UNIVERSIDAD AUTÓNOMA GABRIEL RENÉ MORENO

FACULTAD DE INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN Y TELECOMUNICACIONES



Proyecto No 1

Mini Facebook

**GRUPO 04-SA:**

CADIMA BARRERO JHAFETH 216008131

TORREZ ARAMAYO JORGE RODRIGO 217051367

MENDOZA ALLERDING JOSE CARLOS 216031850

**SIGLA:** INF513-SA **MATERIA:** TECNOLOGIA WEB

**SEMESTRE:** 1-2020

**DOCENTE:** M.sc. Ing. EVANS BALCÁZAR VEIZAGA

Santa Cruz-Bolivia

Indice

1. Introduccion........................................................................................................................................3
2. Antecedentes y justificación................................................................................................................3
3. El problema.........................................................................................................................................3
   1. Situación problemática..............................................................................................................3
   2. Formulación del problema.........................................................................................................3
4. Objetivos.............................................................................................................................................3
   1. Objetivo general........................................................................................................................3
   2. Objetivos específicos.................................................................................................................3
5. Metodología.........................................................................................................................................4
   1. Captura de requisitos.................................................................................................................4
      1. Actores y casos de uso.......................................................................................4
         1. Actores................................................................................................................4
         2. Casos de uso......................................................................................................4
         3. Tabla de actores y casos de uso........................................................................4
      2. Detallar casos de uso.........................................................................................5
      3. Modelo de casos de uso....................................................................................12
   2. Análisis......................................................................................................................................13
      1. Análisis de la arquitectura..................................................................................13
         1. Identificar los paquetes......................................................................................13
      2. Análisis de casos de uso....................................................................................13
      3. Análisis de paquetes...........................................................................................17
   3. Diseño........................................................................................................................................18
      1. Diseño de la arquitectura....................................................................................18
         1. Diseño físico de la arquitectura...........................................................................18
         2. Diseño lógico de la arquitectura..........................................................................18
      2. Diseño de datos..................................................................................................19
         1. Diseño lógico de datos.........................................................................................19
            1. Diagrama de clases................................................................................19
            2. Mapeo.....................................................................................................19
         2. Diseño físico de datos..........................................................................................20
            1. Tabla de volumen....................................................................................20
            2. Script.......................................................................................................23
   4. Implementación...........................................................................................................................26
      1. Plataforma de desarrollo de software...........................................................................26
         1. Lenguaje de programación..................................................................................26
         2. Base de datos......................................................................................................26
         3. Sistema operativo................................................................................................26
      2. Implementacion de la arquitectura del sistema...........................................................27
6. Marco teórico........................................................................................................................................27
7. Desarrollo..............................................................................................................................................27
8. Resultados............................................................................................................................................28
9. Conclusiones.........................................................................................................................................28
10. Recomendaciones...............................................................................................................................28
11. Bibliografía citada................................................................................................................................28
12. Introducción

El presente proyecto ha sido realizado para la demostración de un sistema (en este caso red social) a través de correos. Este sistema funciona con comandos enviados por un usuario desde su servidor de correos. 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 justificación

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 los correos electrónicos ha bajado mucho su uso en áreas no empresariales.

* 1. Formulación del problema

Al no utilizarse los correos como una alternativa a las redes sociales 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 un servidor de correos y una aplicación en java para la lectura de emails.
* Crear una base de datos para el almacenamiento de datos.
* Implementar una aplicación java para la lógica de negocio y reconocimiento de comandos.

1. Metodología
   1. Captura de requisitos
      1. Actores y casos de uso
         1. Actores

* + - 1. Casos de uso
* CU1 Gestionar perfil
* CU2 Gestionar contactos
* CU3 Buscar amigo
* CU4 Buscar mensaje
* CU5 Enviar solicitud de amistad
* CU6 Gestionar mensajes
* CU7 Generar reportes
* CU8 Ver estadísticas
  + - 1. Tabla de actores y casos de uso

|  |  |  |
| --- | --- | --- |
| CU | Usuario | Adminstrador |
| CU1 Gestionar perfil | X |  |
| CU2 Gestionar contactos | X |  |
| CU3 Buscar amigo | X |  |
| CU4 Buscar mensaje | X |  |
| CU5 Enviar solicitud de amistad | X |  |
| CU6 Gestionar mensajes | X |  |
| CU7 Generar reportes |  | X |
| CU8 Ver estadísticas |  | X |

* + 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** | Ninguno |
| **Proceso** | **Registrar**:   1. El usuario envía un comando asignado para el registro de usuarios con los siguientes parámetros: nombres, apellido paterno, apellido materno, fecha de nacimiento (DD-MM-AAAA), sexo, celular, correo, 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 la petición con un mensaje de confirmación   **Modificar**:   1. El usuario envía un comando asignado para la modificación de usuarios con los siguientes parámetros: nombres, apellido paterno, apellido materno, fecha de nacimiento (DD-MM-AAAA), sexo, celular, correo 2. El sistema valida los datos 3. El sistema actualiza la información en la base de datos 4. El sistema responde la petición con un mensaje de confirmación   **Eliminar**   1. El usuario envía un comando asignado para la eliminación de usuarios con los siguientes parámetros: id, contraseña 2. El sistema valida los datos 3. El sistema elimina el usuario de la base de datos 4. El sistema responde la petición con un mensaje de confirmación   **Cambiar contraseña**   1. El usuario envía un comando asignado para la actualización de contraseñas con los siguientes parámetros: id, antigua contraseña, nueva contraseña 2. El sistema valida los datos 3. El sistema actualiza la información de la base de datos 4. El sistema responde con un mensaje de confirmación |
| **Post-condición** | Ninguna |
| **Excepción** | **Registrar**  2. Datos incorrectos  **Modificar**  2. Datos incorrectos  **Eliminar**  2. Datos incorrectos  **Cambiar contraseña**  2. Datos incorrectos |

* CU2 Gestionar contactos

|  |  |
| --- | --- |
| **Caso de uso** | CU2 Gestionar contactos |
| **Propósito** | Listar, registrar y eliminar los contactos de un usuario. |
| **Resumen** | El software 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**:   * Recibir una solicitud de amistad de otro usuario. |
| **Proceso** | **Registrar**:   1. El usuario pide la lista de solicitudes pendientes a través de un comando. 2. El usuario escogerá qué contactos registrar en base a la lista obtenida y efectuará dicho registro a través de un comando con el sistema 3. El sistema valida los datos y registra al nuevo contacto a ambos usuarios. 4. El sistema responde la petición con un mensaje de confirmación   **Listar**   1. El usuario envía un comando asignado para ver todos sus contactos 2. El sistema responde la petición con a lista de contactos.   **Eliminar**   1. El usuario envía un comando asignado para la eliminación de un contacto 2. El sistema valida los datos 3. El sistema elimina al contacto del usuario 4. El sistema responde la petición con un mensaje de confirmación |
| **Post-condición** | Gestionar mensajes |
| **Excepción** | **Registrar**  2. 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 software nos permitirá encontrar a los amigos que deseamos contactar, mostrando la información básica del usuario. |
| **Actores** | Usuario |
| **Actor iniciador** | Usuario |
| **Pre-condición** | Ninguno |
| **Proceso** | **Buscar**   1. El usuario envía un comando asignado para ver las coincidencias de su búsqueda 2. El sistema responde la petición con la lista de usuarios coincidentes |
| **Post-condición** | Enviar solicitud de amistad |
| **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** | Ninguno |
| **Proceso** | **Listar**   1. El usuario envía un comando asignado para ver las coincidencias de su búsqueda 2. El sistema responde la petición con a lista de mensajes coincidentes |
| **Post-condición** | Gestionar mensajes |
| **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** | Buscar amigo |
| **Proceso** | 1. El usuario envía un comando asignado para solicitar ser su amigo 2. El sistema responde con un mensaje de confirmación |
| **Post-condición** | Gestionar contacto |
| **Excepción** | No existe el usuario |

* CU6 Gestionar mensajes

|  |  |
| --- | --- |
| **Caso de uso** | CU6 Gestionar mensajes |
| **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** | Gestionar contacto |
| **Proceso** | **Registrar**   1. El usuario envía un comando asignado para crear un mensaje y enviarlo 2. El sistema responde la petición con un mensaje de confirmación   **Modificar**   1. El usuario envía un comando asignado para editar un mensaje ya enviado 2. El sistema valida los datos 3. El sistema responde con un mensaje de confirmación   **Eliminar**   1. El usuario envía un comando asignado para eliminar un mensaje ya enviado 2. El sistema valida los datos 3. El sistema responde con un mensaje de confirmación |
| **Post-condición** | Ninguno |
| **Excepción** | **Modificar**   1. El mensaje no existe   **Eliminar**   1. El mensaje no existe |

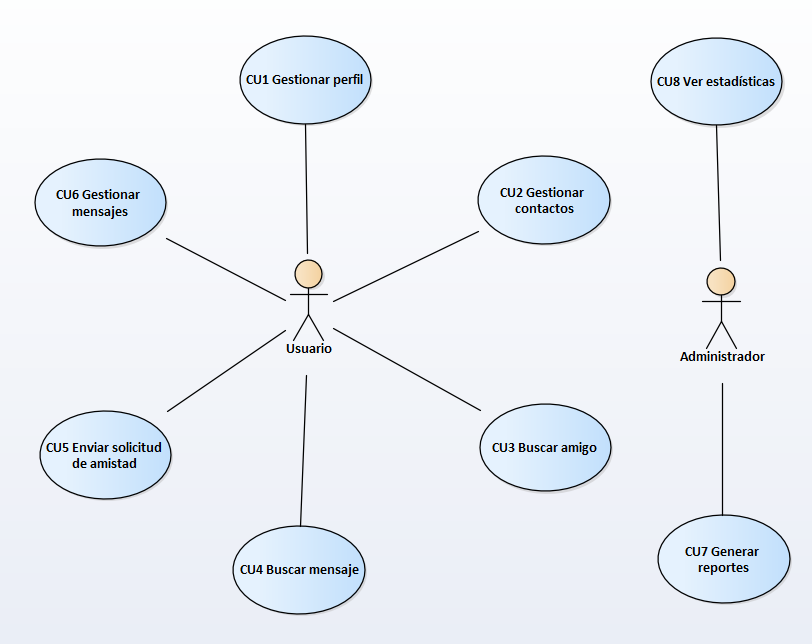
* CU7 Generar reportes

|  |  |
| --- | --- |
| **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** | Ninguno |
| **Proceso** | 1. El administrador envía un comando asignado para poder ver el reporte generado por el sistema 2. El sistema responde la petición con el reporte |
| **Post-condición** | Ninguno |
| **Excepción** | Ninguno |

* CU8 Ver estadísticas

|  |  |
| --- | --- |
| **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** | Ninguno |
| **Proceso** | 1. El administrador envía un comando asignado para poder ver la estadística generada por el sistema 2. El sistema responde la petición con la estadística |
| **Post-condición** | Ninguno |
| **Excepción** | Ninguno |

* + 1. Modelo de casos de uso



* 1. Análisis
     1. Análisis de la arquitectura
        1. Identificar los paquetes

P1 Comunicaciones

P2 Negocio

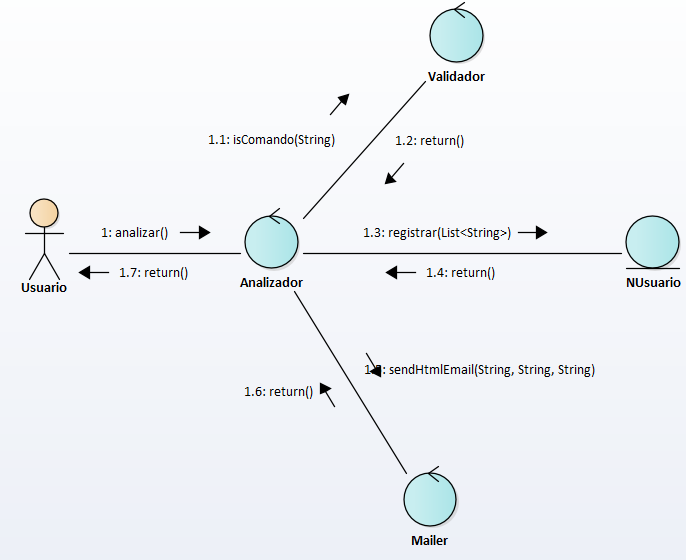
P3 Datos

P4 Templates

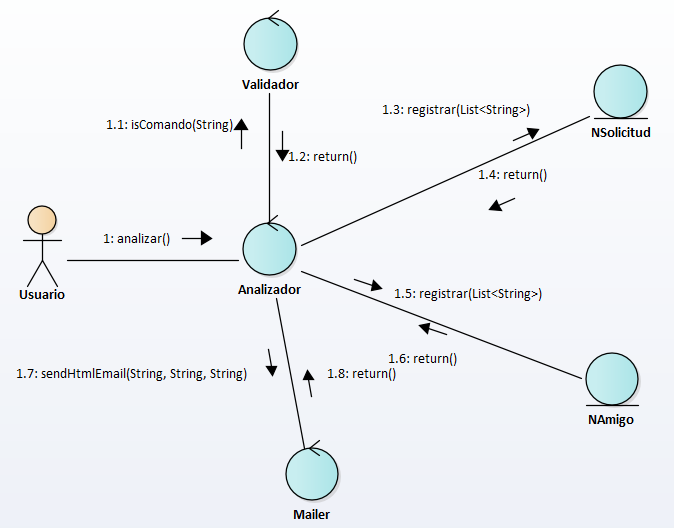
P5 MiniFB

* + 1. Análisis de casos de uso

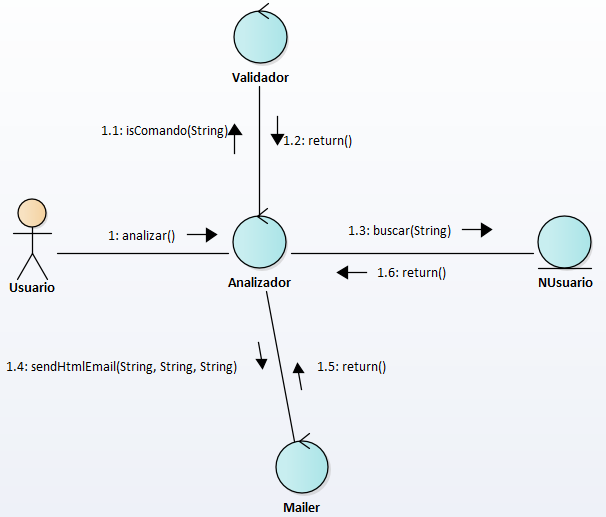
CU1 Gestionar perfil



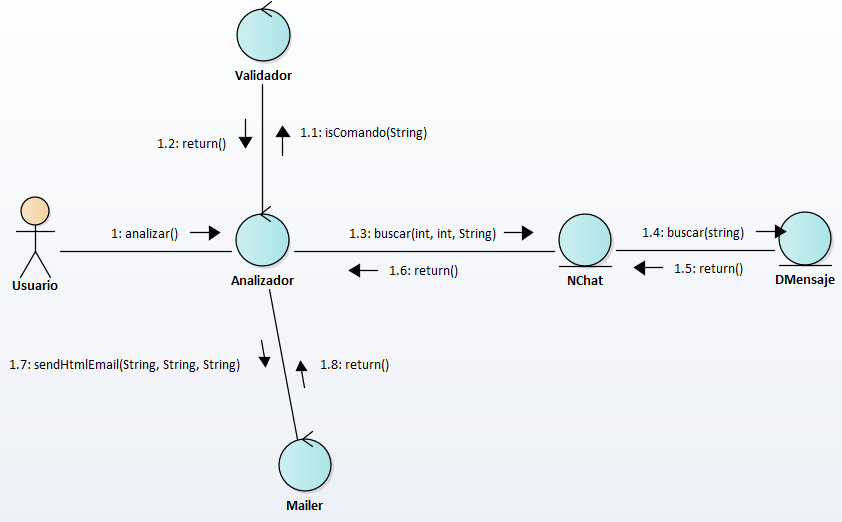
CU2 Gestionar contactos



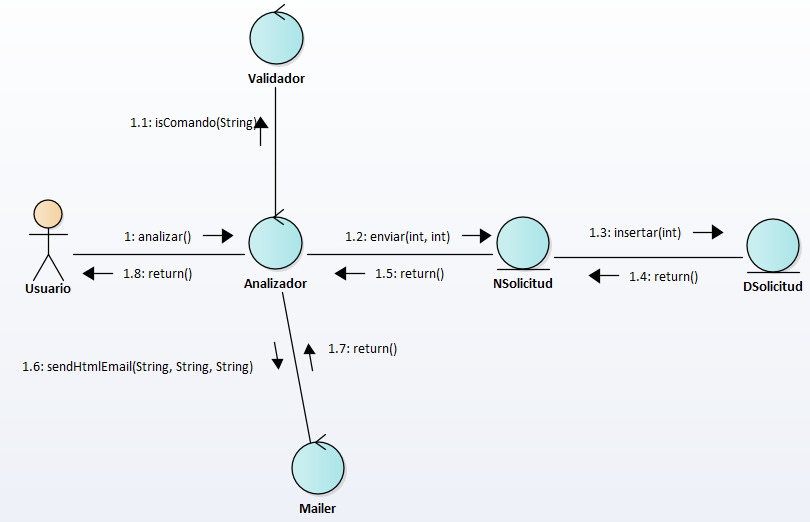
CU3 Buscar amigo



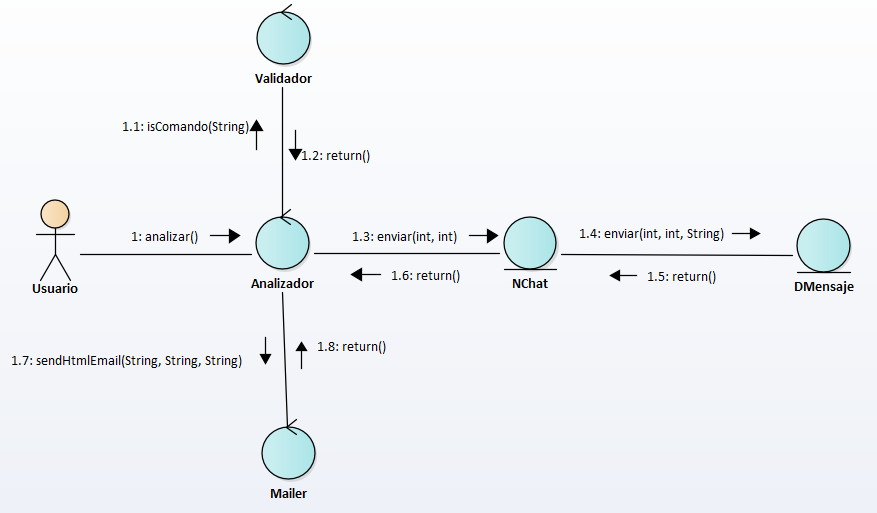
CU4 Buscar mensaje



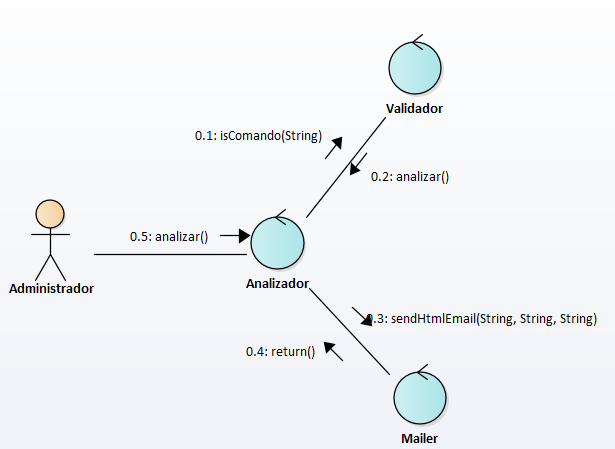
CU5 Enviar solicitud de amistad



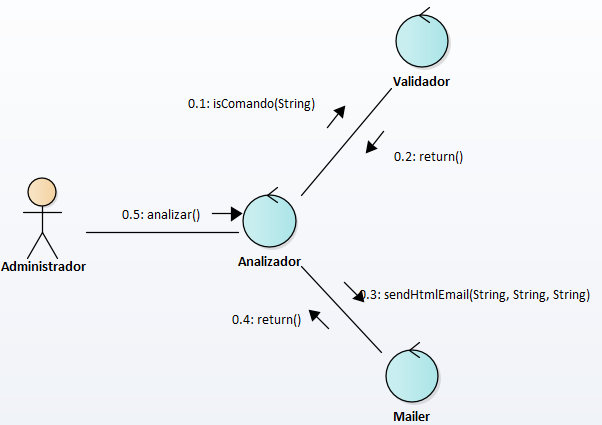
CU6 Gestionar mensajes



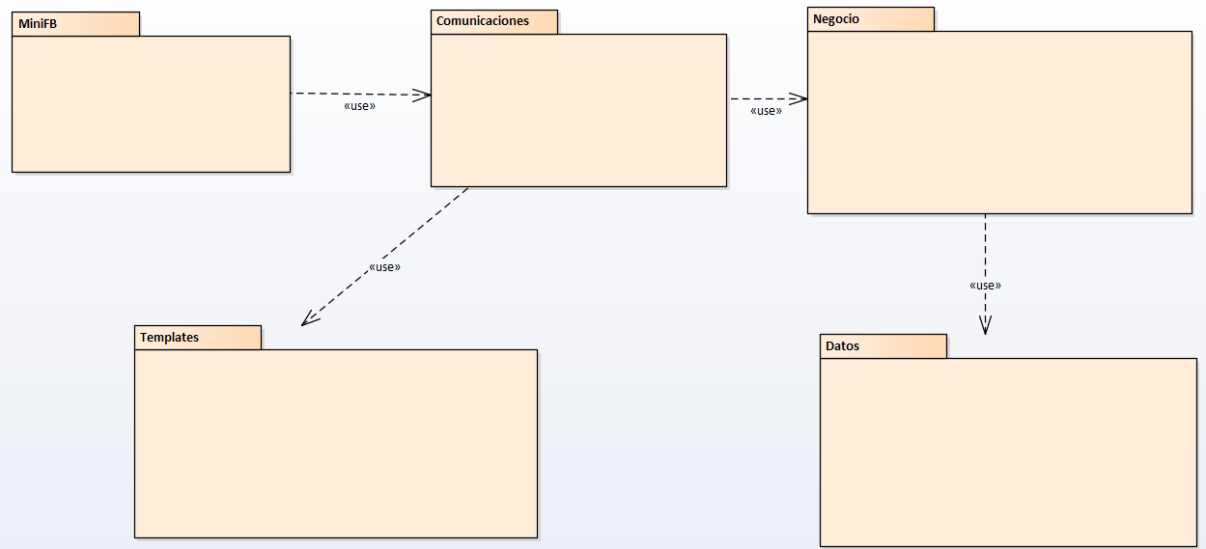
CU7 Generar reportes



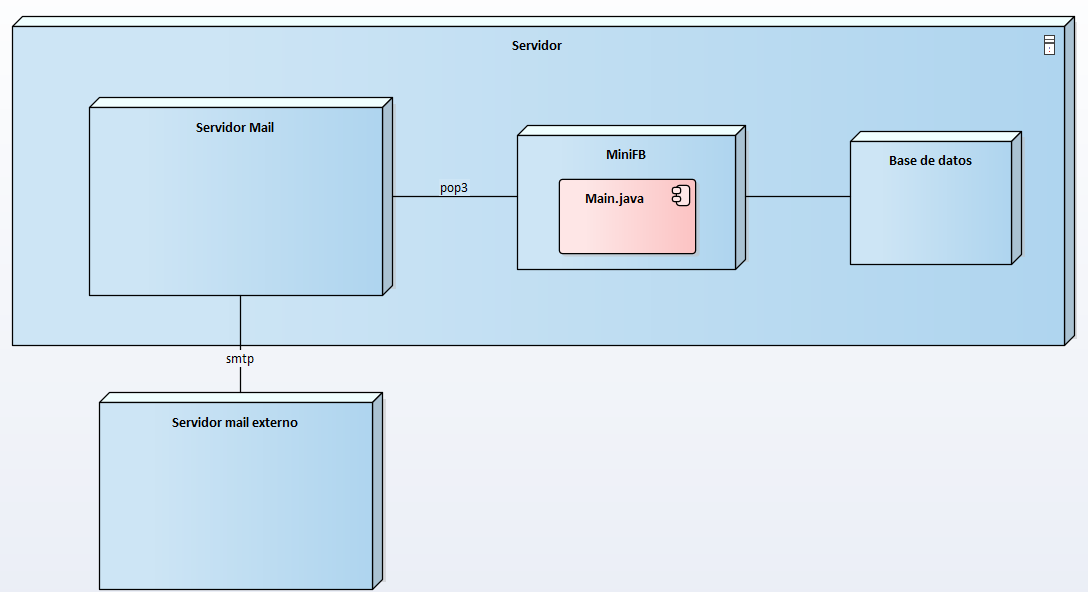
CU8 Ver estadísticas



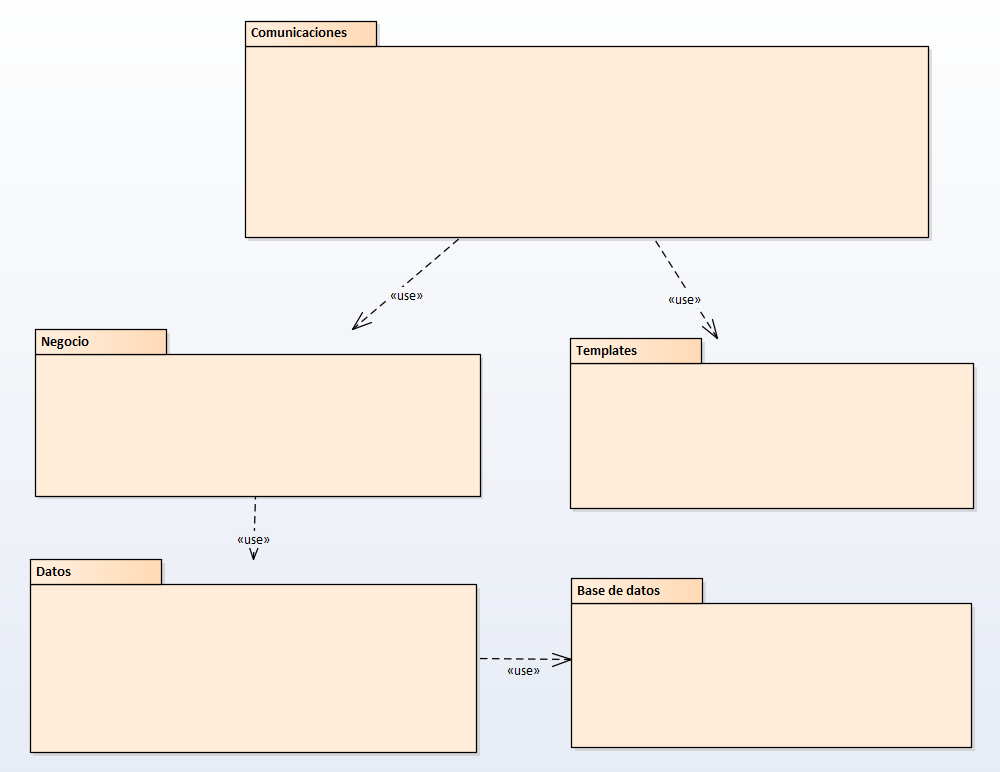
* + 1. Análisis de paquetes



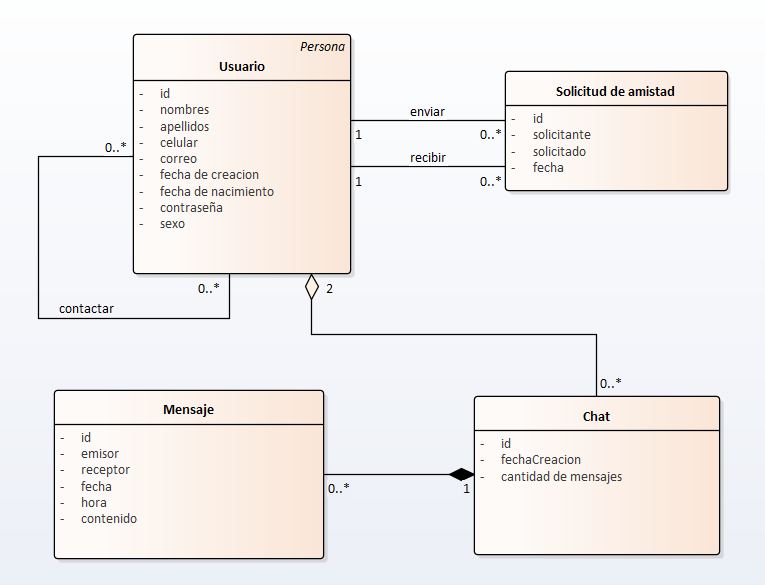
* 1. Diseño
     1. DIseño de la arquitectura
        1. Diseño físico de la arquitectura



* + - 1. Diseño lógico de la arquitectura



* + 1. DIseño de datos
       1. Diseño lógico de datos
          1. Diagrama de clases



* + - * 1. Mapeo

|  |  |
| --- | --- |
| LLAVE PRIMARIA | LLAVE FORANEA |

**usuario**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| id | nombres | apellidos | celular | correo |

|  |  |  |  |
| --- | --- | --- | --- |
| fecha\_de\_creacion | fecha\_de\_nacimiento | contraseña | sexo |

**solicitud\_de\_amistad**

|  |  |  |  |
| --- | --- | --- | --- |
| id | solicitante | solicitado | fecha |

**mensaje**

|  |  |  |  |
| --- | --- | --- | --- |
| id | emisor | receptor | fecha |

|  |  |  |
| --- | --- | --- |
| hora | contenido | chat\_id |

**contacto**

|  |  |
| --- | --- |
| contacto\_a\_id | contacto\_b\_id |

**chat**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| id | fecha\_creacion | cantidad\_mensajes | user\_a\_id | user\_b\_id |

* + - 1. DIseño físico de datos
         1. Tabla de volumen

**usuario**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Descripcion** | **Tamaño** | **Nulo** | **Llave** |
| id | Numérico entero | identificador | 8 bytes | no | Primaria |
| nombres | Cadena | Nombres del usuario | 50 bytes | no | no |
| apellidos | Cadena | Apellidos del usuario | 50 bytes | no | no |
| celular | Cadena | Número de celular del usuario | 8 bytes | no | no |
| correo | Cadena | Correo del usuario | 50 bytes | no | no |
| fecha\_de\_creacion | Date | Fecha en la que se registra el usuario | 10 bytes | no | no |
| fecha\_de\_nacimiento | Date | Fecha de nacimiento del usuario | 10 bytes | no | no |
| contraseña | Cadena | Contraseña del usuario | 10 bytes | no | no |

**solicitud\_de\_amistad**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Descripcion** | **Tamaño** | **Nulo** | **Llave** |
| id | Numérico entero | Identificador | 8 bytes | no | Primaria |
| solicitante | Numérico entero | Identificador de usuario solicitante | 8 bytes | no | Foranea |
| solicitado | Numérico entero | Identificador del usuario solicitado | 8 bytes | no | Foranea |
| fecha | Date | Fecha del envio de la solicitud | 10 bytes | no | no |

**mensaje**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Descripcion** | **Tamaño** | **Nulo** | **Llave** |
| id | Numerico entero | Identificador del mensaje | 8 bytes | no | Primaria |
| emisor | Numerico entero | Identificador del usuario emisor del mensaje | 8 bytes | no | Foranea |
| receptor | Numerico entero | Identificador del usuario receptor del mensaje | 8 bytes | no | Foranea |
| fecha | Date | Fecha del envio del mensaje | 10 bytes | no | no |
| hora | Time | Hora del envio del mensaje | 5 bytes | no | no |
| contenido | Text | Contenido del mensaje enviado | 1 Gb | no | no |
| chat\_id | Numerico entero | Id del chat | 8 bytes | no | Foranea |

**contacto**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Descripcion** | **Tamaño** | **Nulo** | **Llave** |
| contacto\_a\_id | Numerico entero | Identificador del contacto | 8 bytes | no | Foranea |
| contacto\_b\_id | Numerico entero | Identificador del contacto | 8 bytes | no | Foranea |

**chat**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Descripcion** | **Tamaño** | **Nulo** | **Llave** |
| id | Numerico entero | Identificador del chat | 8 bytes | no | Primaria |
| fecha\_creacion | Date | Fecha de creacion del chat | 10 bytes | no | no |
| cantidad\_de\_mensajes | Numerico entero | Cantidad de mensajes enviados en el chat | 8 bytes | no | no |
| user\_a\_id | Numerico entero | Identificador del usuario enviador del mensaje | 8 bytes | no | no |
| user\_b\_id | Numerico entero | Identificador del usuario recibidor del mensaje | 8 bytes | no | no |

* + - * 1. Script

CREATE DATABASE mini\_facebook

CREATE TABLE public.usuario

(

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

nombres character varying(50) COLLATE pg\_catalog."default",

apellidos character varying(50) COLLATE pg\_catalog."default",

celular character varying(8) COLLATE pg\_catalog."default",

correo character varying(50) COLLATE pg\_catalog."default",

fecha\_de\_creacion date,

fecha\_de\_nacimiento date,

pass character varying(10) COLLATE pg\_catalog."default",

sexo boolean,

CONSTRAINT usuario\_pkey PRIMARY KEY (id)

)

TABLESPACE pg\_default;

ALTER TABLE public.usuario

OWNER to cadima;

CREATE TABLE public.solicitud\_de\_amistad

(

id bigint NOT NULL DEFAULT nextval('solicitud\_de\_amistad\_seq'::regclass),

solicitante bigint,

solicitado bigint,

fecha date,

CONSTRAINT solicitud\_de\_amistad\_pkey PRIMARY KEY (id),

CONSTRAINT solicitado FOREIGN KEY (solicitado)

REFERENCES public.usuario (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION,

CONSTRAINT solicitante FOREIGN KEY (solicitante)

REFERENCES public.usuario (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

)

TABLESPACE pg\_default;

ALTER TABLE public.solicitud\_de\_amistad

OWNER to cadima;

CREATE TABLE public.mensaje

(

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

emisor bigint,

receptor bigint,

fecha date,

hora time(4) without time zone,

contenido text COLLATE pg\_catalog."default",

chat\_id bigint,

CONSTRAINT mensaje\_pkey PRIMARY KEY (id),

CONSTRAINT mensaje\_chat\_id\_fkey FOREIGN KEY (chat\_id)

REFERENCES public.chat (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION,

CONSTRAINT mensaje\_emisor\_fkey FOREIGN KEY (emisor)

REFERENCES public.usuario (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION,

CONSTRAINT mensaje\_receptor\_fkey FOREIGN KEY (receptor)

REFERENCES public.usuario (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

)

TABLESPACE pg\_default;

ALTER TABLE public.mensaje

OWNER to cadima;

CREATE TABLE public.contacto

(

contacto\_a\_id bigint,

contacto\_b\_id bigint,

CONSTRAINT contacto\_a\_id FOREIGN KEY (contacto\_a\_id)

REFERENCES public.usuario (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION,

CONSTRAINT contacto\_b\_id FOREIGN KEY (contacto\_b\_id)

REFERENCES public.usuario (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

)

TABLESPACE pg\_default;

ALTER TABLE public.contacto

OWNER to cadima;

CREATE TABLE public.chat

(

id bigint NOT NULL,

fecha\_creacion date,

cantidad\_de\_mensajes bigint,

user\_a\_id bigint,

user\_b\_id bigint,

CONSTRAINT chat\_pkey PRIMARY KEY (id),

CONSTRAINT user\_a\_id FOREIGN KEY (user\_a\_id)

REFERENCES public.usuario (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION,

CONSTRAINT user\_b\_id FOREIGN KEY (user\_b\_id)

REFERENCES public.usuario (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

)

TABLESPACE pg\_default;

ALTER TABLE public.chat

OWNER to cadima;

* 1. Implementación
     1. Plataforma de desarrollo de software
        1. Lenguaje de programación

Java 1.8

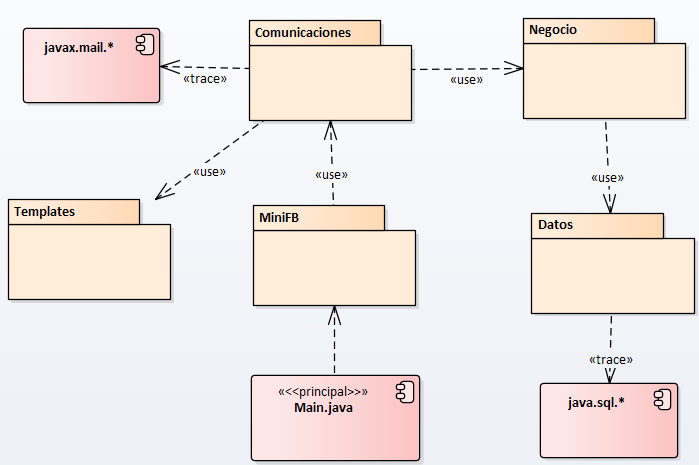
* + - 1. Base de datos

Postgresql 9.6

* + - 1. 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 un medio de comunicación por correos, 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 cliente servidor para la simulación del uso de una red social. Como servidor implementando un programa con Java que permite detectar los correos nuevos y así validar las instrucciones enviadas por el cliente. El cliente será el gestor de correos de cualquier persona la cual le permite enviar un email al servidor.

Para poder identificar los comandos se han creado validadores de cadenas de texto 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 web con sus respectivas validaciones. El protocolo de comunicación utilizado ha sido lo más breve posible para la correcta comprensión del cliente con el contexto de cada uno de los comandos.

1. Conclusiones

Después de haber detectado los resultados positivos y negativos, se ha podido concluir que una red social necesita una forma más potente de comunicación más allá de simples comandos de textos interpretados por un servidor.

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.>