1. Introducción

El presente proyecto ha sido realizado para la demostración de un sistema (en este caso red social) a través de una aplicacion movil. Este sistema funciona con la interaccion meidante un navegador web desde una computadora o celular. El sistema también presenta una conexión con una base de datos postgreSQL para el almacenamiento de datos de los usuarios que se registran.

1. Antecedentes y jutificación

Se ha investigado sobre los sistemas en servidores privados utilizando la misma funcionalidad y alcance. Las bases de datos son imprescindibles para el correcto funcionamiento, es por ello que se ha decidido realizar una conexión con una base de datos para el almacenamiento de datos.

1. El problema
   1. Situación problemática

La falta de optimización en las redes sociales actuales ha bajado mucho su uso en áreas no empresariales.

* 1. Formulación del problema

Al no utilizarse las redes sociales como una alternativa a las comunicaciones en tiempo real no son utilizados a su máximo potencial en nuestro día a día.

1. Objetivos
   1. Objetivo general

Desarrollar una red social aplicada en los correos electrónicos.

* 1. Objetivos específicos
* Implementar una conexión entre el servidor y el servidor Pusher para las notificaciones en php.
* Crear una base de datos para el almacenamiento de la informacion.
* Implementar una aplicación php para la lógica de negocio y validacion de datos.

1. Metodología
   1. Captura de requisitos
      1. Encontrar actores y casos de uso

Actores

User

Administrator

Casos de uso

Gestionar perfil

Gestionar contactos

Buscar amigos

Buscar mensajes

Enviar solicitud de amistad

Gestionar mensaje y notifiacion

Reportes

Estadisticas

* + 1. Detallar casos de uso

CU1 Gestionar perfil

|  |  |
| --- | --- |
| **Caso de uso** | CU1 Gestionar perfil |
| **Propósito** | Registrar, modificar, eliminar y cambiar contraseña en perfiles de usuarios de minifacebook. |
| **Resumen** | El software nos permitirá el registro y modificación de la información de los usuarios, así además la eliminación de perfiles. |
| **Actores** | Usuario |
| **Actor iniciador** | Usuario |
| **Pre-condición** | **Modificar**   * Iniciar sesion   **Eliminar**   * Iniciar sesion   **Cambiar contraseña**   * Iniciar sesion |
| **Proceso** | **Registrar**:   1. El usuario ingresará al fomrulario de registro y rellenará los siguientes parámetros: nombres, apellidos, sexo, contraseña 2. El sistema valida los datos 3. El sistema asigna un identificador y una fecha de creación 4. El sistema inserta la información en la base de datos 5. El sistema responde redireccionando al usuario a la pagina de inicio   **Modificar**:   1. El usuario selecciona la opción de editar perfil 2. El ususario edita el formulario de información personal con los siguientes parámetros: nombres, apellidos, fecha de nacimiento, sexo, celular, correo 3. El sistema valida los datos 4. El sistema actualiza la información en la base de datos 5. El sistema responde con un mensaje de confirmación   **Eliminar**   1. El usuario selecciona la opción de editar cuenta 2. El usuario selecciona la opción de eliminar cuenta 3. El sistema cambia todas las referencias al usuario a una cuenta por defecto 4. El sistema elimina el usuario de la base de datos 5. El sistema responde con un mensaje de confirmación redireccionándolo a la página de registro   **Cambiar contraseña**   1. El usuario selecciona la opcion de editar perfil 2. El usuario rellena el formulario de cambio de contraseña 3. El sistema valida los datos 4. El sistema actualiza la información de la base de datos 5. El sistema responde con un mensaje de confirmación |
| **Post-condición** | Ninguna |
| **Excepción** | **Registrar**  2. Datos incorrectos  **Modificar**  3. Datos incorrectos  **Cambiar contraseña**  3. Datos incorrectos |

CU2 Gestionar contactos

|  |  |
| --- | --- |
| **Caso de uso** | CU2 Gestionar contactos |
| **Propósito** | Listar, registrar y eliminar los contactos de un usuario. |
| **Resumen** | El sistema nos permitirá la gestión de nuestros contactos (amigos) que nos envíen solicitudes, nos permitirá realizar acciones como registrarlos, eliminarlos y listarlos. |
| **Actores** | Usuario |
| **Actor iniciador** | Usuario |
| **Pre-condición** | **Registrar**:   * Iniciar sesion * Recibir una solicitud de amistad de otro usuario.   **Listar**   * Iniciar sesión   **Eliminar**   * Ver perfil de usuario |
| **Proceso** | **Registrar**:   1. El usuario abre la lista de solicitudes pendientes. 2. El usuario aceptará a un usuario de la lista obtenida 3. El sistema valida los datos y registra al nuevo contacto a ambos usuarios. 4. El sistema responde con un mensaje de confirmación   **Listar**   1. El usuario visualizará su lista de contactos en todo momento que su sesión esté abierta   **Eliminar**   1. El usuario selecciona la opción de eliminar en el perfil del contacto 2. El sistema valida los datos y elimina al contacto 3. El sistema responde con un mensaje de confirmación |
| **Post-condición** | Ninguno |
| **Excepción** | **Registrar**  3. Datos incorrectos  **Eliminar**  2. Datos incorrectos |

CU3 Buscar amigo

|  |  |
| --- | --- |
| **Caso de uso** | CU3 Buscar amigo |
| **Propósito** | Listar usuarios que coincidan con la búsqueda. |
| **Resumen** | El sistema nos permitirá encontrar a los amigos que deseamos contactar, mostrando la información básica del usuario. |
| **Actores** | Usuario |
| **Actor iniciador** | Usuario |
| **Pre-condición** | Iniciar sesion |
| **Proceso** | **Buscar**   1. El usuario ingresa la información a buscar en el campo de texto de búsueda 2. El sistema responde con la lista de usuarios coincidentes |
| **Post-condición** | Ninguno |
| **Excepción** | Ninguno |

CU4 Buscar mensaje

|  |  |
| --- | --- |
| **Caso de uso** | CU4 Buscar mensaje |
| **Propósito** | Listar mensajes que coincidan con la búsqueda |
| **Resumen** | El software nos permitirá encontrar los mensajes que deseamos encontrar, mostrando los mensajes completos |
| **Actores** | Usuario |
| **Actor iniciador** | Usuario |
| **Pre-condición** | Tener al otro usuario del chat registrado como contacto |
| **Proceso** | **Listar**   1. El usuario selecciona la opcion de buscar mensaje en el chat 2. El sistema responde con la lista de mensajes coincidentes |
| **Post-condición** | Ninguno |
| **Excepción** | Ninguno |

CU5 Enviar solicitud de amistad

|  |  |
| --- | --- |
| **Caso de uso** | CU5 Enviar solicitud de amistad |
| **Propósito** | Contactar con otro usuario |
| **Resumen** | El usuario solicita contactar con otro usuario |
| **Actores** | Usuario |
| **Actor iniciador** | Usuario |
| **Pre-condición** | Ver perfil de usuario |
| **Proceso** | 1. El usuario selecciona la opcion de enviar solicitud de amistad en el perfil de usuario objetivo 2. El sistema responde con un mensaje de confirmación |
| **Post-condición** | Ninguno |
| **Excepción** | Ninguno |

CU6 Gestionar mensaje

|  |  |
| --- | --- |
| **Caso de uso** | CU6 Gestionar mensaje |
| **Propósito** | Registrar, modificar y eliminar un mensaje escrito por el usuario. |
| **Resumen** | El usuario envía un mensaje a uno de sus contactos y luego podrá editarlo o eliminarlo a conveniencia |
| **Actores** | Usuario |
| **Actor iniciador** | Usuario |
| **Pre-condición** | Tener al otro usuario del chat registrado como contacto |
| **Proceso** | **Registrar**   1. El usuario crea un mensaje y lo envia a traves del chat 2. El sistema responde actualizando los mensajes del chat   **Modificar**   1. El usuario selecciona la opcion de editar de un mensaje ya enviado 2. El usuario escribe el nuevo contenido del mensaje en un campo de texto 3. El sistema valida los datos 4. El sistema responde actualizando los mensajes del chat   **Eliminar**   1. El usuario selecciona la opcion de eliminar de un mensaje ya enviado 2. El sistema valida los datos 3. El sistema responde actualizando los mensajes |
| **Post-condición** | Ninguno |
| **Excepción** | Ninguno |

CU7 Generar reportes

|  |  |
| --- | --- |
| **Caso de uso** | CU7 Generar reportes |
| **Propósito** | Ver información interna del sistema |
| **Resumen** | El administrador realiza una petición para ver los datos |
| **Actores** | Administrador |
| **Actor iniciador** | Administrador |
| **Pre-condición** | Iniciar sesion |
| **Proceso** | 1. El administrador selecciona la opción para ver el reporte generado por el sistema 2. El sistema responde con el reporte |
| **Post-condición** | Ninguno |
| **Excepción** | Ninguno |

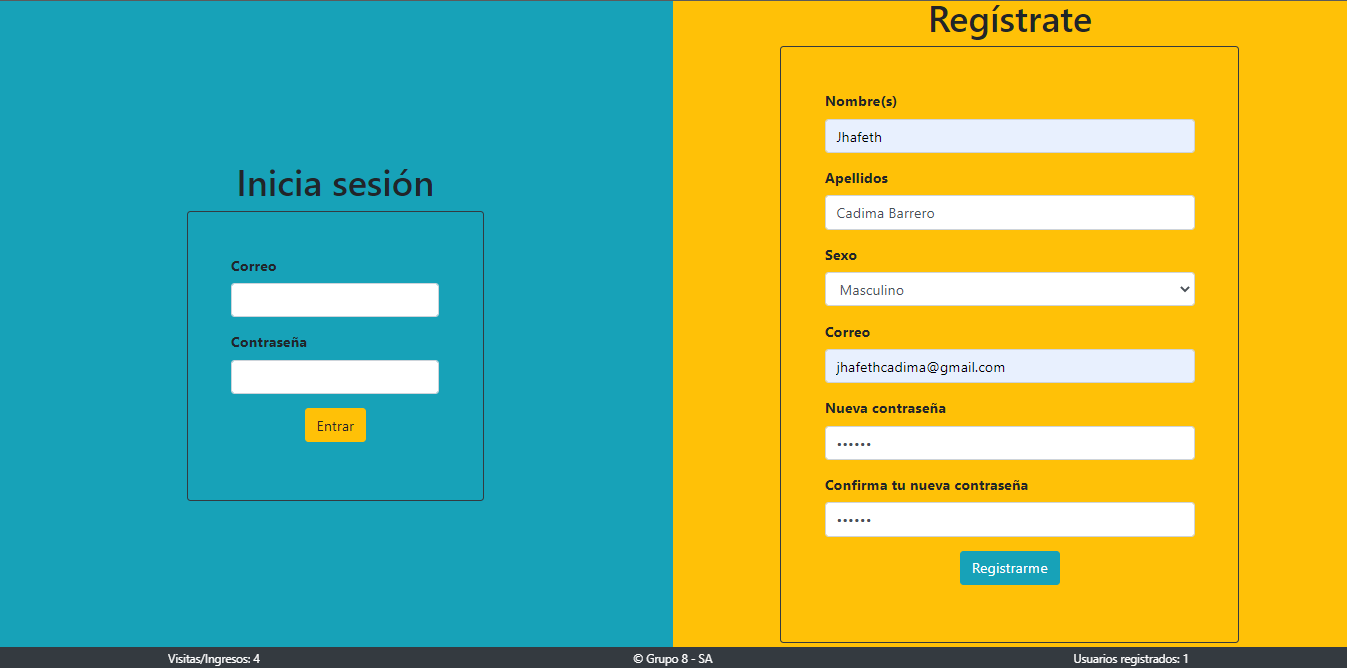
CU8 Ver estadisticas

|  |  |
| --- | --- |
| **Caso de uso** | CU8 Ver estadísticas |
| **Propósito** | Ver información interna del sistema |
| **Resumen** | El administrador realiza una petición para ver las estadísticas |
| **Actores** | Administrador |
| **Actor iniciador** | Administrador |
| **Pre-condición** | Iniciar sesion |
| **Proceso** | 1. El administrador selecciona la opcion para poder ver la estadística generada por el sistema 2. El sistema responde la petición con la estadística |
| **Post-condición** | Ninguno |
| **Excepción** | Ninguno |

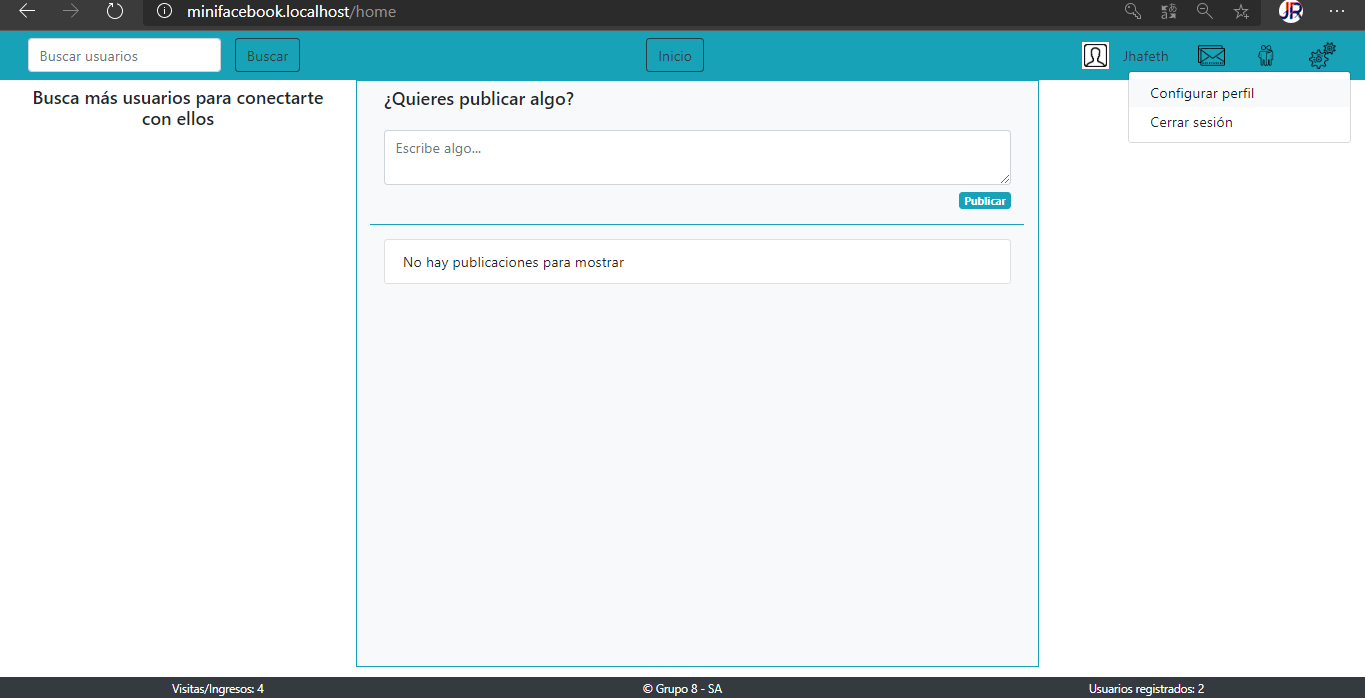
* + 1. Prototipar interfaz de usuario

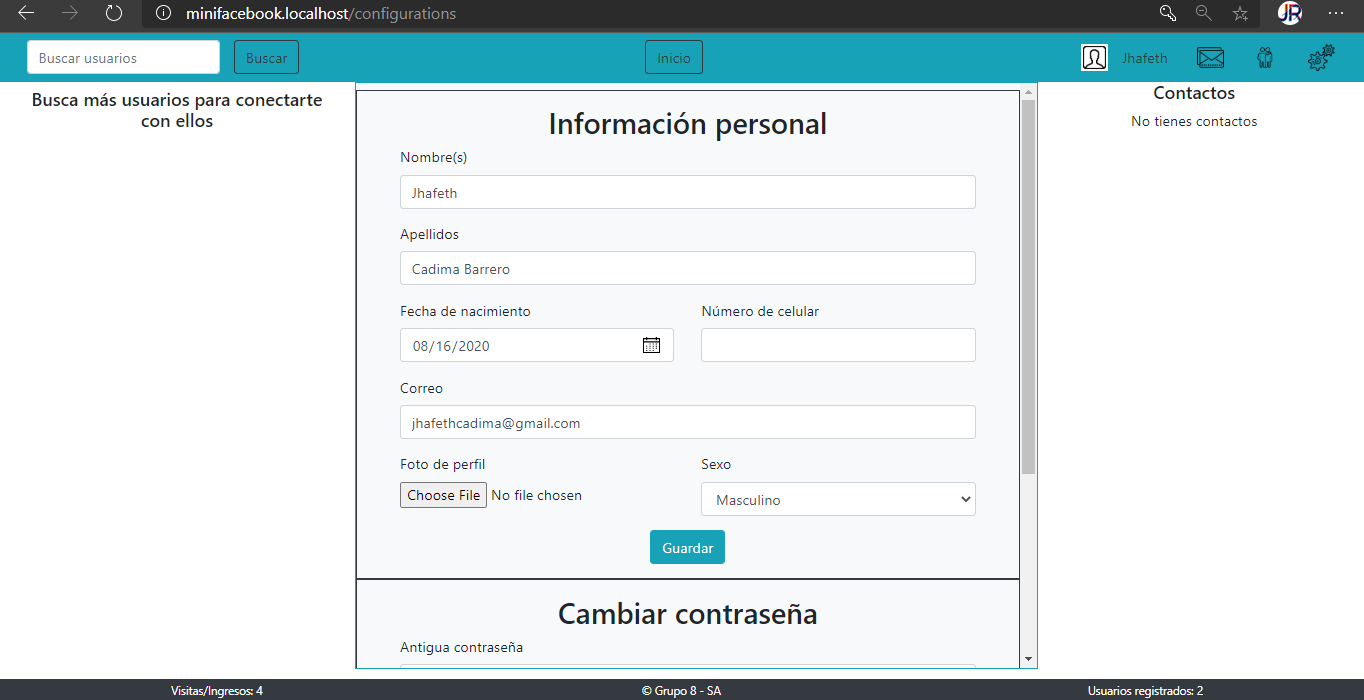
CU1 Gestionar perfil

* Crear perfil

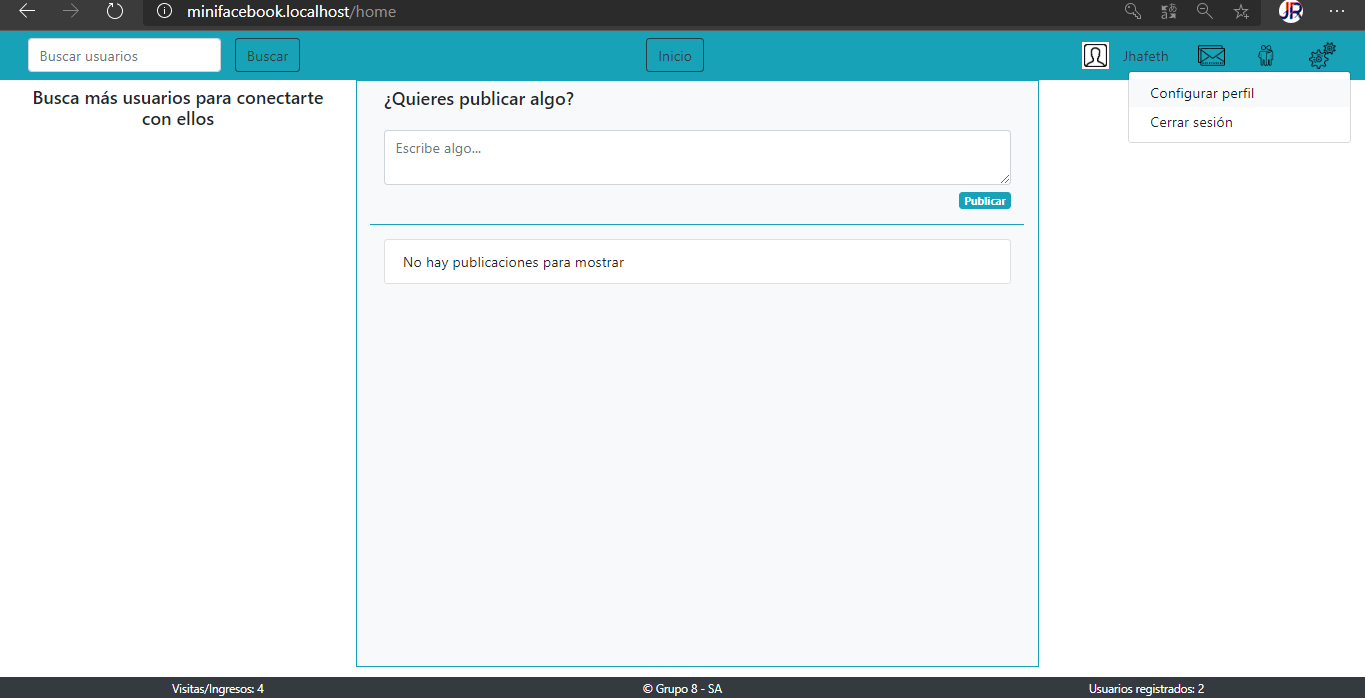


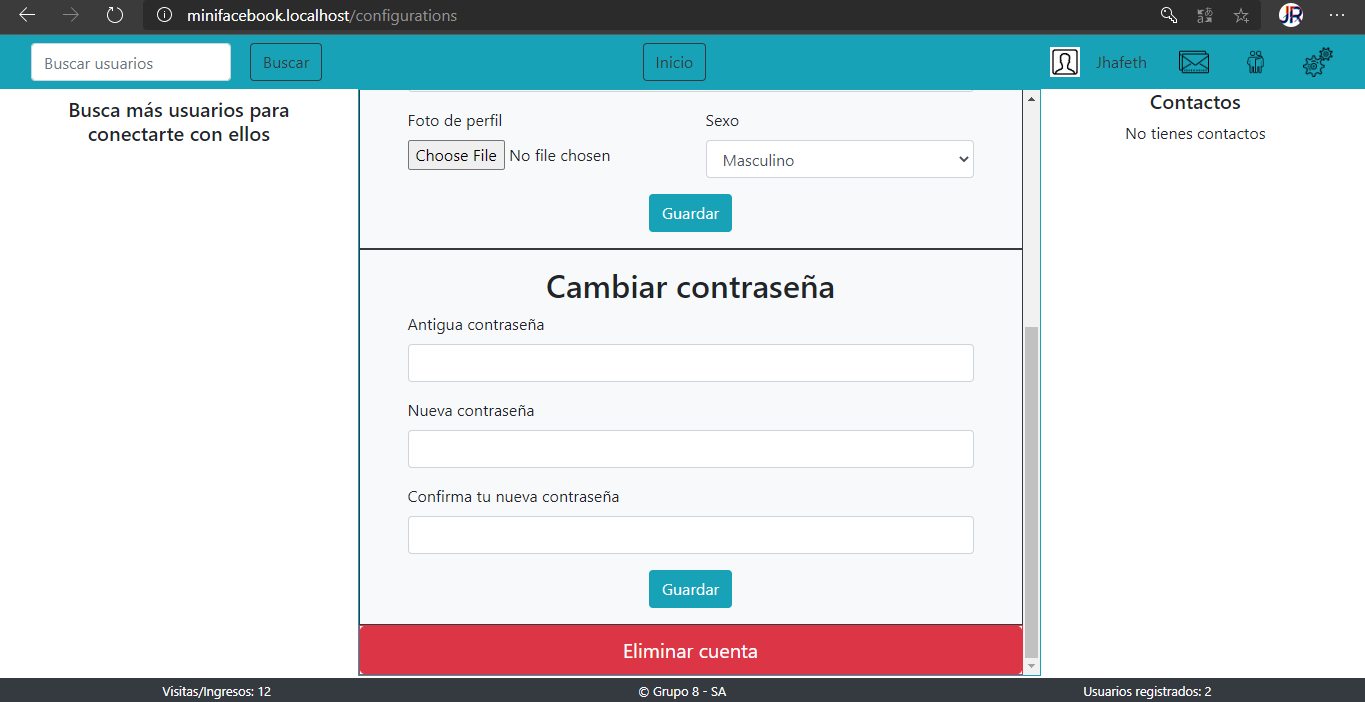
* Editar perfil

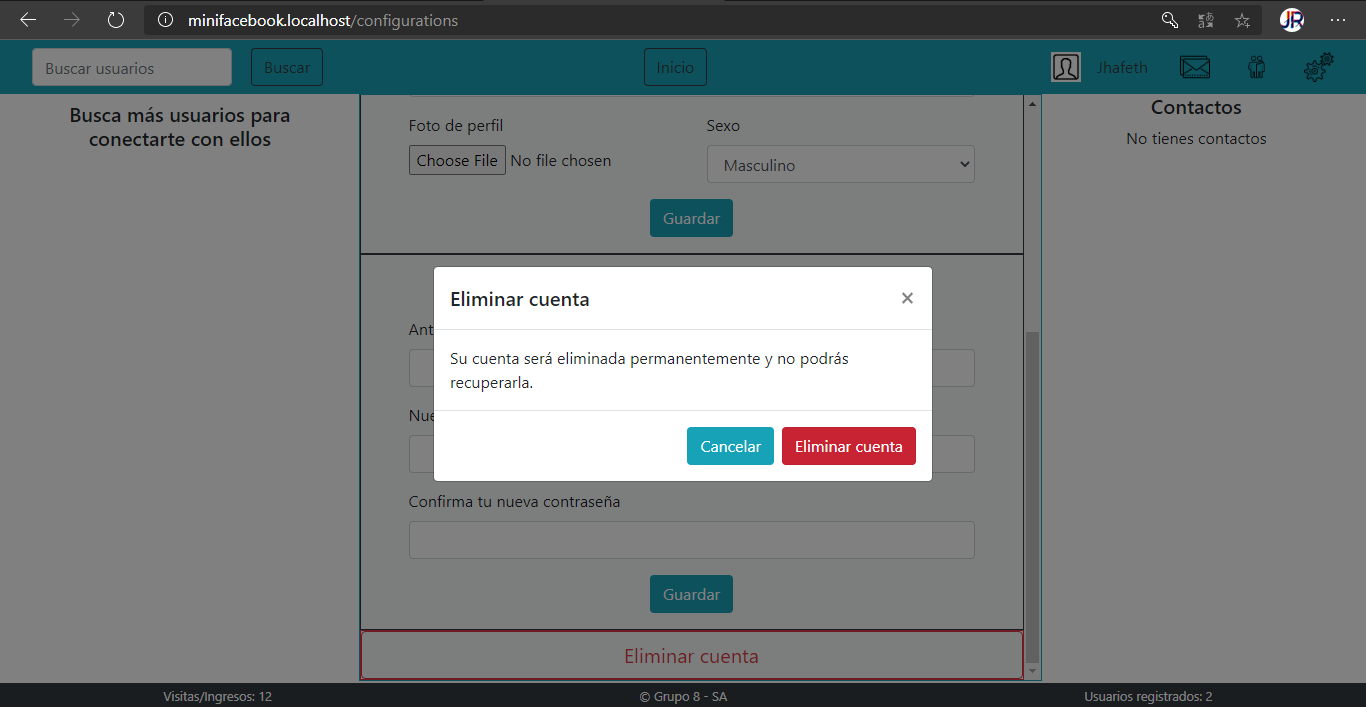




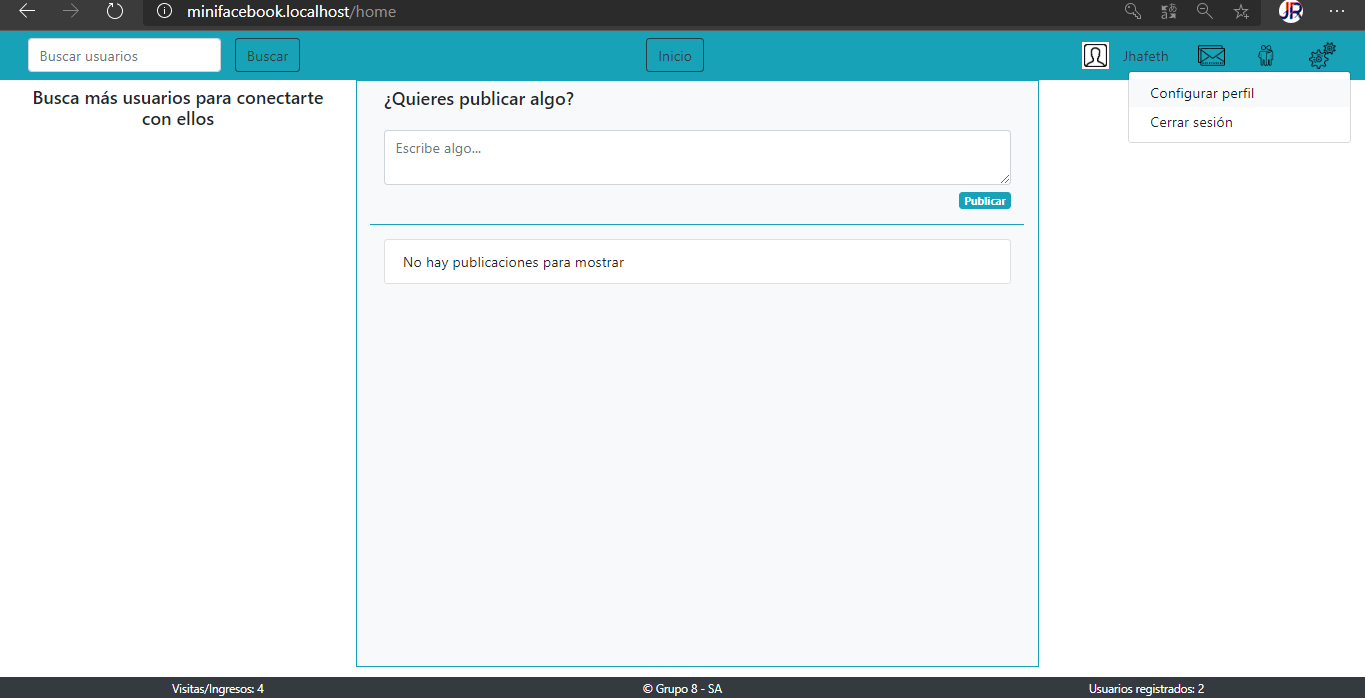
* Eliminar perfil

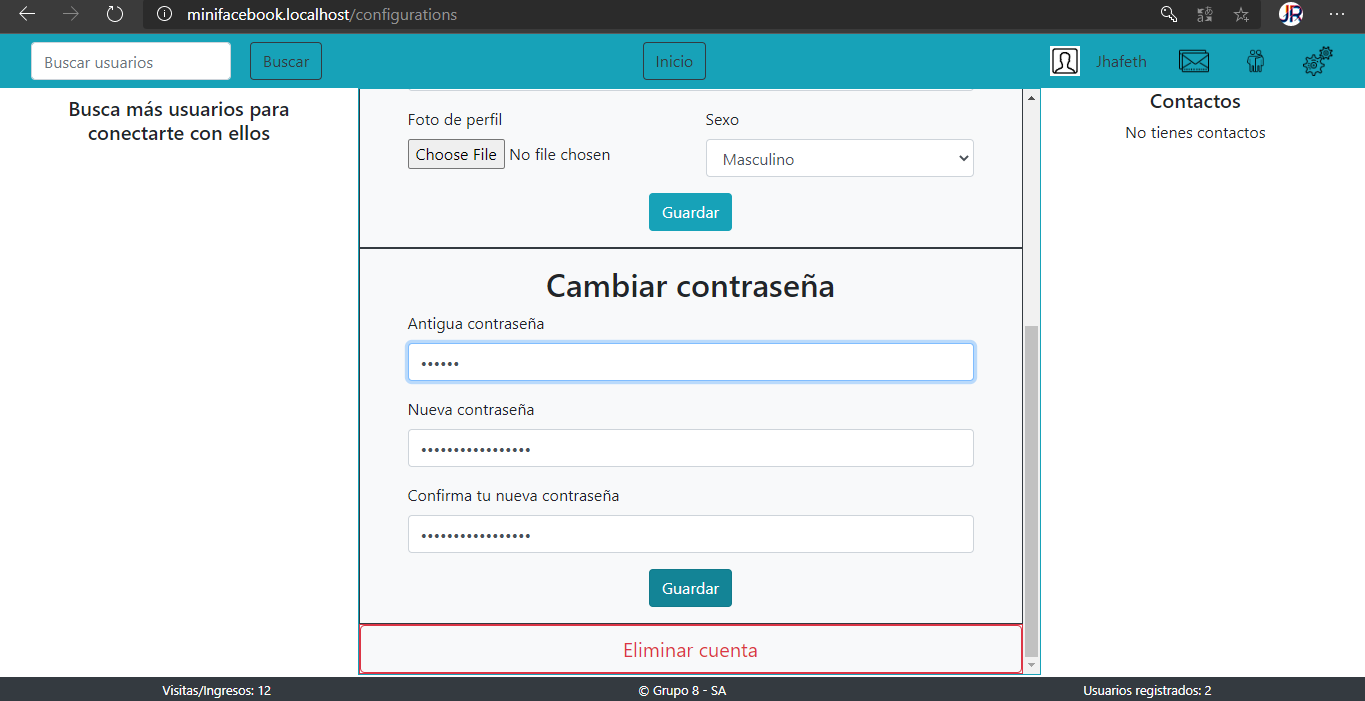






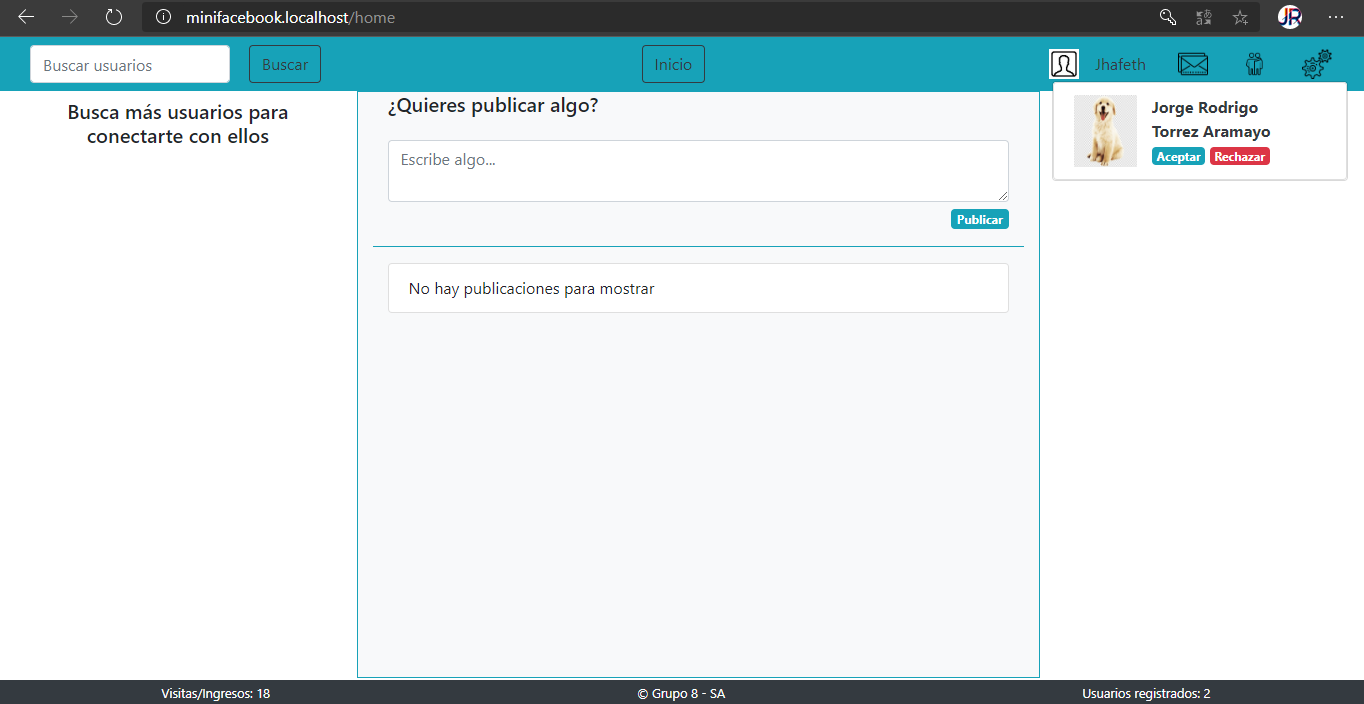
* Cambiar contraseña



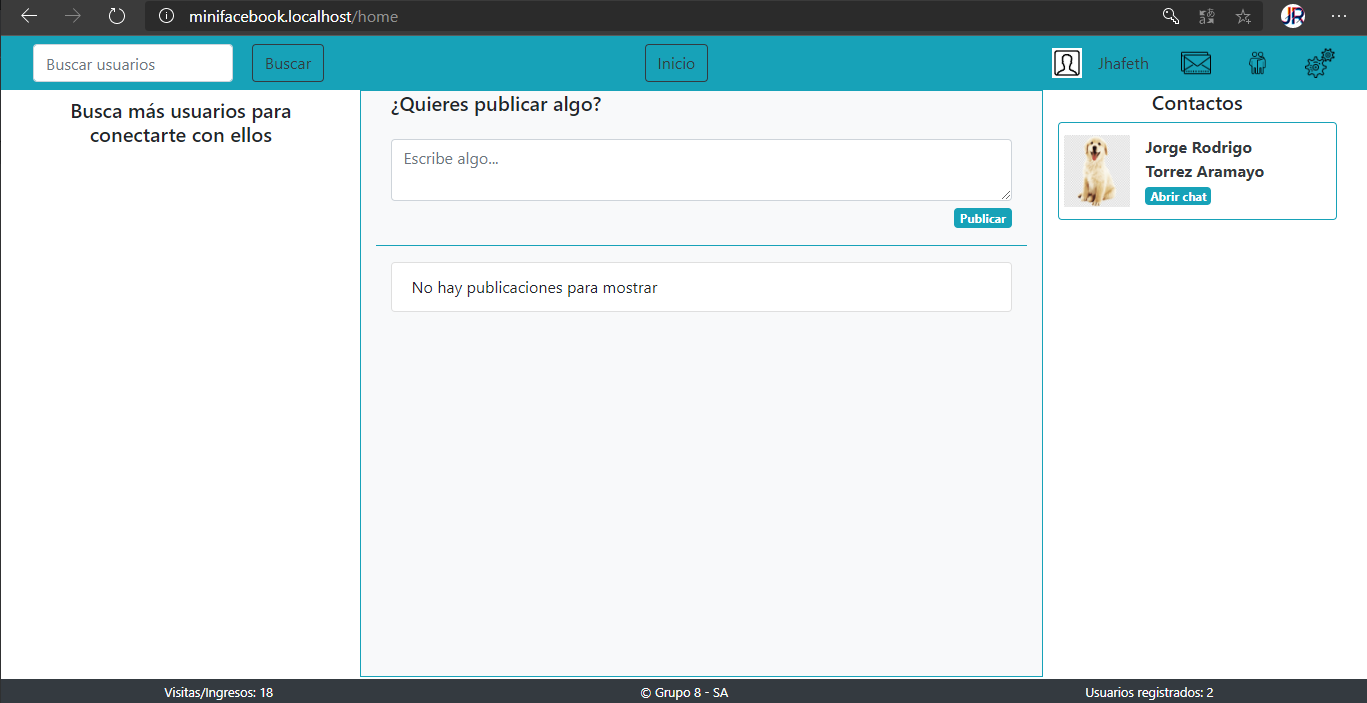


CU2 Gestionar contactos

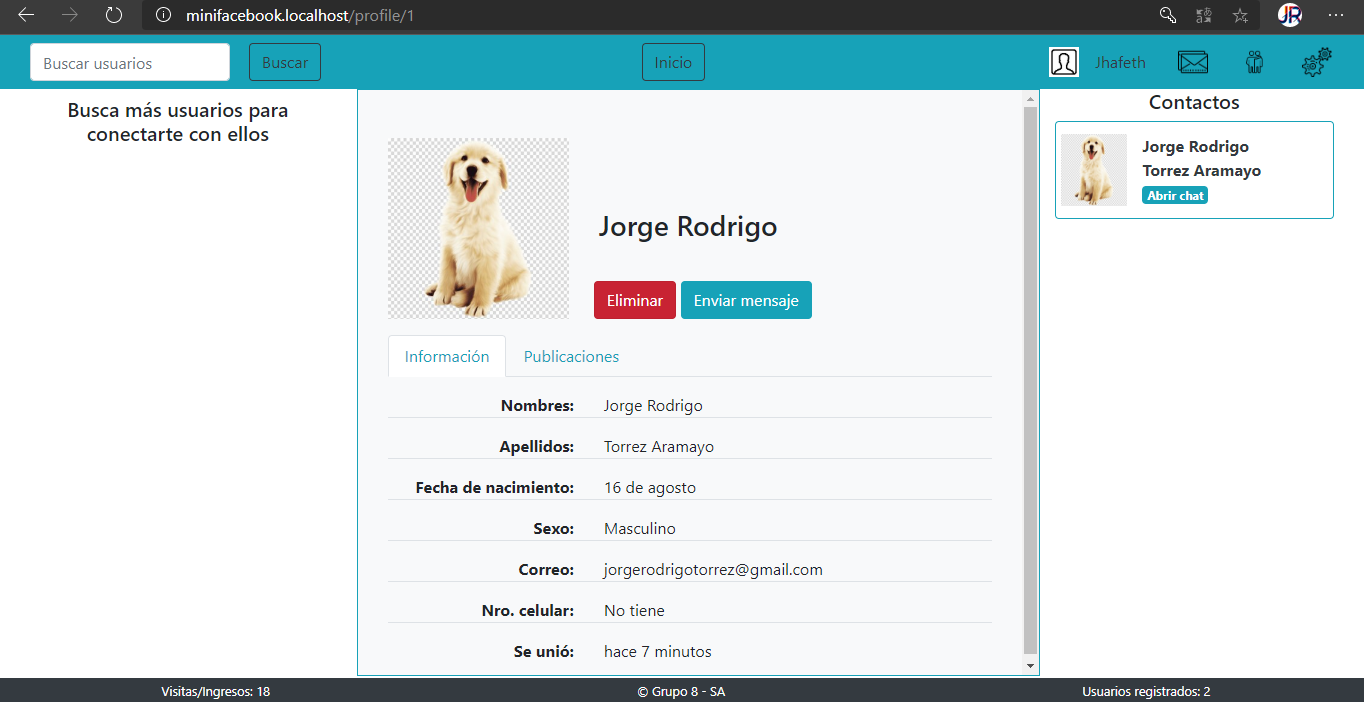
* Registrar

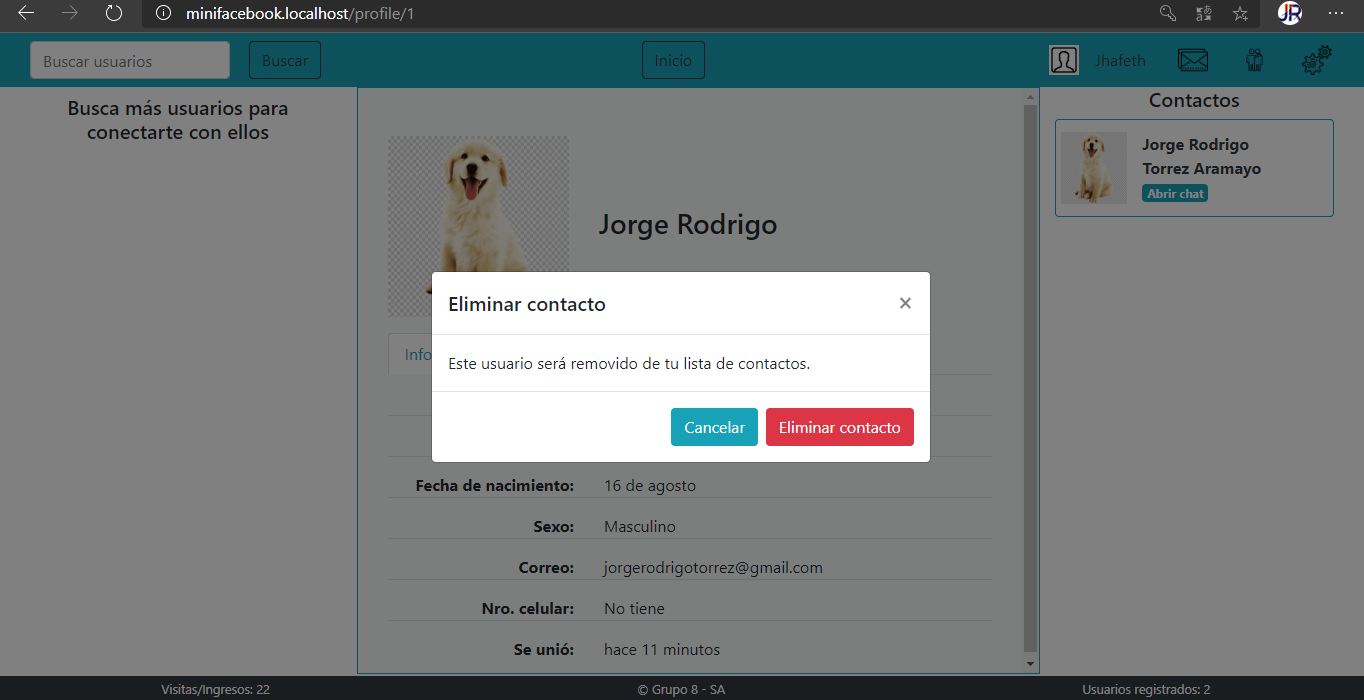


* Listar

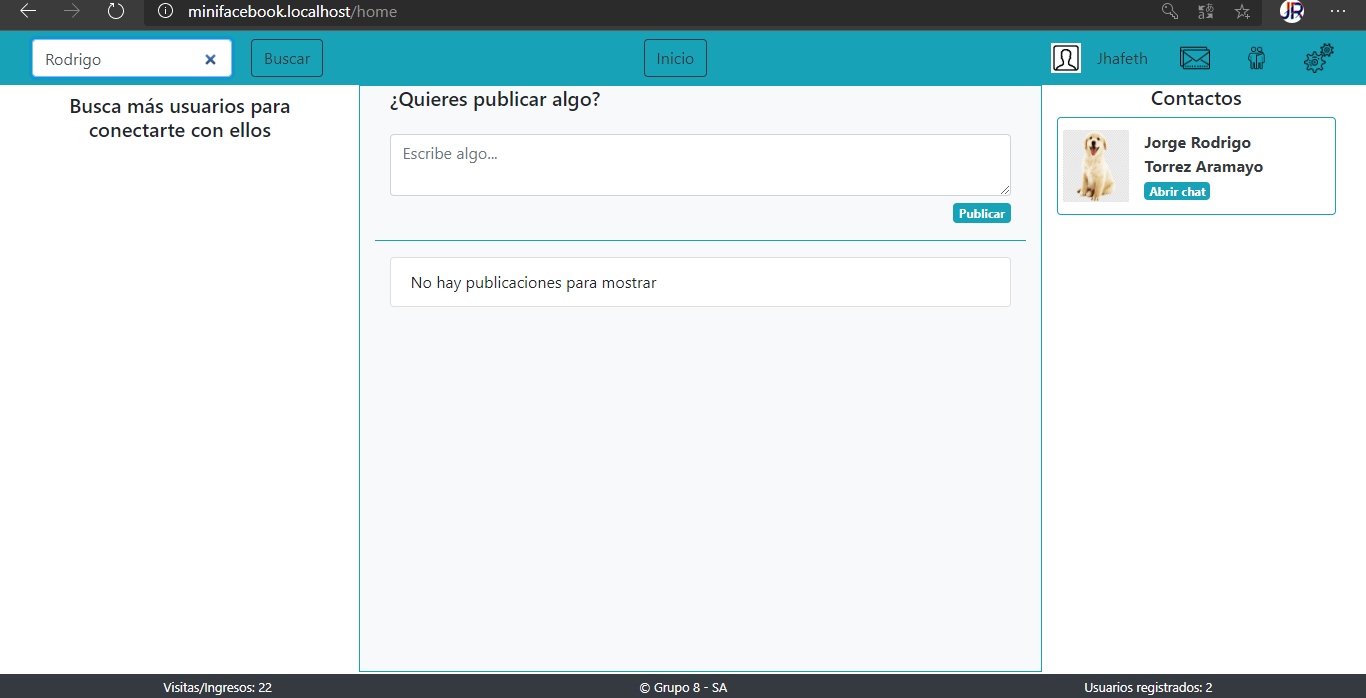


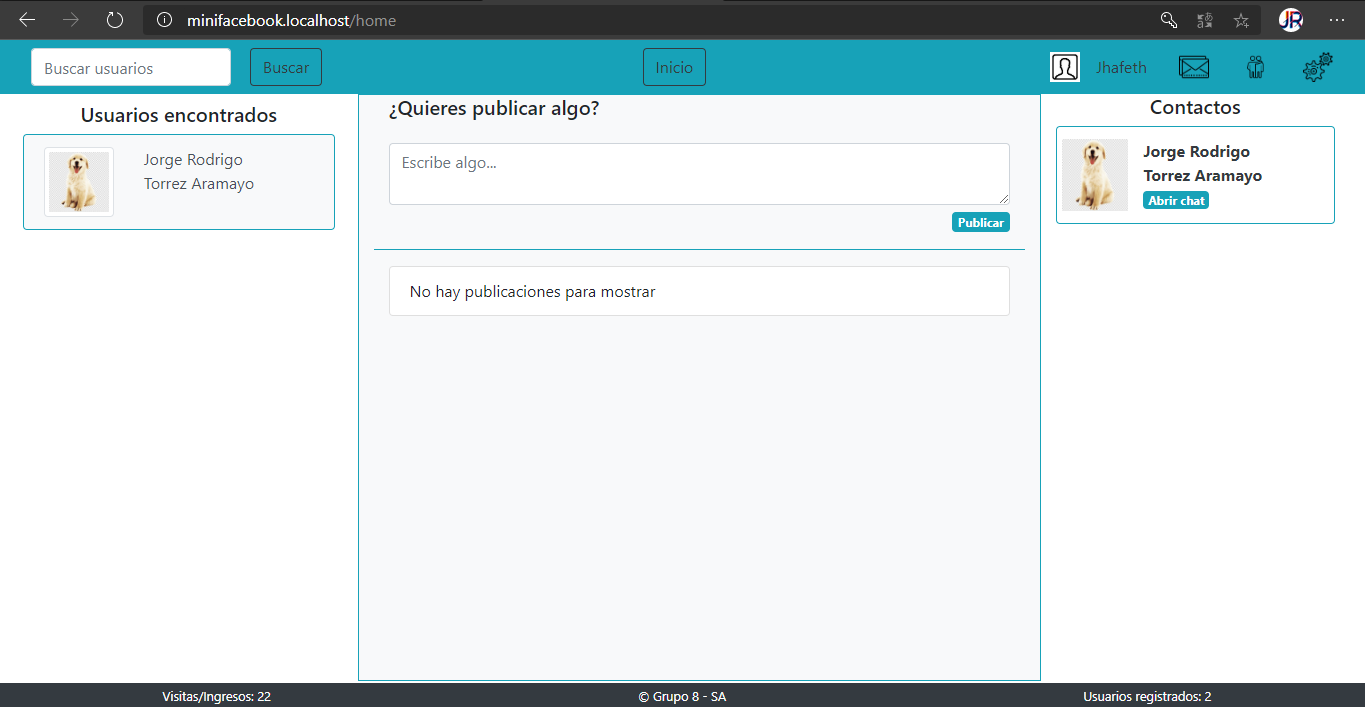
* Eliminar



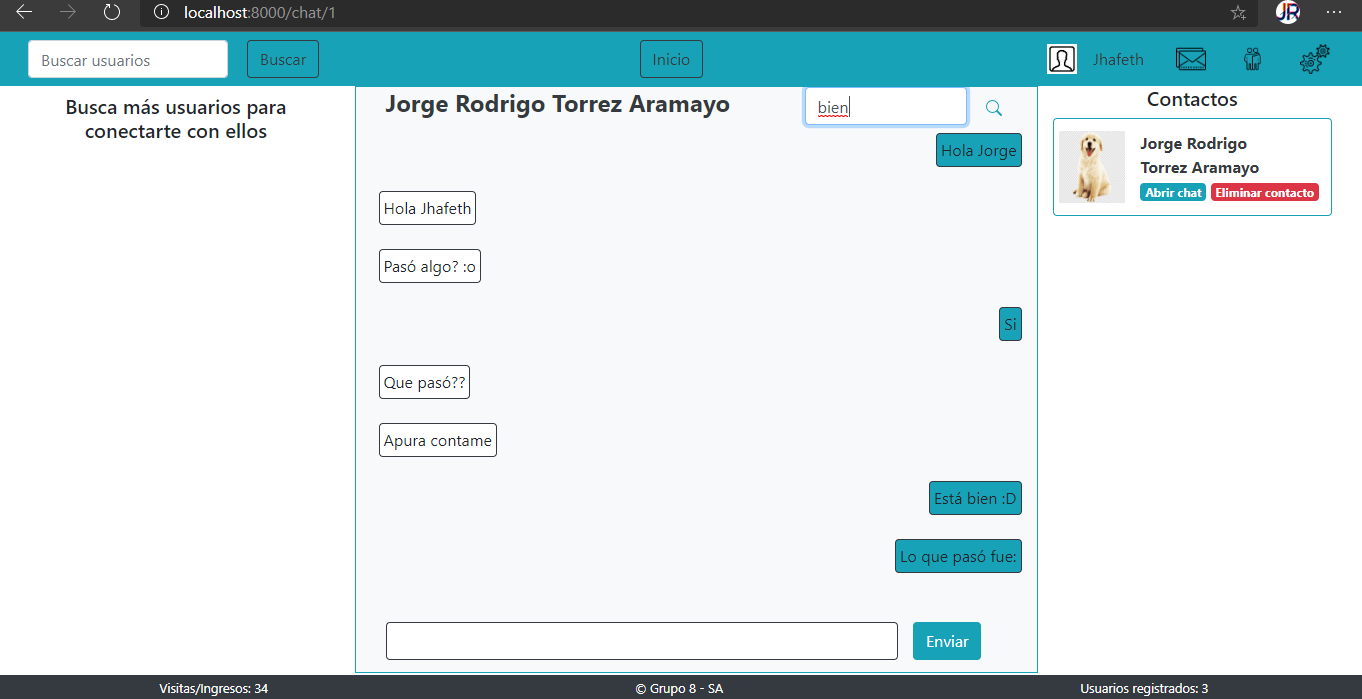


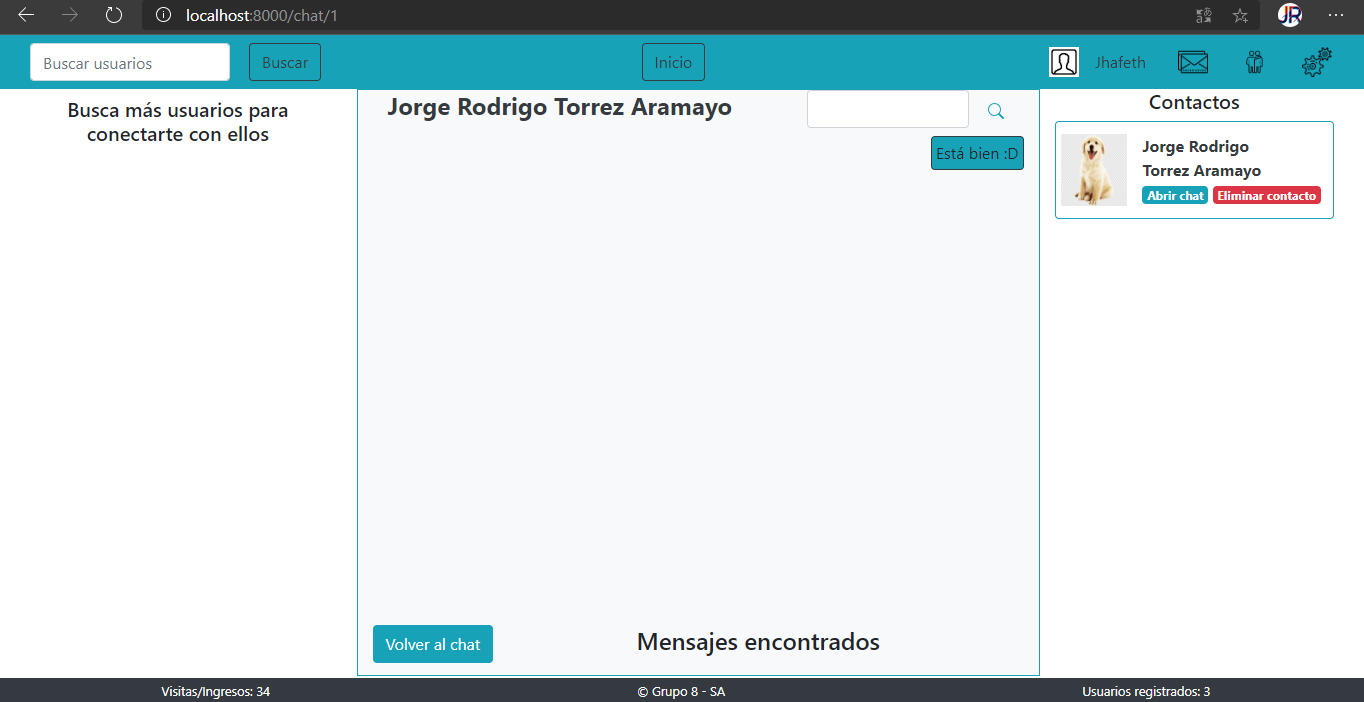
CU3 Buscar amigos



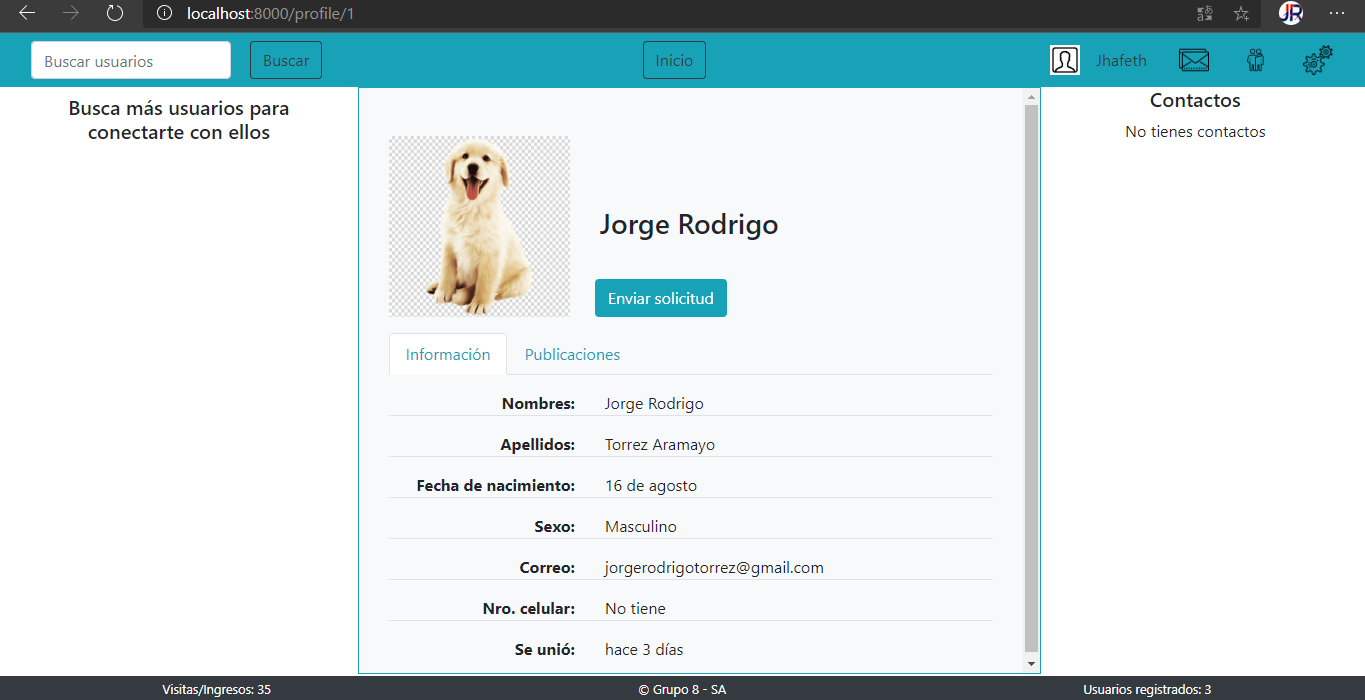


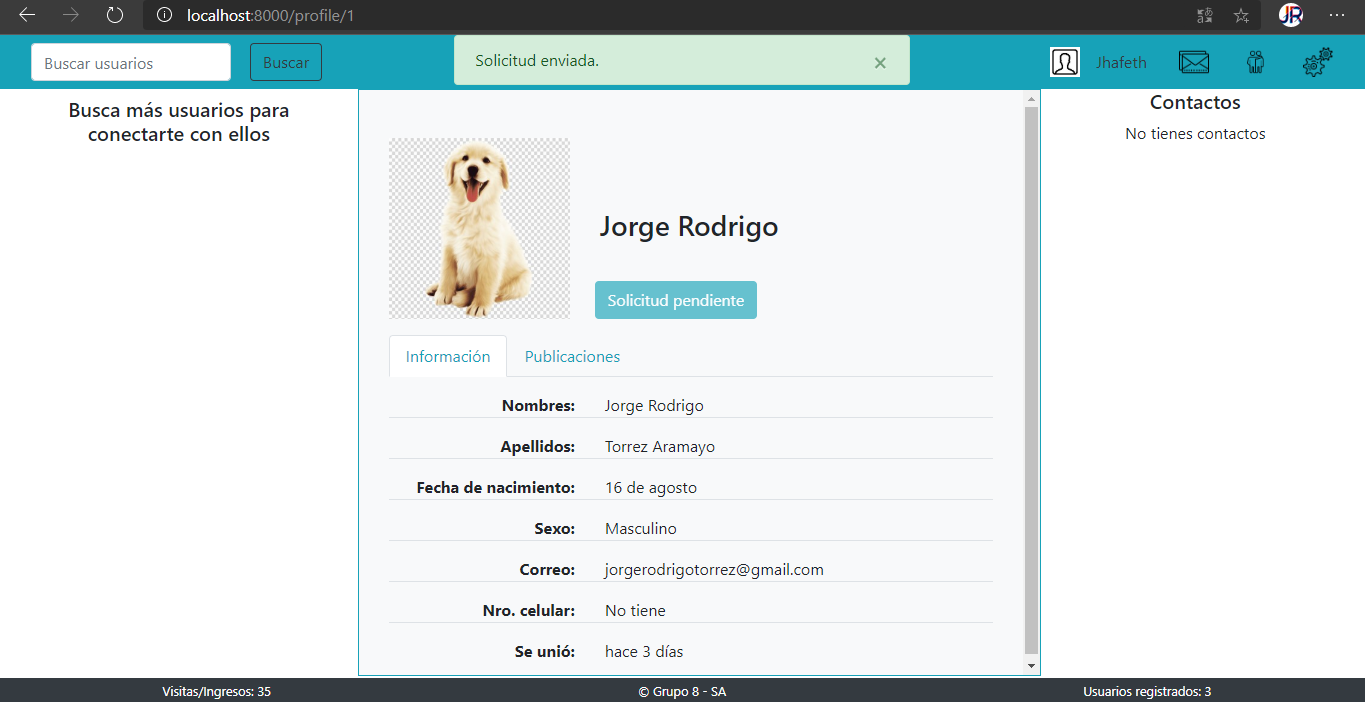
CU4 Buscar mensajes





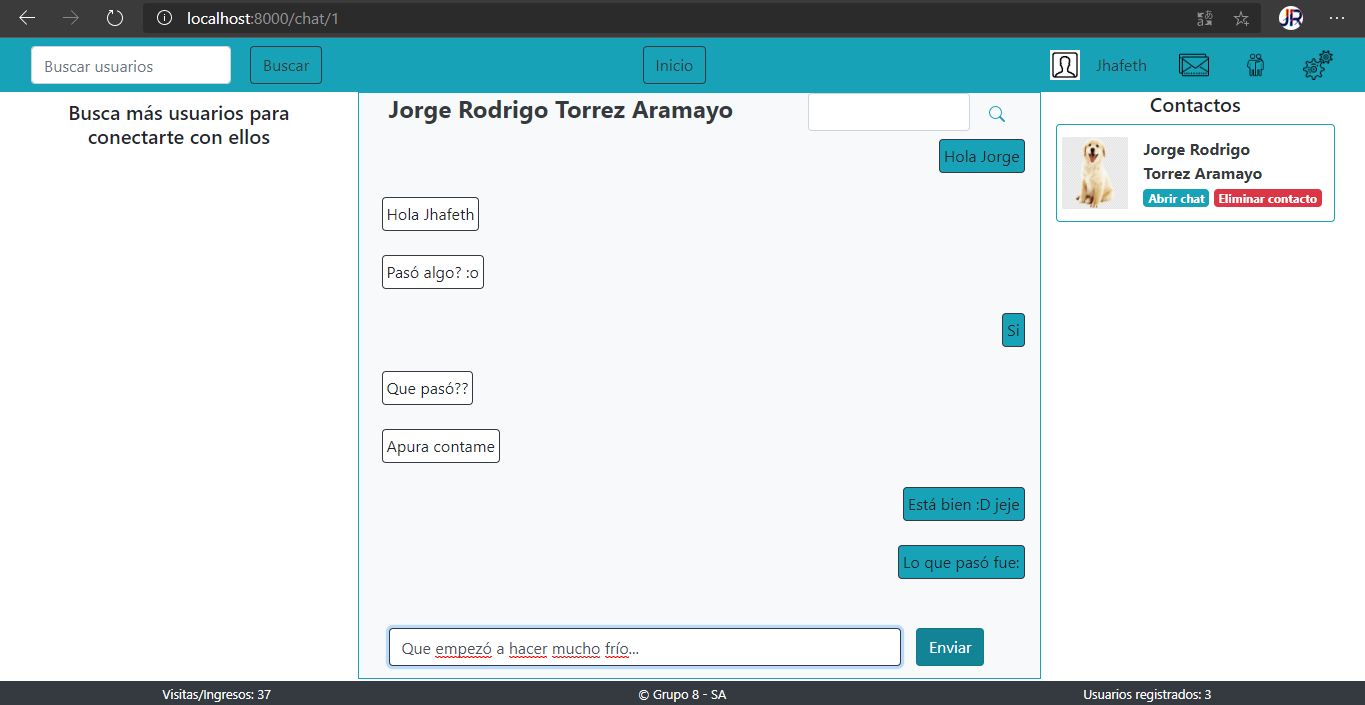
CU5 Enviar solicitud de amistad



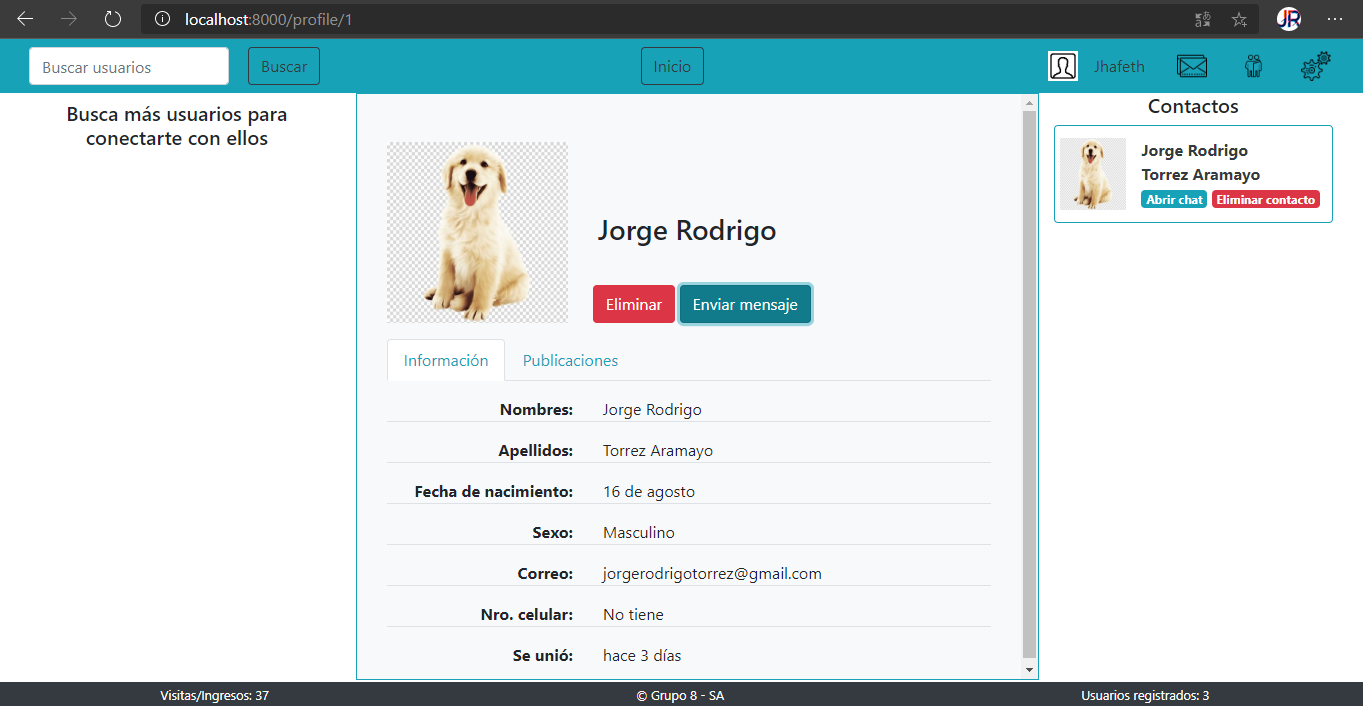


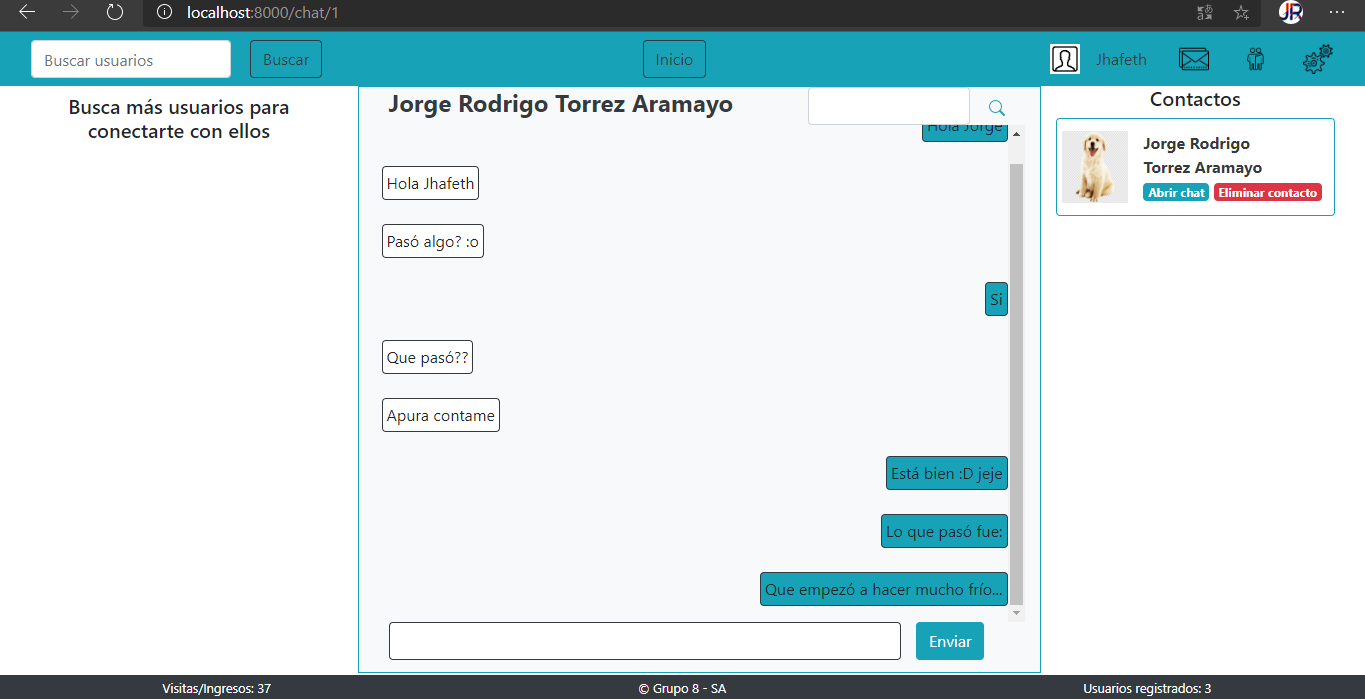
CU6 Gestionar mensajes y notificaciones

* Crear mensaje

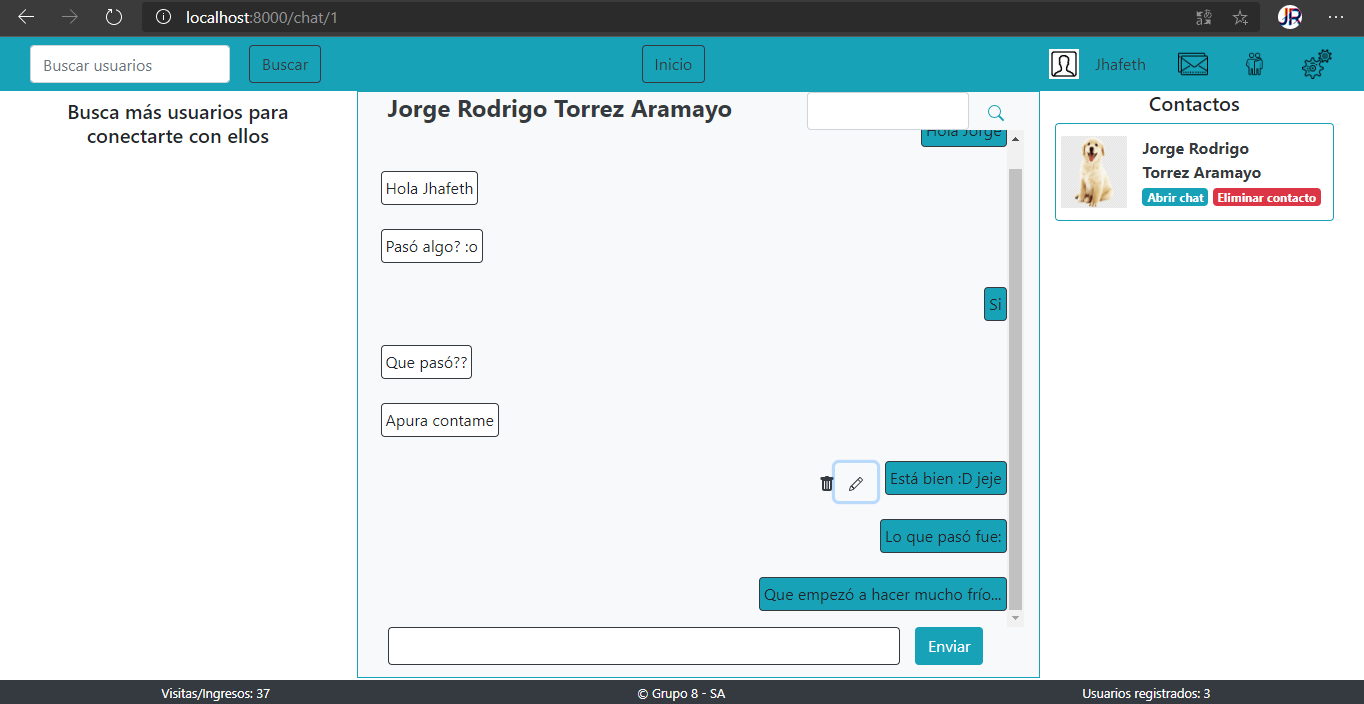


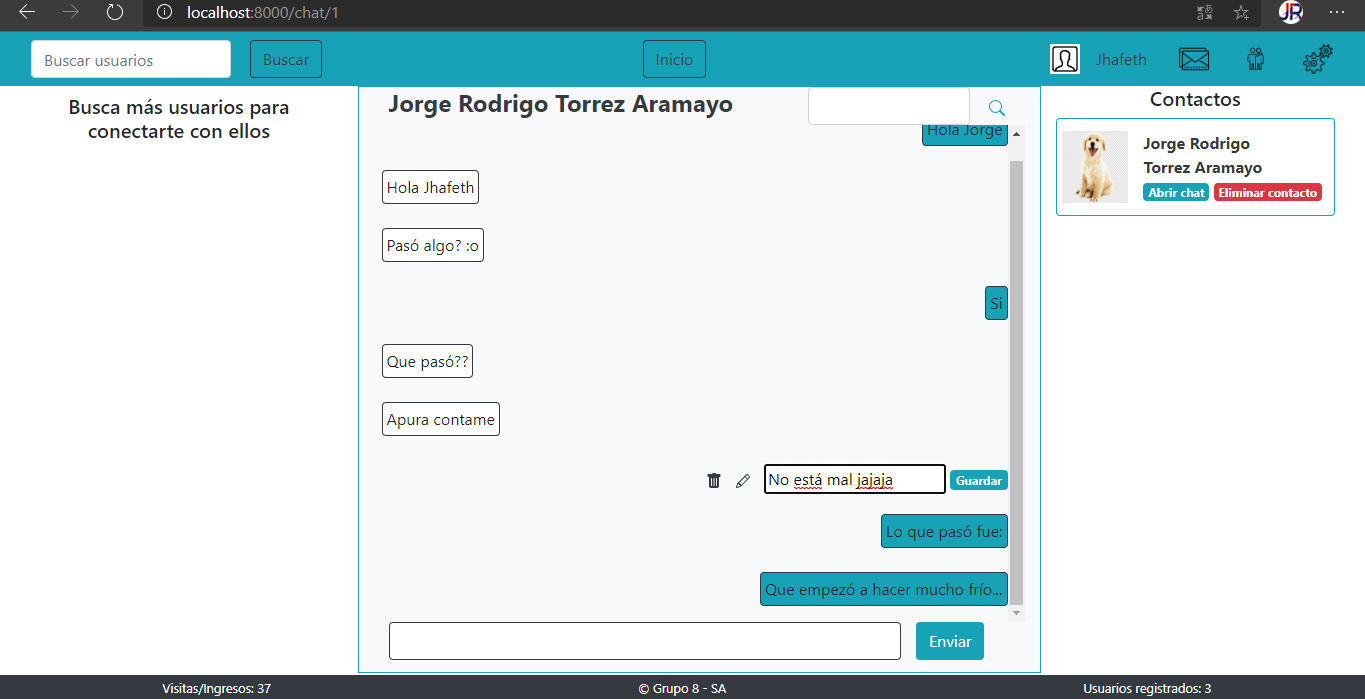
* Ver mensajes



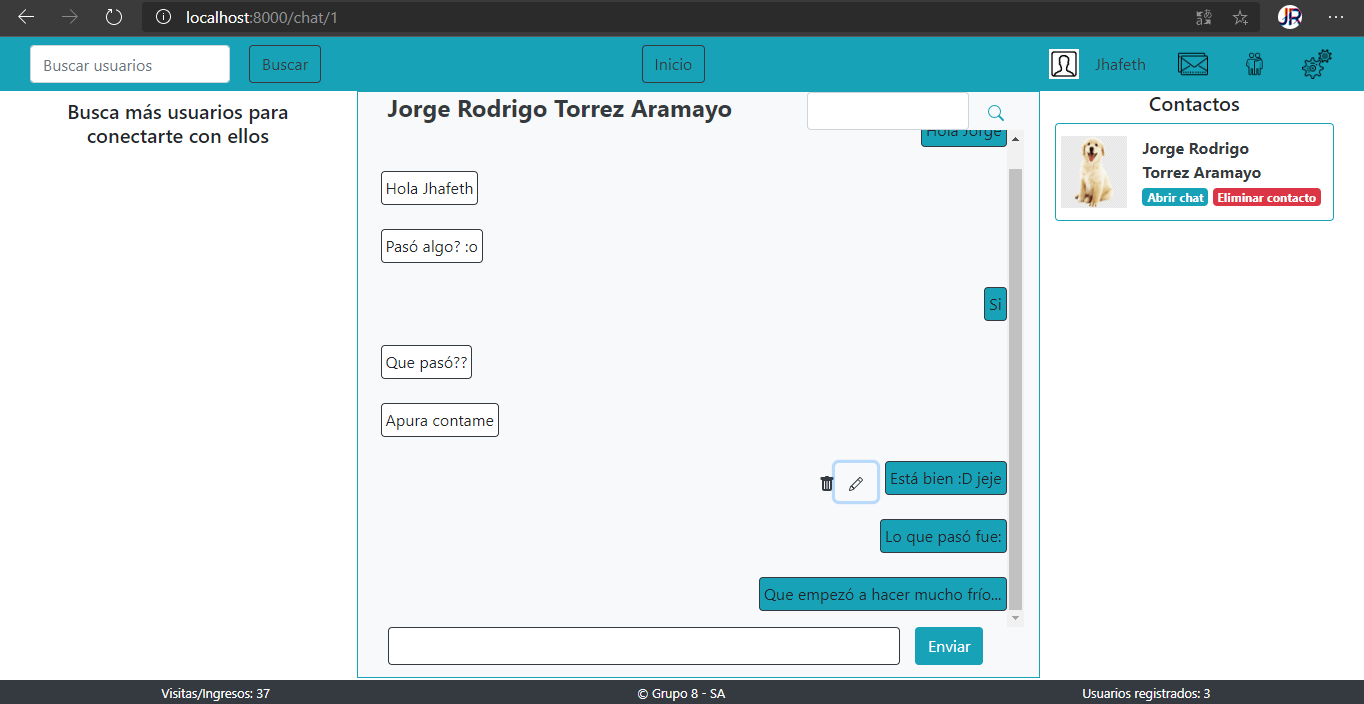


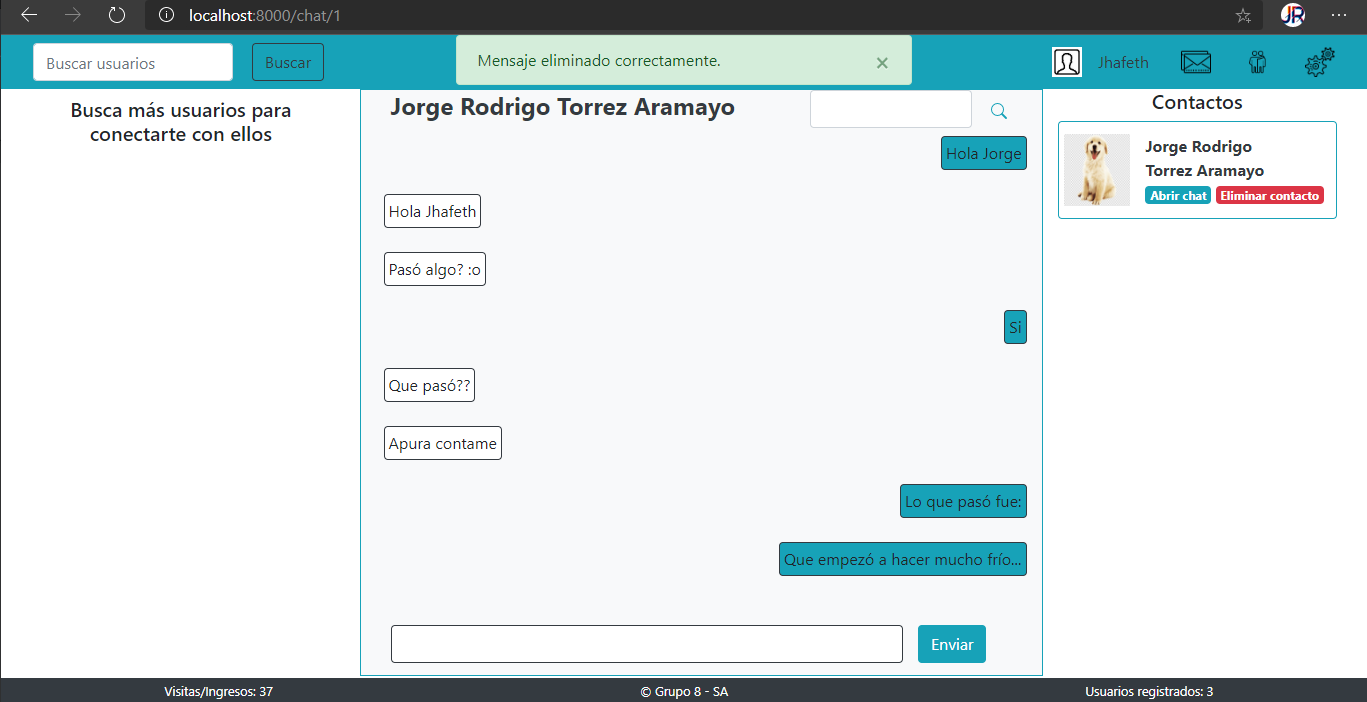
* Editar mensaje



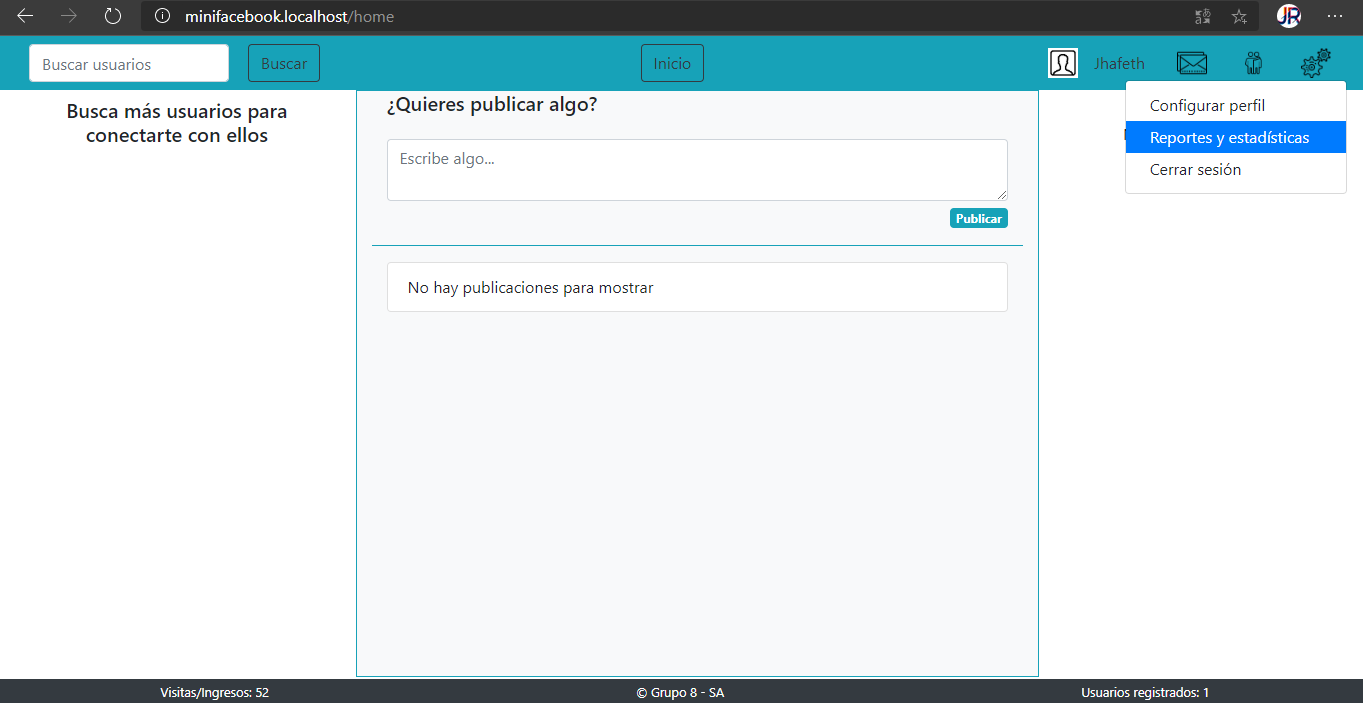


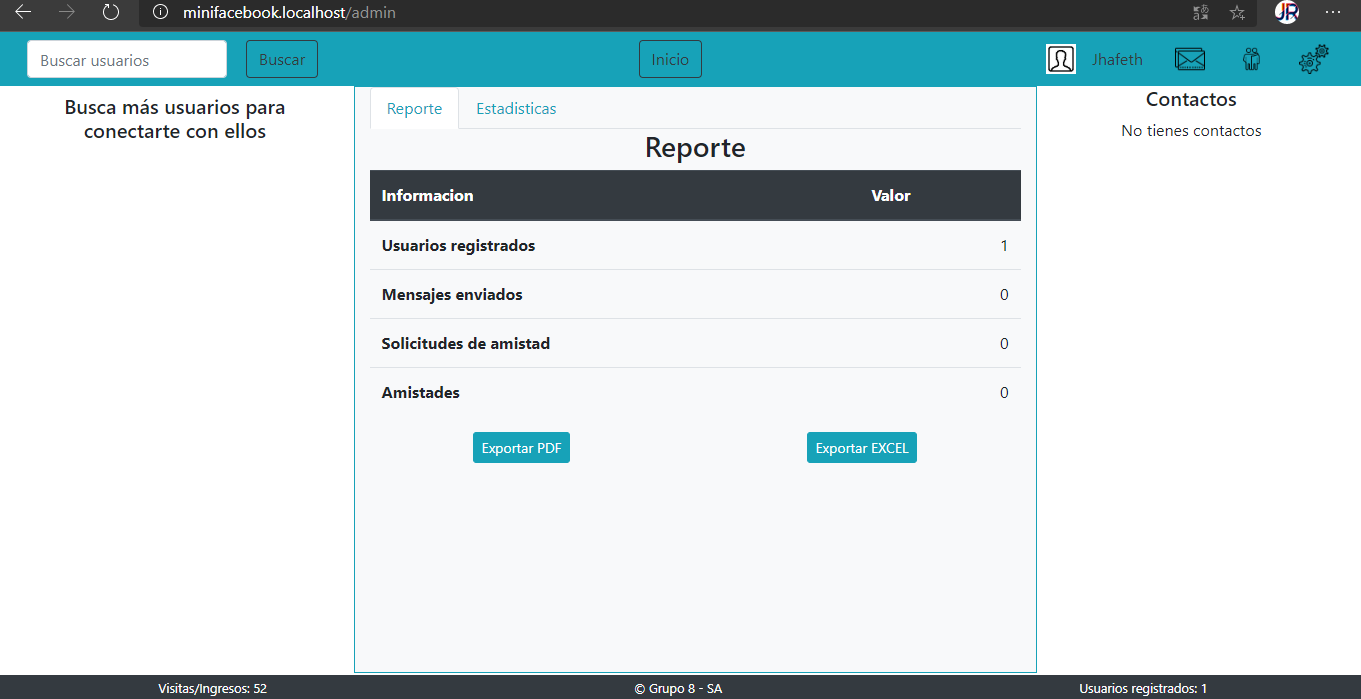
* Borrar mensaje



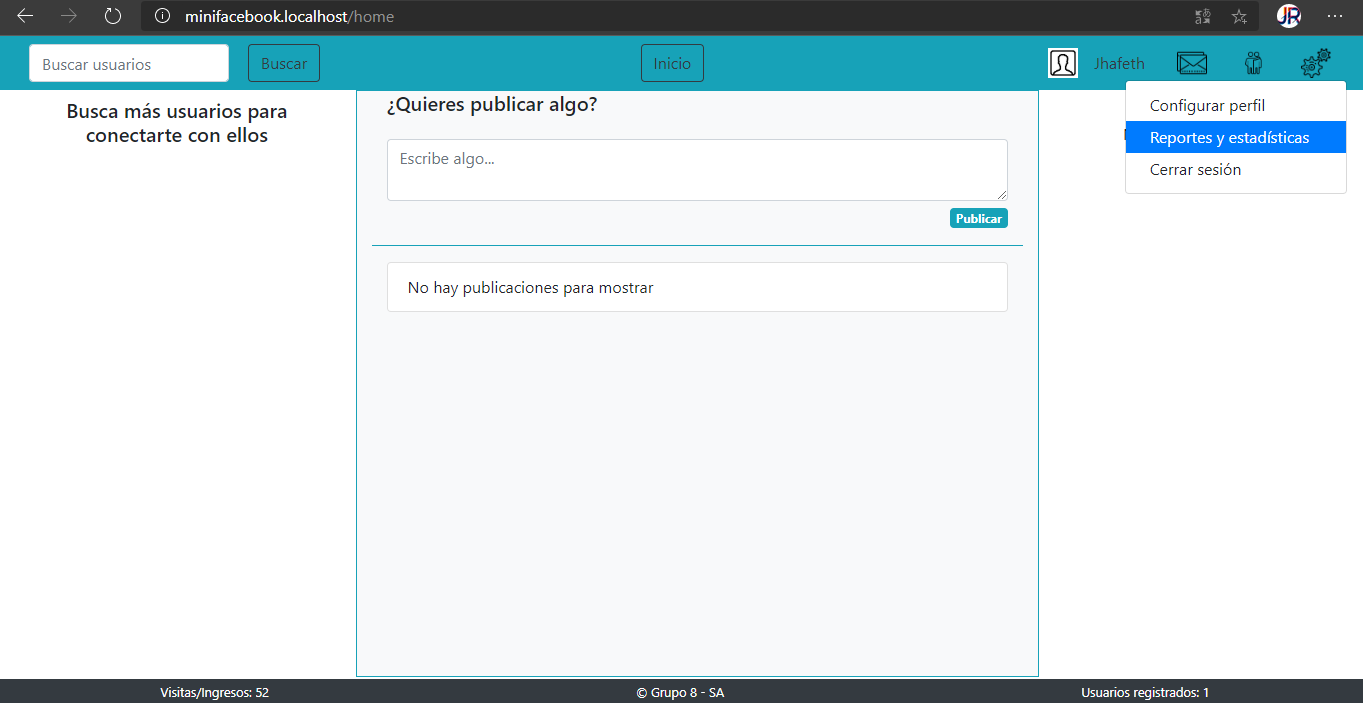


CU7 Generar reportes



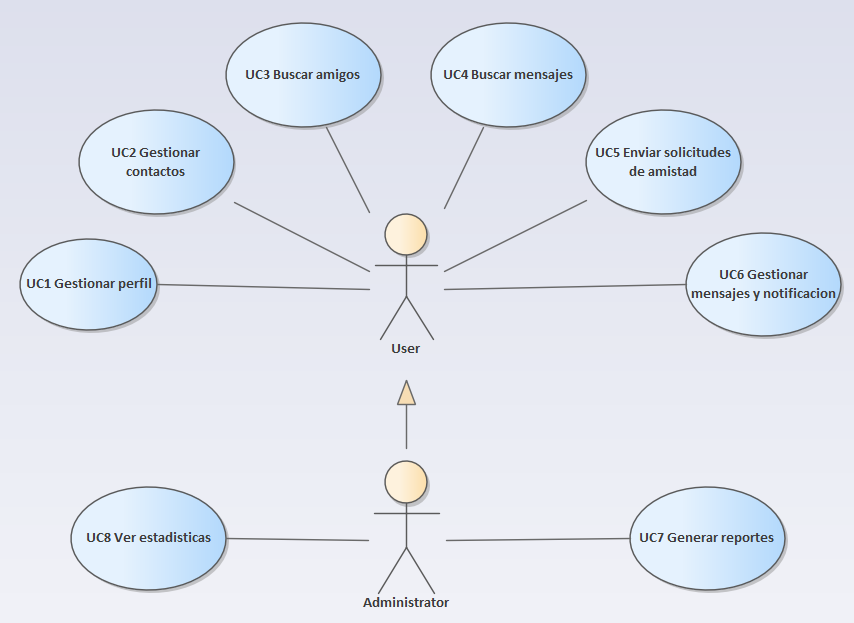


CU7 Generar reportes





* + 1. Modelo de casos de uso



* 1. Análisis
     1. Análisis de la arquitectura

Identificar los paquetes

P1. Models

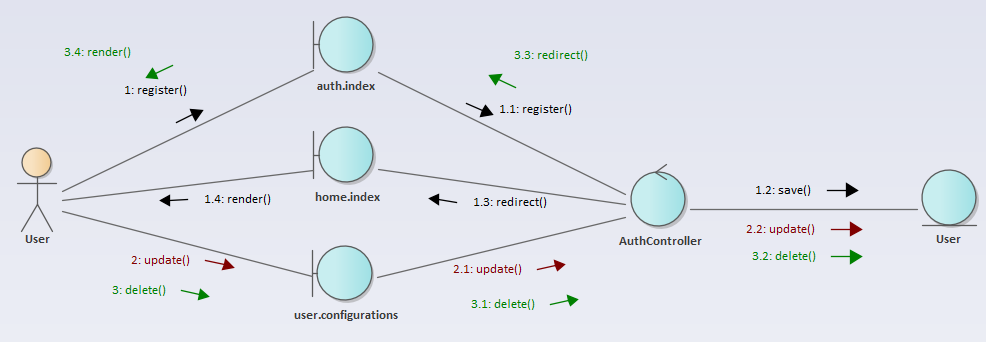
P2. Views

P3. Controllers

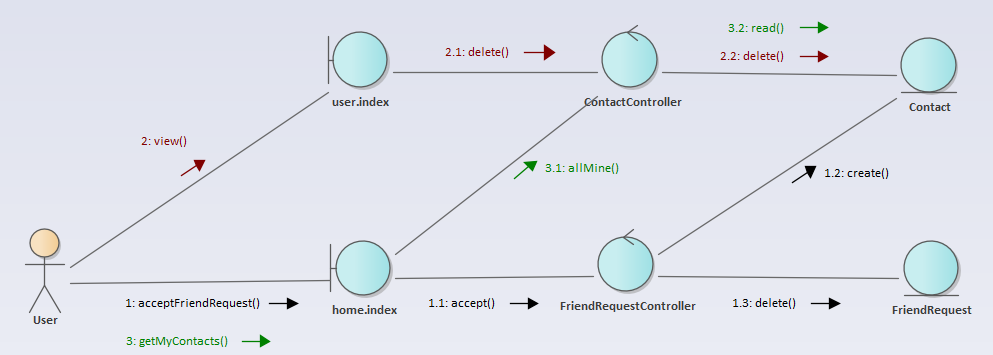
P4. Events

* + 1. Análisis de casos de uso

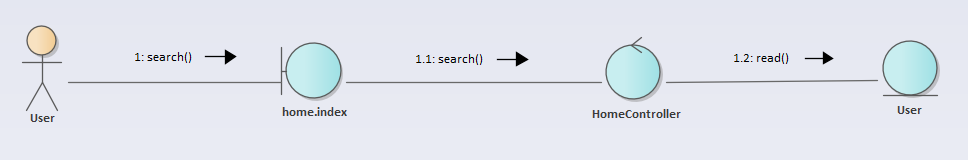
CU1 Gestionar perfil



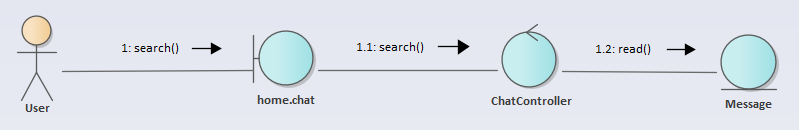
CU2 Gestionar contactos



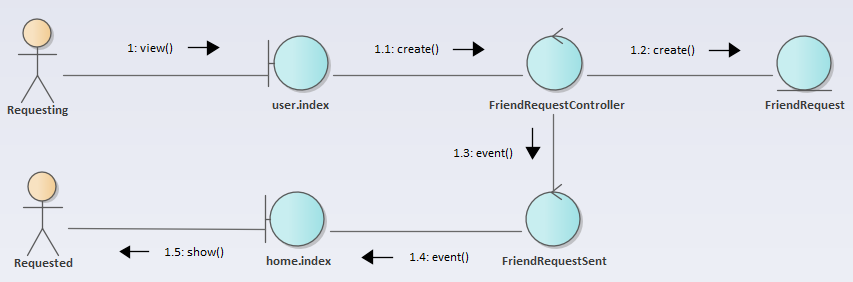
CU3 Buscar amigos



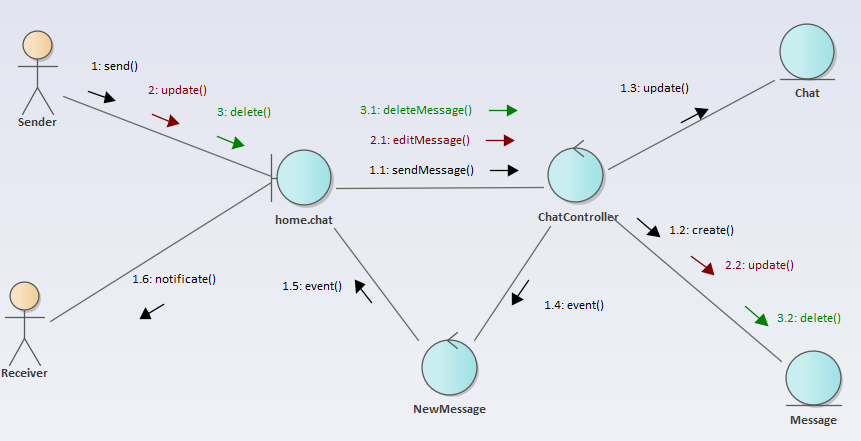
CU4 Buscar mensaje



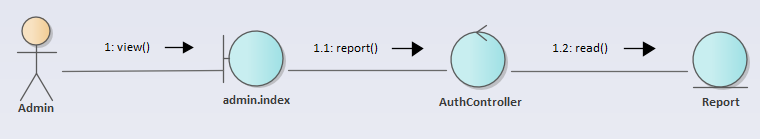
CU5 Enviar solicitudes de amistad



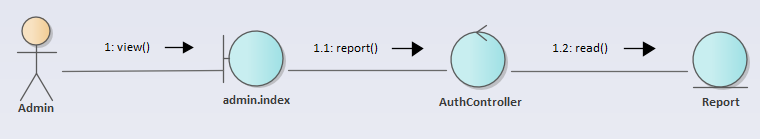
CU6 Gestionar mensajes y notificaciones



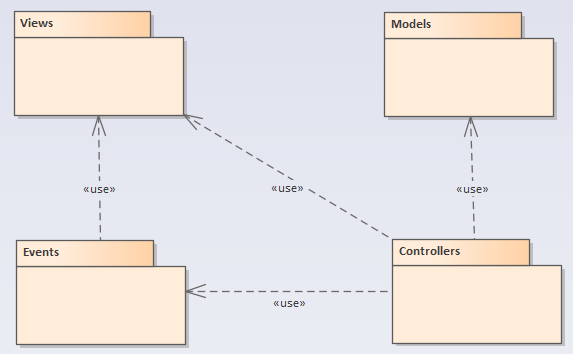
CU7 Generar reporte



CU8 Ver estadisticas

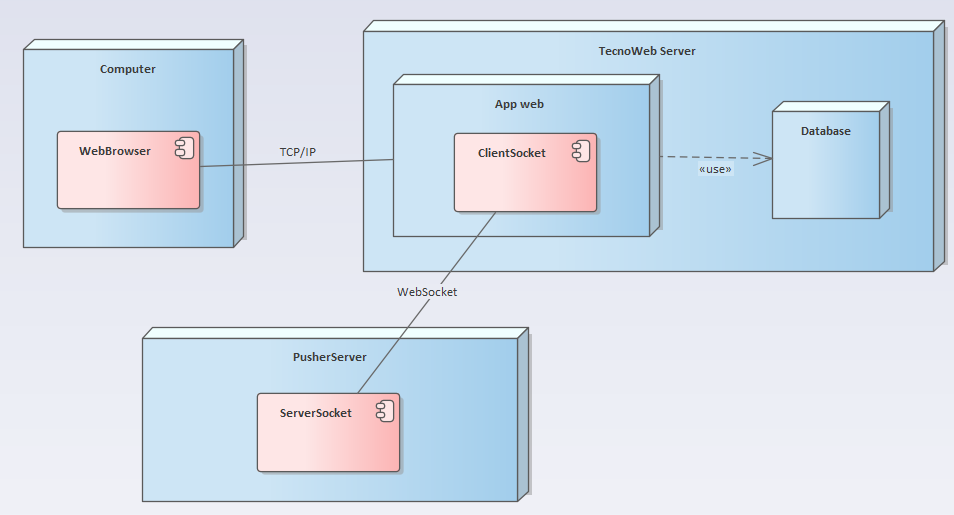


* + 1. Análisis de paquetes

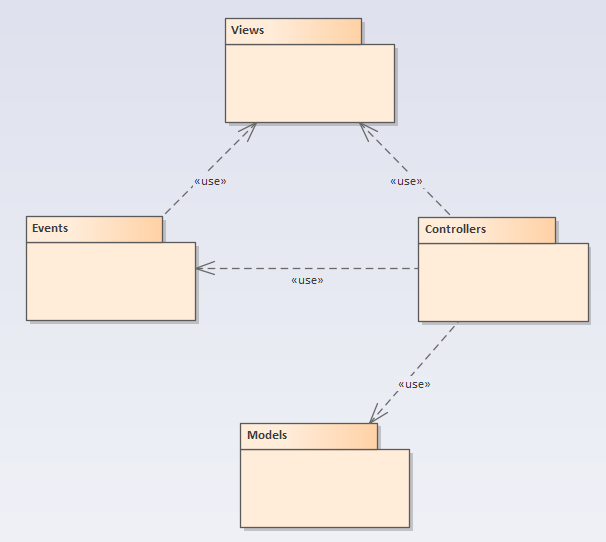


* 1. Diseño
     1. Diseño de la arquitectura

Diseño físico de la arquitectura

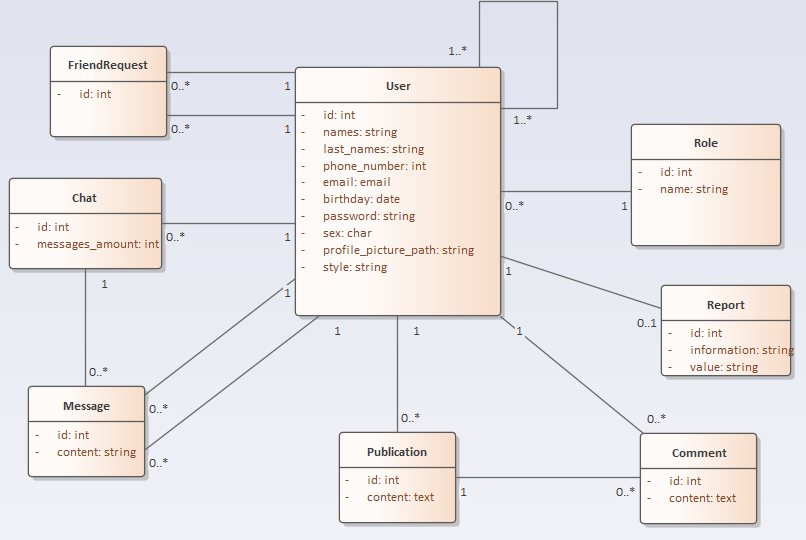


Diseño lógico de la arquitectura



* + 1. Diseño de datos

Diseño lógico de datos



MAPEO

|  |  |
| --- | --- |
| Llave primaria | Llave foranea |

**User**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| id | Names | Last\_names | phone\_number | email |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| birthday | password | sex | profile\_picture\_path | style | role\_id |

**Role**

|  |  |
| --- | --- |
| id | name |

**Report**

|  |  |  |
| --- | --- | --- |
| id | information | value |

**Comment**

|  |  |  |  |
| --- | --- | --- | --- |
| id | content | user\_id | publication\_id |

**Publication**

|  |  |  |
| --- | --- | --- |
| id | content | user\_id |

**Message**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| id | content | sender | receiver | chat\_id |

**Chat**

|  |  |  |  |
| --- | --- | --- | --- |
| id | content | invited | creator |

**FriendRequest**

|  |  |  |
| --- | --- | --- |
| id | requesting | requested |

**Contact**

|  |  |
| --- | --- |
| user\_a | user\_b |

DISEÑO FISICO DE DATOS

TABLA DE VOLUMEN

**User**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Descripcion** | **Tamaño** | **Nulo** | **Llave** |
| id | Numérico entero | identificador | 8 bytes | no | Primaria |
| names | Cadena | Nombres del usuario | 50 bytes | no | no |
| last\_names | Cadena | Apellidos del usuario | 50 bytes | no | no |
| phone\_number | Cadena | Número de celular del usuario | 8 bytes | si | no |
| email | Cadena | Correo del usuario | 50 bytes | no | no |
| birthday | Date | Fecha de nacimiento del usuario | 10 bytes | no | no |
| password | Cadena | Contraseña del usuario | 10 bytes | no | no |
| sex | Caracter | Sexo del usuario | 1 byte | no | no |
| profile\_picture\_path | Cadena | Ruta de la foto de perfil del usuario | 255 bytes | no | no |
| style | Cadena | Información de la personalizacion del GUI del usuario | 15 bytes | no | no |
| role\_id | Numérico entero | Identificador del rol del usuario | 8 bytes | no | Foranea |

**Role**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Descripcion** | **Tamaño** | **Nulo** | **Llave** |
| id | Numérico entero | identificador | 8 bytes | no | Primaria |
| name | Cadena | Nombre del rol | 255 bytes | no | no |

**Report**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Descripcion** | **Tamaño** | **Nulo** | **Llave** |
| id | Numérico entero | identificador | 8 bytes | no | Primaria |
| information | Cadena | Nombre de la informacion guardada | 255 bytes | no | no |
| value | Cadena | Valor de la informacion guardada | 255 bytes | no | no |

**Comment**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Descripcion** | **Tamaño** | **Nulo** | **Llave** |
| id | Numérico entero | identificador | 8 bytes | no | Primaria |
| content | Cadena | Contenido del comentario | 255 bytes | No | no |
| user\_id | Numerico entero | Identificador del creador del comentario | 8 bytes | no | foranea |
| publication\_id | Numerico entero | Identificador de la publicacion a la que pertenece el comentario | 8 bytes | no | foranea |

**Publication**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Descripcion** | **Tamaño** | **Nulo** | **Llave** |
| id | Numérico entero | identificador | 8 bytes | no | Primaria |
| content | Cadena | Contenido de la publicacion | 255 bytes | no | no |
| user\_id | Numerico entero | Identificador del creador de la publicacion | 8 bytes | no | foranea |

**Message**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Descripcion** | **Tamaño** | **Nulo** | **Llave** |
| id | Numérico entero | identificador | 8 bytes | no | Primaria |
| content | Cadena | Contenido del mensaje | 255 bytes | no | no |
| receiver | Numerico entero | Identificador del enviador del mensaje | 8 bytes | no | foranea |
| sender | Numerico entero | Identificador del recibidor del mensaje | 8 bytes | no | foranea |
| chat\_id | Numerico entero | Identificador del chat al que pertenece el mensaje | 8 bytes | no | foranea |

**Chat**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Descripcion** | **Tamaño** | **Nulo** | **Llave** |
| id | Numérico entero | identificador | 8 bytes | no | Primaria |
| messages\_amount | Numerico entero | Cantidad de mensajes en el chat | 8 bytes | no | no |
| invited | Numerico entero | Identificador del usuario invitado al chat | 8 bytes | no | foranea |
| creator | Numerico entero | Identificador del usuario creador del chat | 8 bytes | no | foranea |

FriendRequest

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Descripcion** | **Tamaño** | **Nulo** | **Llave** |
| id | Numérico entero | identificador | 8 bytes | no | Primaria |
| requesting | Numerico entero | Identificador del usuario solicitante | 8 bytes | no | foranea |
| requested | Numerico entero | Identificador del usuario solicitado | 8 bytes | no | foranea |

**Contact**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Descripcion** | **Tamaño** | **Nulo** | **Llave** |
| user\_a | Numérico entero | Identificador del usuario a | 8 bytes | no | foranea |
| user\_b | Numérico entero | Identificador del usuario a | 8 bytes | no | foranea |

SCRIPT

CREATE TABLE public.chats

(

id bigint NOT NULL DEFAULT nextval('chats\_id\_seq'::regclass),

messages\_amount bigint NOT NULL,

creator bigint NOT NULL,

invited bigint NOT NULL,

created\_at timestamp(0) without time zone,

updated\_at timestamp(0) without time zone,

CONSTRAINT chats\_pkey PRIMARY KEY (id),

CONSTRAINT chats\_creator\_foreign FOREIGN KEY (creator)

REFERENCES public.users (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION,

CONSTRAINT chats\_invited\_foreign FOREIGN KEY (invited)

REFERENCES public.users (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE public.chats

OWNER to grupo08sa;

CREATE TABLE public.comments

(

id bigint NOT NULL DEFAULT nextval('comments\_id\_seq'::regclass),

content text COLLATE pg\_catalog."default" NOT NULL,

publication\_id bigint NOT NULL,

user\_id bigint NOT NULL,

created\_at timestamp(0) without time zone,

updated\_at timestamp(0) without time zone,

CONSTRAINT comments\_pkey PRIMARY KEY (id),

CONSTRAINT comments\_publication\_id\_foreign FOREIGN KEY (publication\_id)

REFERENCES public.publications (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION,

CONSTRAINT comments\_user\_id\_foreign FOREIGN KEY (user\_id)

REFERENCES public.users (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE public.comments

OWNER to grupo08sa;

CREATE TABLE public.contacts

(

user\_a bigint NOT NULL,

user\_b bigint NOT NULL,

created\_at timestamp(0) without time zone,

updated\_at timestamp(0) without time zone,

CONSTRAINT contacts\_user\_a\_foreign FOREIGN KEY (user\_a)

REFERENCES public.users (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION,

CONSTRAINT contacts\_user\_b\_foreign FOREIGN KEY (user\_b)

REFERENCES public.users (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE public.contacts

OWNER to grupo08sa;

CREATE TABLE public.friend\_requests

(

id bigint NOT NULL DEFAULT nextval('friend\_requests\_id\_seq'::regclass),

requesting bigint NOT NULL,

requested bigint NOT NULL,

created\_at timestamp(0) without time zone,

updated\_at timestamp(0) without time zone,

CONSTRAINT friend\_requests\_pkey PRIMARY KEY (id),

CONSTRAINT friend\_requests\_requested\_foreign FOREIGN KEY (requested)

REFERENCES public.users (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION,

CONSTRAINT friend\_requests\_requesting\_foreign FOREIGN KEY (requesting)

REFERENCES public.users (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE public.friend\_requests

OWNER to grupo08sa;

CREATE TABLE public.messages

(

id bigint NOT NULL DEFAULT nextval('messages\_id\_seq'::regclass),

sender bigint NOT NULL,

receiver bigint NOT NULL,

content character varying(255) COLLATE pg\_catalog."default" NOT NULL,

chat\_id bigint NOT NULL,

created\_at timestamp(0) without time zone,

updated\_at timestamp(0) without time zone,

CONSTRAINT messages\_pkey PRIMARY KEY (id),

CONSTRAINT messages\_chat\_id\_foreign FOREIGN KEY (chat\_id)

REFERENCES public.chats (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION,

CONSTRAINT messages\_receiver\_foreign FOREIGN KEY (receiver)

REFERENCES public.users (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION,

CONSTRAINT messages\_sender\_foreign FOREIGN KEY (sender)

REFERENCES public.users (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE public.messages

OWNER to grupo08sa;

CREATE TABLE public.migrations

(

id integer NOT NULL DEFAULT nextval('migrations\_id\_seq'::regclass),

migration character varying(255) COLLATE pg\_catalog."default" NOT NULL,

batch integer NOT NULL,

CONSTRAINT migrations\_pkey PRIMARY KEY (id)

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE public.migrations

OWNER to grupo08sa;

CREATE TABLE public.publications

(

id bigint NOT NULL DEFAULT nextval('publications\_id\_seq'::regclass),

content text COLLATE pg\_catalog."default" NOT NULL,

user\_id bigint NOT NULL,

created\_at timestamp(0) without time zone,

updated\_at timestamp(0) without time zone,

CONSTRAINT publications\_pkey PRIMARY KEY (id),

CONSTRAINT publications\_user\_id\_foreign FOREIGN KEY (user\_id)

REFERENCES public.users (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE public.publications

OWNER to grupo08sa;

CREATE TABLE public.reports

(

id bigint NOT NULL DEFAULT nextval('reports\_id\_seq'::regclass),

information character varying(255) COLLATE pg\_catalog."default" NOT NULL,

value character varying(255) COLLATE pg\_catalog."default" NOT NULL,

type character varying(255) COLLATE pg\_catalog."default" NOT NULL,

created\_at timestamp(0) without time zone,

updated\_at timestamp(0) without time zone,

CONSTRAINT reports\_pkey PRIMARY KEY (id)

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE public.reports

OWNER to grupo08sa;

CREATE TABLE public.roles

(

id bigint NOT NULL DEFAULT nextval('roles\_id\_seq'::regclass),

name character varying(255) COLLATE pg\_catalog."default" NOT NULL,

created\_at timestamp(0) without time zone,

updated\_at timestamp(0) without time zone,

CONSTRAINT roles\_pkey PRIMARY KEY (id)

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE public.roles

OWNER to grupo08sa;

CREATE TABLE public.users

(

id bigint NOT NULL DEFAULT nextval('users\_id\_seq'::regclass),

names character varying(255) COLLATE pg\_catalog."default" NOT NULL,

last\_names character varying(255) COLLATE pg\_catalog."default" NOT NULL,

phone\_number character varying(255) COLLATE pg\_catalog."default",

email character varying(255) COLLATE pg\_catalog."default" NOT NULL,

birthday date,

password character varying(255) COLLATE pg\_catalog."default" NOT NULL,

sex character(255) COLLATE pg\_catalog."default" NOT NULL,

profile\_picture\_path character varying(255) COLLATE pg\_catalog."default" NOT NULL,

style character varying(255) COLLATE pg\_catalog."default",

role\_id bigint NOT NULL,

created\_at timestamp(0) without time zone,

updated\_at timestamp(0) without time zone,

CONSTRAINT users\_pkey PRIMARY KEY (id),

CONSTRAINT users\_role\_id\_foreign FOREIGN KEY (role\_id)

REFERENCES public.roles (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE public.users

OWNER to grupo08sa;

* 1. Implementación

Lenguaje de programacion

PHP 7.2.5

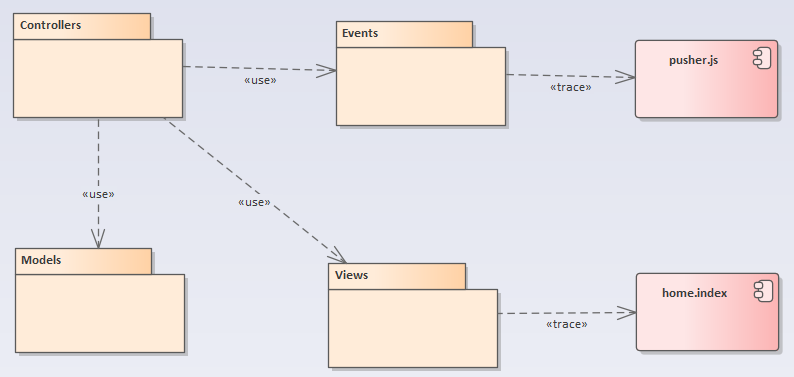
Base de datos

Postgresql

Sistema operativo

Linux fedora 31

* + 1. Implementación de la arquitectura del sistema



1. Marco teórico

Las redes sociales son el medio más grande que existe actualmente para la comunicación a distancia a través del internet. A mediados de los 90 salió la primera red social para el uso público. Hoy en día las redes sociales han pasado a ser más que simples medios de comunicación, y es que su capacidad de integración con la era digital ha cambiado en gran medida su comportamiento, siendo utilizadas para diferentes áreas como la publicidad, juegos, entretenimiento, media, etc.

En este proyecto se ha realizado una réplica de la red social Facebook a baja escala y aplicada a una aplicacion web, dando oportunidad a que se pueda profundizar en otros aspectos que también son usados por nosotros en el dia a dia.

1. Desarrollo

Se ha realizado una comunicación de aplicacion web para la simulación del uso de una red social. Implementando un programa con Php que permite detectar la interaccion del usuario y así validar las instrucciones enviadas por el cliente.

Para poder identificar las acciones del usuario se han creado paginas y seguidamente se hacen consultas a la base de datos para registrar, leer o actualizar datos.

1. Resultados

Una vez terminada la implementación, se ha detectado la falta de interacción por parte del cliente para poder facilitar las instrucciones requeridas. No obstante, se han aplicado soluciones alternativas permitiendo un flujo similar al de una aplicación movil con sus respectivas validaciones.

1. Conclusiones

Se ha creado una base de datos para el almacenamiento de la informacion. La implementacion de la aplicacion php ha sido concluida con exito y se ha establecido una conexion websocket con el servidor pusher para las notificaciones.

1. Recomendaciones

Para una próxima versión de este software se recomienda utilizar otras herramientas más complejas para poder sacar el máximo provecho a la red social y a todas sus funcionalidades que se le pueden añadir.

1. Bibliografía citada

Primera red social:  
<https://www.bbc.com/mundo/noticias-48558989#:~:text=BBC%20Extra-,Six%20Degrees%3A%20c%C3%B3mo%20fue%20y%20qui%C3%A9n%20cre%C3%B3%20la%20primera%20red,teor%C3%ADa%20de%20los%20%22seis%20grados%22&text=Derechos%20de%20autor%20de%20la,Eran%20mediados%20de%20los%2090.>