



Universidad de Costa Rica,
Sede del Atlántico,
Recinto de Paraíso.

Documentación Proyecto I de Lenguajes

IF 4101 Lenguajes para Aplicaciones Comerciales

Profesor: Cristian Guillen

I Semestre 2019

Integrantes:

Esteban Barrantes Mora B40894

Andrés Coto Serrano B52265

Alfonso Jiménez Ulloa B63685

Tabla de contenido

Sentencia del problema.	3
Prototipos de las pantallas.	4
Modelo de dominio	9
Modelo de la Base de Datos	9
Para cada funcionalidad determinar las clases requeridas en cada capa. Usar diagramas de clase UML.	9
Modelo Entidad Relación en Tercera Forma Normal (Visual Paradigm - Pata de Gallo).	9
Mejoras y extensiones del proyecto	9
Conclusiones del proyecto	9

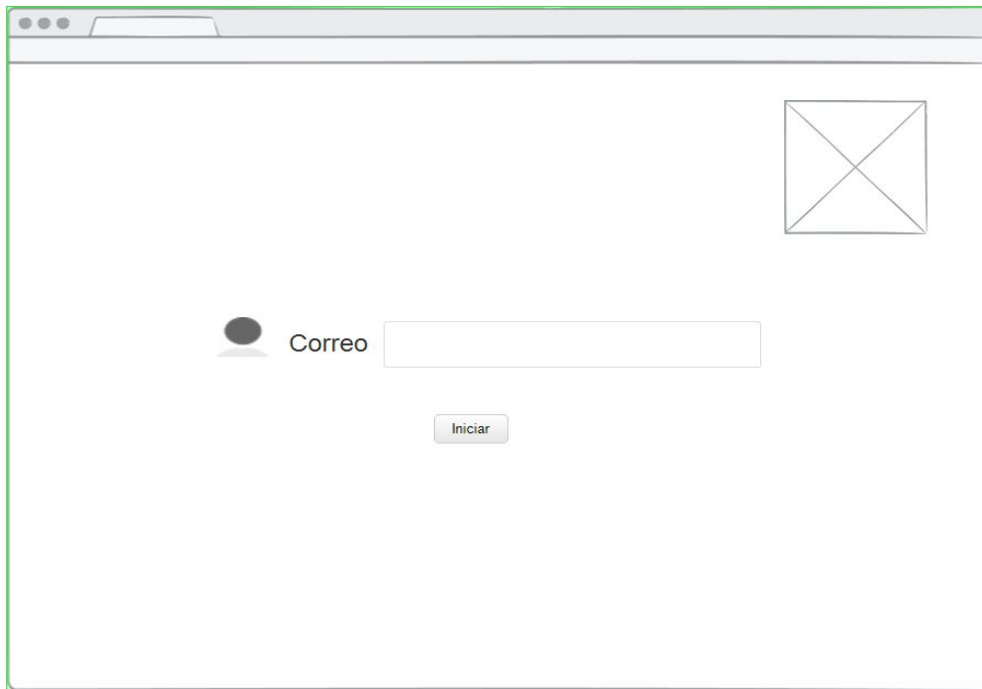
Sentencia del problema.

El problema consiste en realizar una solución informática para crear un chat con ayuda del framework Spring tool Suite 4. El cual permite la creación de grupos para poder chatear entre personas con un máximo de 50 personas por grupo, estas personas serán invitadas mediante un correo electrónico para acceso a la aplicación, además los mensajes tienen un campo máximo de 500 caracteres. Toda información pertinente y necesaria será guardada en una base de datos MySQL.

Por otro lado se tendrá el acceso a la parte administrativa de la aplicación en la cual se desplegará la información general de la aplicación los cuales son una lista de métricas definidas por el usuario, todos los grupos creados en la aplicación, el manejo de usuarios pertenecientes a estos grupos los cuales podrán ser eliminados del chat si el administrador así lo decide.

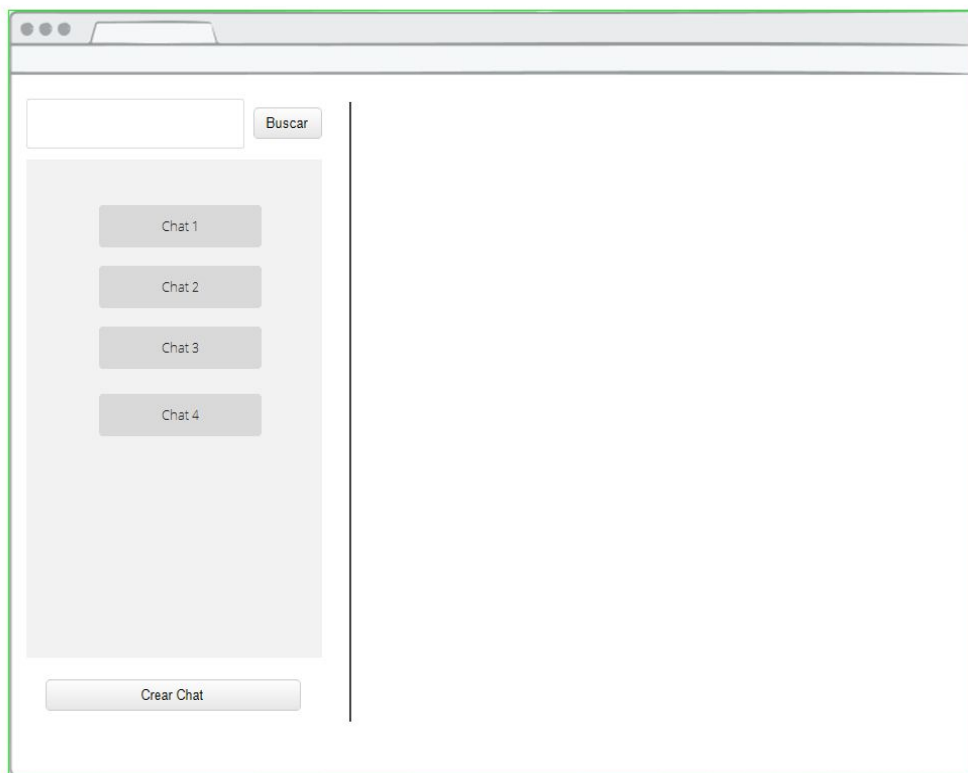
Prototipos de las pantallas.

Inicio de la Aplicación



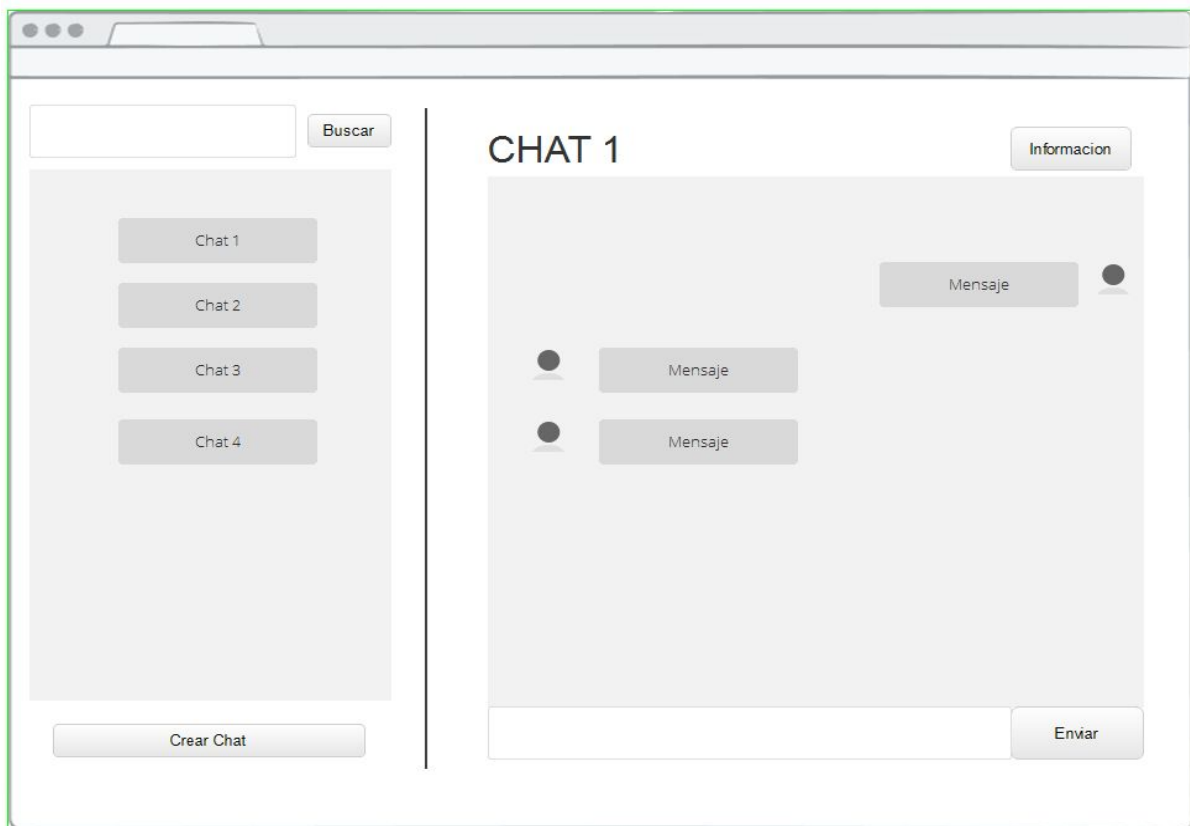
A wireframe of a login screen within a browser window. The window has a standard macOS-style title bar with three window control buttons (red, yellow, green) on the top left. In the top right corner of the page area, there is a square placeholder with an 'X' inside, likely for a logo. Centered on the page is a login form consisting of a small circular icon to the left of the text 'Correo', followed by a rectangular input field. Below the input field is a button labeled 'Iniciar'.

Grupos Pertenecientes al usuario Ingresado

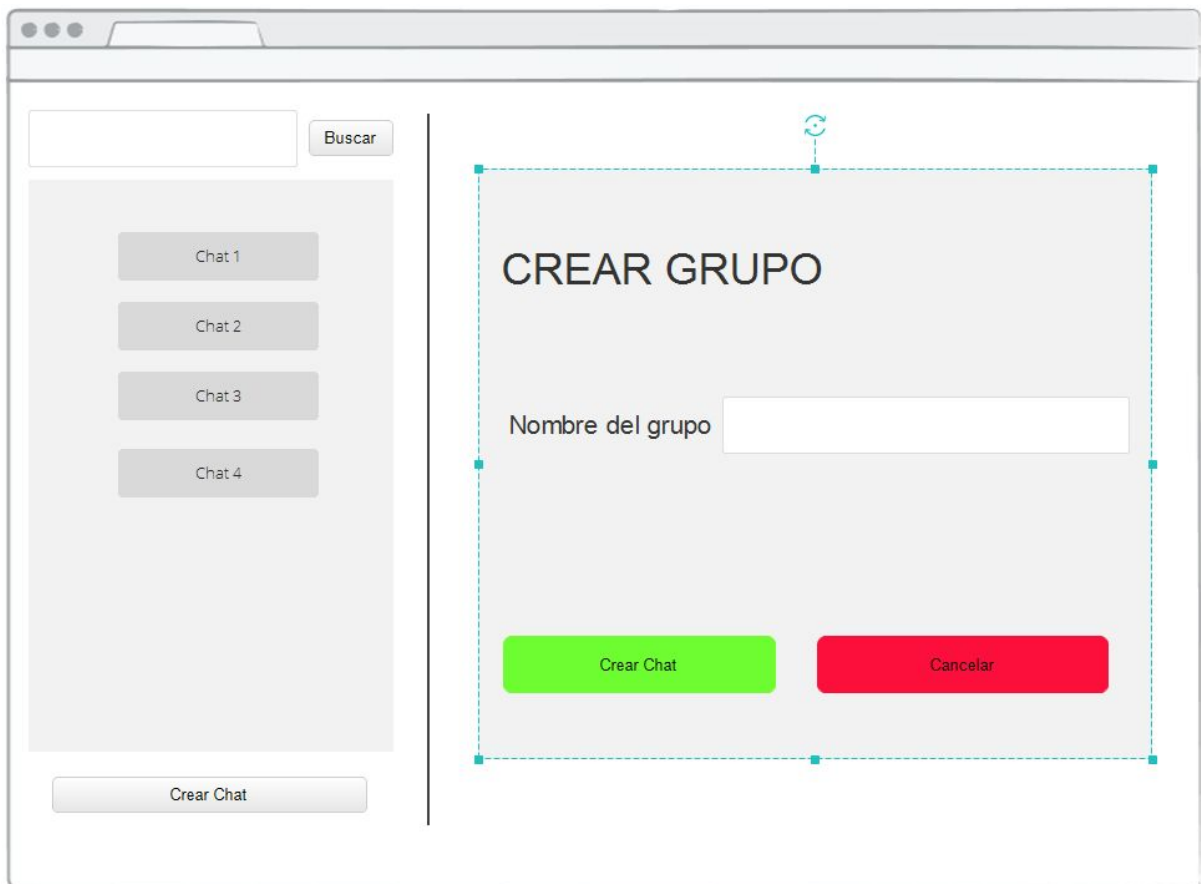


A wireframe of a screen showing chat groups for a logged-in user, presented within a browser window. The window has a title bar with three window control buttons. The main content area is divided into two sections by a vertical line. On the left side, there is a search bar with a 'Buscar' button to its right. Below the search bar is a list of four chat group buttons labeled 'Chat 1', 'Chat 2', 'Chat 3', and 'Chat 4'. At the bottom of this list is a button labeled 'Crear Chat'. The right side of the screen is currently empty, representing the chat interface area.

Chat Perteneciente a un Grupo



Crear Un Grupo



Información del Grupo

CHAT 1

Participantes

Persona 1

Persona 2

Persona 3

Persona 4

Administrador

Persona 1

Persona 2

Persona 3

Persona 4

Creador del Chat

Persona 4

Invitar Participante

Invitación al Grupo

Buscar

Chat 1

Chat 2

Chat 3

Chat 4

Crear Chat

Grupo 1

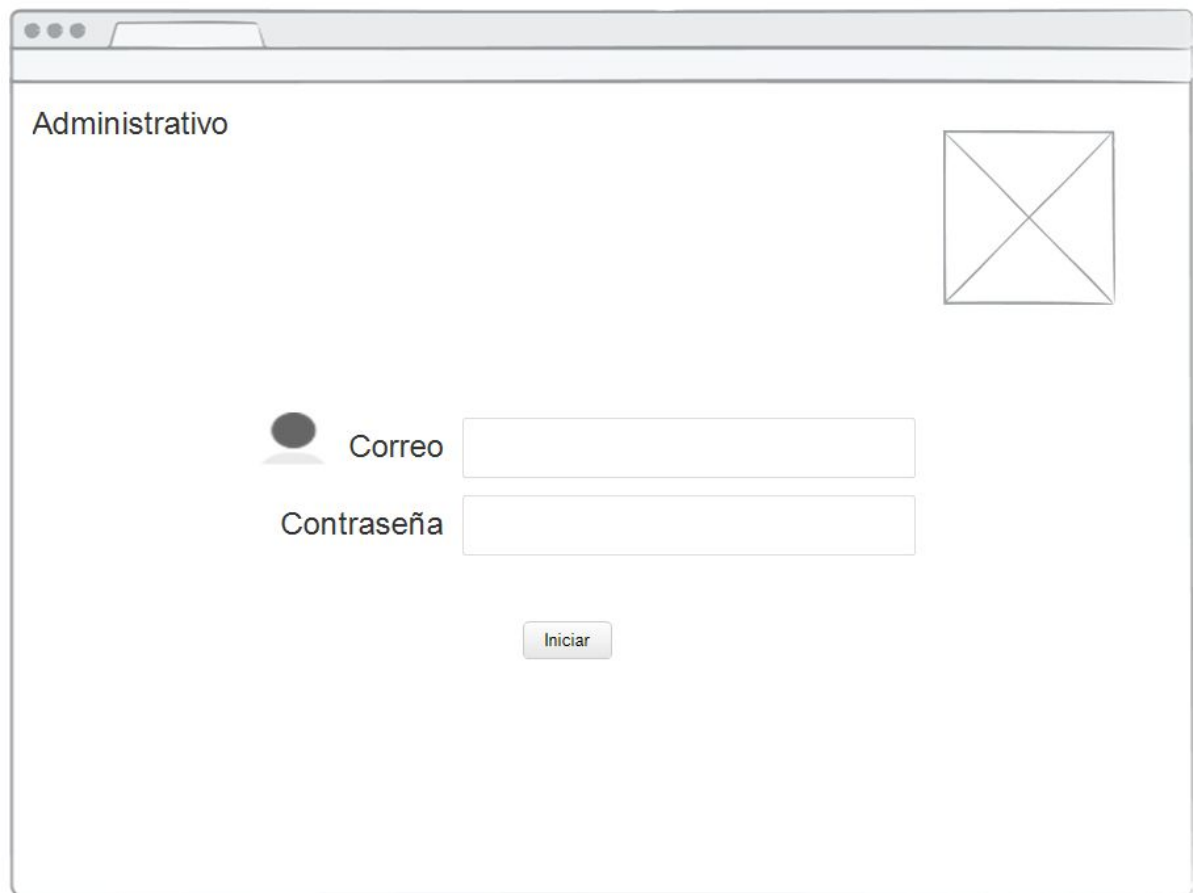
Invitar

Correo de los invitados

Invitar al chat

Cancelar

Ingreso Administrativo



A web browser window displaying the administrative login page. The title bar shows three window control buttons. The page has a header with the word 'Administrativo' on the left and a square placeholder with an 'X' on the right. The main content area features a login form with a user icon, a 'Correo' (Email) label and input field, a 'Contraseña' (Password) label and input field, and an 'Iniciar' (Login) button.

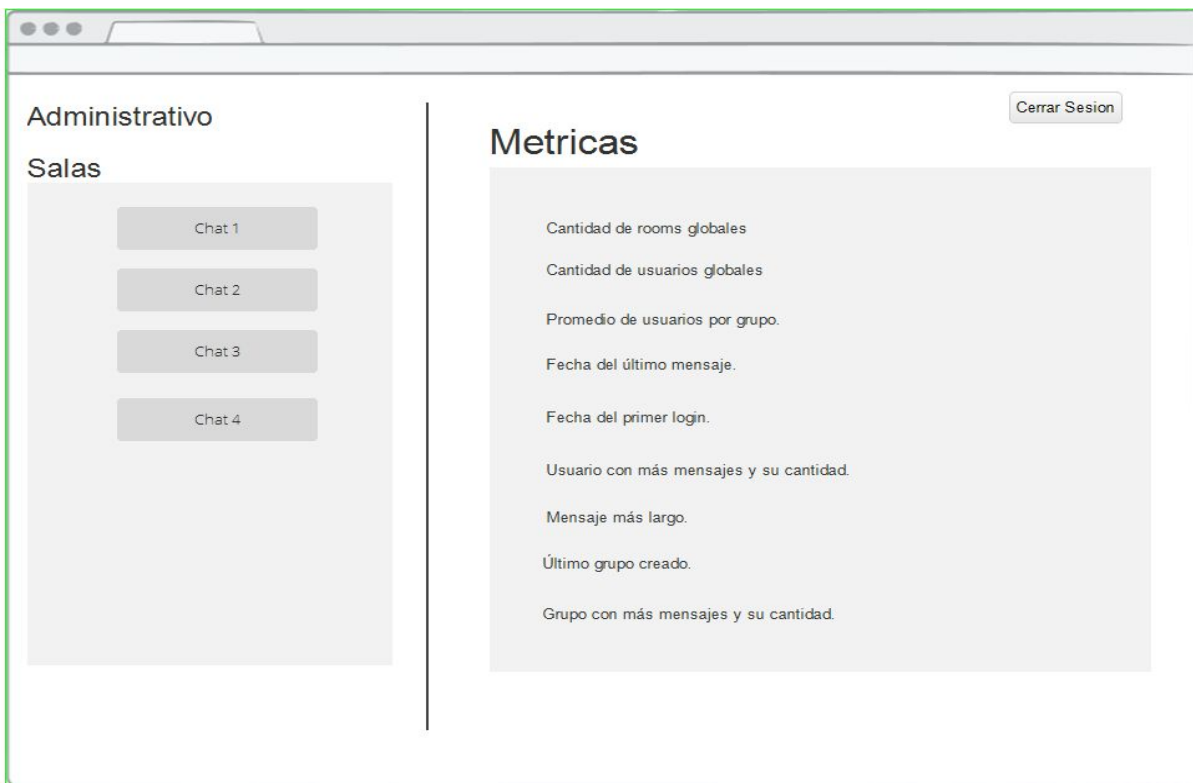
Administrativo

Correo

Contraseña

Iniciar

Pantalla Principal Administrativo



A web browser window displaying the administrative main dashboard. The title bar shows three window control buttons. The page is divided into two main sections. The left section, titled 'Administrativo', contains a 'Salas' (Rooms) sub-header and a list of four chat buttons labeled 'Chat 1', 'Chat 2', 'Chat 3', and 'Chat 4'. The right section, titled 'Metricas' (Metrics), contains a list of nine metrics: 'Cantidad de rooms globales', 'Cantidad de usuarios globales', 'Promedio de usuarios por grupo.', 'Fecha del último mensaje.', 'Fecha del primer login.', 'Usuario con más mensajes y su cantidad.', 'Mensaje más largo.', 'Último grupo creado.', and 'Grupo con más mensajes y su cantidad.'. A 'Cerrar Sesión' (Logout) button is located in the top right corner of the dashboard area.

Administrativo

Salas

Chat 1

Chat 2

Chat 3

Chat 4

Metricas

Cerrar Sesión

Cantidad de rooms globales

Cantidad de usuarios globales

Promedio de usuarios por grupo.

Fecha del último mensaje.

Fecha del primer login.

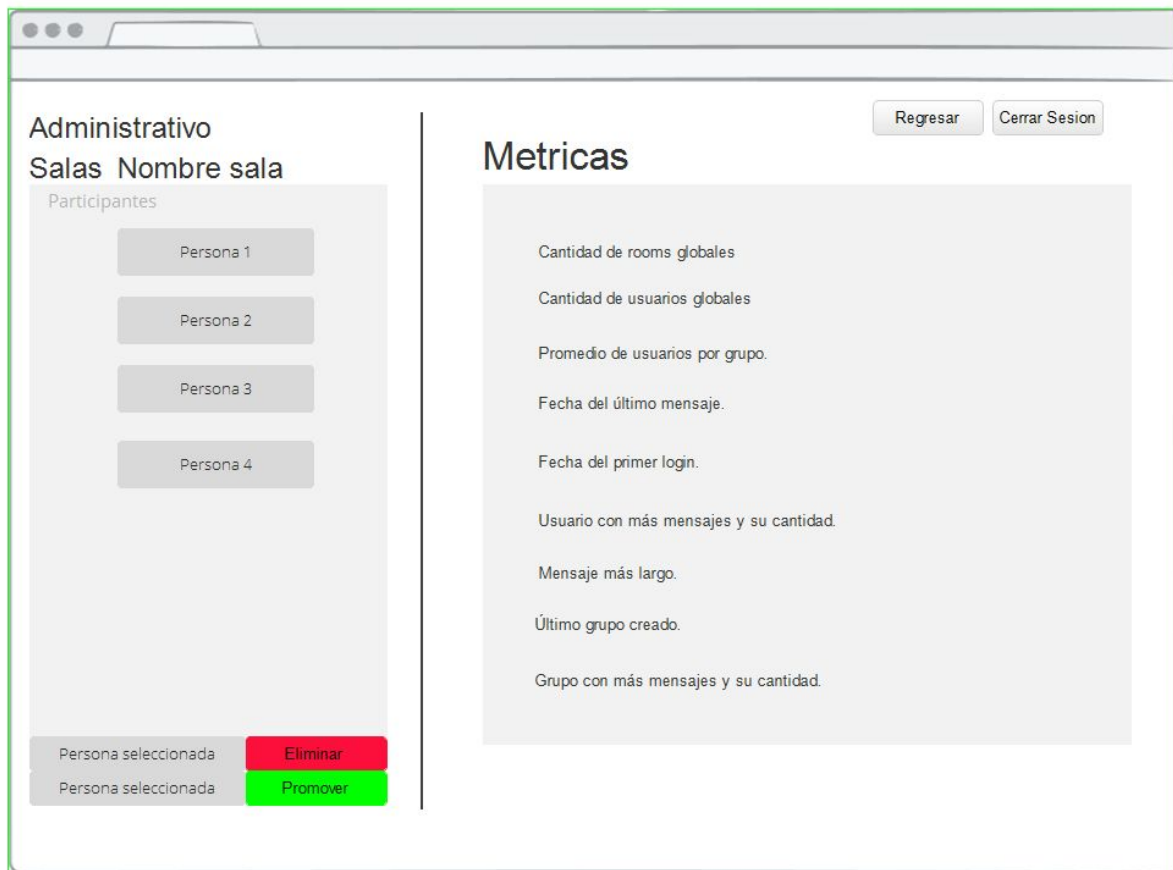
Usuario con más mensajes y su cantidad.

Mensaje más largo.

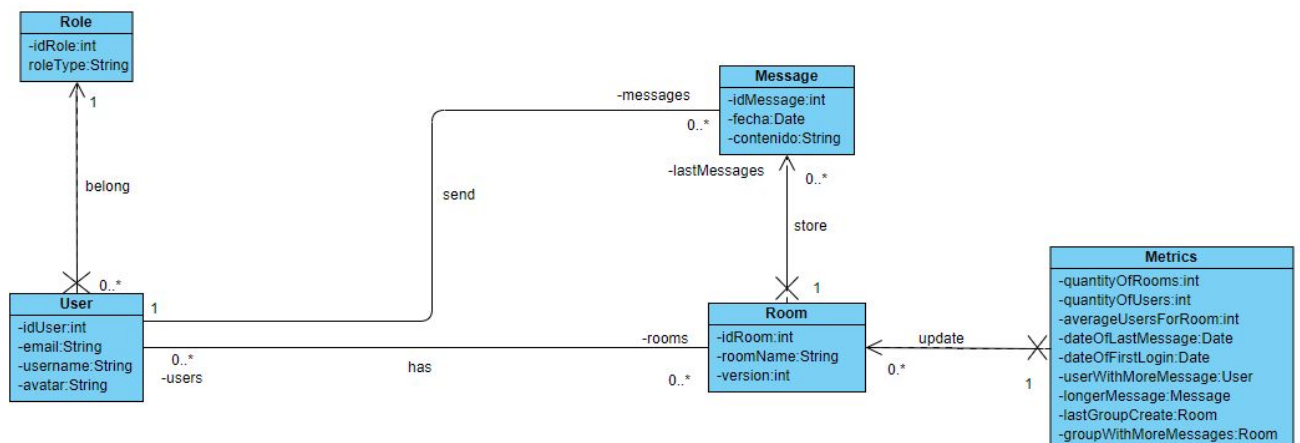
Último grupo creado.

Grupo con más mensajes y su cantidad.

Eliminación o promoción de un Usuario



Modelo de dominio

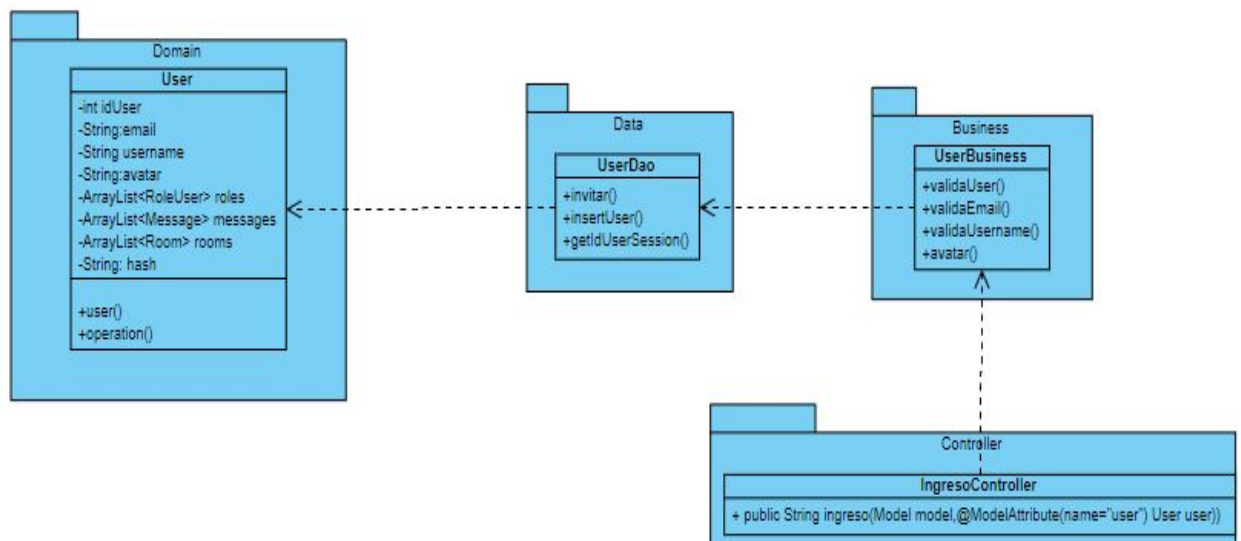


Modelo de la Base de Datos

Para cada funcionalidad determinar las clases requeridas en cada capa. Usar diagramas de clase UML.

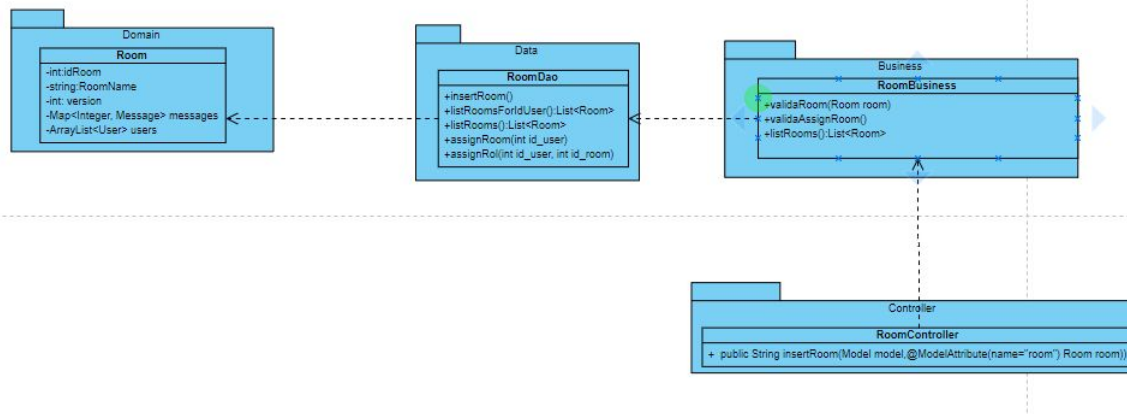
Ingresar usuario

Registrar Usuario



Crear Room

Crear Room



Invitar usuario

Invitar Usuario

```
classDiagram
    class Domain {
        class User {
            -int idUser
            -String email
            -String username
            -String avatar
            -Array<RoleUser> roles
            -Array<Message> messages
            -Array<Room> rooms
            -String hash
            +user()
        }
    }
    class Data {
        class UserDao {
            +invitar()
            +insertUser()
            +getUserSession()
        }
    }
    class Business {
        class UserBusiness {
            +validaUser()
            +validaEmail()
            +validaUsername()
            +avatar()
        }
    }
    class Controller {
        class InvitarController {
            + public String invitar(@RequestBody String emails)
        }
    }
    Domain <--> Data
    Data <--> Business
    Business <--> Controller
```

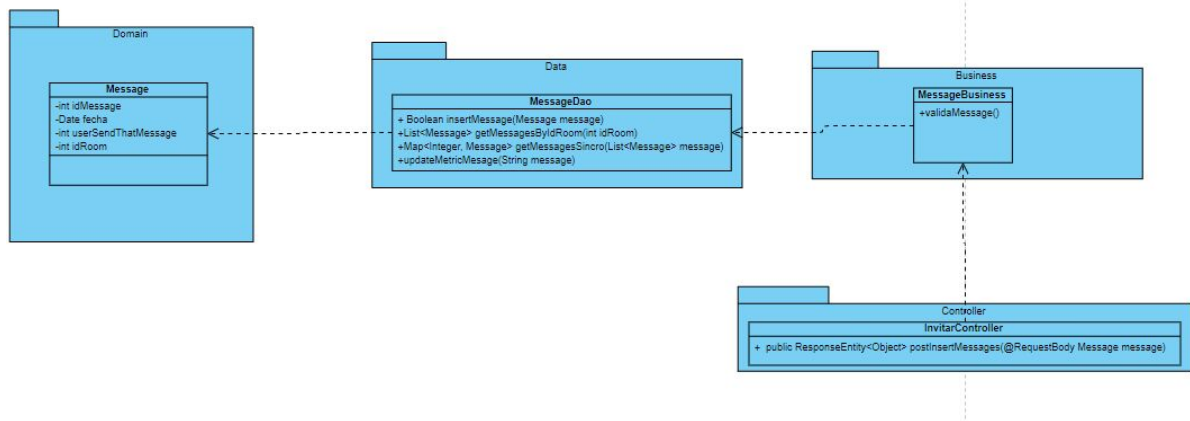
The diagram illustrates the architecture for the 'Invitar Usuario' system, organized into four layers: Domain, Data, Business, and Controller.

- Domain Layer:** Contains the **User** class with attributes: `-int idUser`, `-String email`, `-String username`, `-String avatar`, `-Array<RoleUser> roles`, `-Array<Message> messages`, `-Array<Room> rooms`, and `-String hash`. It also has a `+user()` method.
- Data Layer:** Contains the **UserDao** class with methods: `+invitar()`, `+insertUser()`, and `+getUserSession()`.
- Business Layer:** Contains the **UserBusiness** class with methods: `+validaUser()`, `+validaEmail()`, `+validaUsername()`, and `+avatar()`.
- Controller Layer:** Contains the **InvitarController** class with a public method: `+ public String invitar(@RequestBody String emails)`.

Dependencies are shown with dashed arrows:

- Domain (User) depends on Data (UserDao).
- Data (UserDao) depends on Business (UserBusiness).
- Business (UserBusiness) depends on Controller (InvitarController).

Chatear

[illegible]

Mejoras y extensiones del proyecto

La utilización de frameworks de javascript con el fin de facilitar el desarrollo y promover una mayor investigación con tecnologías más modernas en el consumo de api's.

Por otro lado, una posible extensión podría ser la implementación de web sockets con el fin de que varias computadoras puedan acceder a la misma aplicación incluso se podría implementar una vista para celulares, con el fin de hacer el proyecto más dinámico.

Conclusiones del proyecto.

En conclusión el proyecto enseña a los involucrados una gran cantidad de aspectos tales como: aplicación y consumo de api REST, también diversos estándares de desarrollo, aspectos de seguridad y validaciones que toda aplicación debería tener, así como también se investi.