFC7519](https://datatracker.ietf.org/doc/html/rfc7519) , [ÚLTIMA VISITA: 20/12/2021]

X. [HTTP://EXPRESSJS.COM/](http://expressjs.com/) [ÚLTIMA VISITA: 1/11/2021]

X. “MAIN GOAL OF WEBSOCKET PROTOCOL”, IETF  
[HTTPS://TRAC.IETF.ORG/TRAC/HYBI/WIKI/FAQ](https://trac.ietf.org/trac/hybi/wiki/faq)

## **15 Anexos**

*INCLUIRANSE OUTROS ELEMENTOS DE INTERESE NO TFG QUE SE CONSIDEREN NECESARIOS  
PARA A MELLOR COMPRENSIÓN DO MESMO.*

**Socket.IO**

**WebSockets**

**HTTP Long-Polling**

**RUP (Rational Unified Process)**

## NOTAS

DÉBESE TER EN CONTA QUE, DEPENDENDO DAS DIFERENTES CARACTERÍSTICAS DE CADA TFG PARTICULAR, PODEN NON SER PROCEDENTES TODOS OS APARTADOS NUMERADOS DO 5.11 AO 5.15. NO CASO DE NON EXISTIR ALGÚN DELES, DEBERASE XUSTIFICAR DEBIDAMENTE A SÚA AUSENCIA NA DOCUMENTACIÓN. NESTE CASO, O TRIBUNAL DE AVALIACIÓN VALORARÁ ESTE FEITO DO XEITO QUE ESTIME MÁIS OPORTUNO.

A DOCUMENTACIÓN TERÁ UNHA EXTENSIÓN MÁXIMA RECOMENDADA DE 50 PÁXINAS, SEN INCLUÍR OS PUNTOS SEGUINTE:

DO 5.1 (PORTADA SEGUNDO MODELO OFICIAL, COS DATOS DE IDENTIFICACIÓN DO TRABALLO )

AO 5.6. (ÍNDICE DE TÁBOAS )

, E 5.22. (ANEXOS )