**Documentación del Sistema Tercera rueda**

**Autores:**

* Kevin Adonis Vargas Benavides

**Contenido**

[1. Objetivo del Proyecto 3](#_Toc85730146)

[2. Descripción General 3](#_Toc85730147)

[3. Especificaciones de software y hardware 3](#_Toc85730148)

[3.1. Selección de software 3](#_Toc85730149)

[3.2. Requisitos de hardware 3](#_Toc85730150)

[4. Descripciones funcionales 4](#_Toc85730151)

[5. Modelo conceptual 4](#_Toc85730152)

[6. Modelo Lógico 4](#_Toc85730153)

[7. Diccionario de Datos 4](#_Toc85730154)

[8. Diseño de ventanas/vistas 5](#_Toc85730155)

# Objetivo del Proyecto

* Proporcionar una herramienta que facilita opciones de entretenimiento a personas de la tercera edad de acuerdo a las características de su perfil.

# Descripción General

El sistema tiene como objetivo lograr una mejora de convivencia para las personas de tercera edad mediante una serie de opciones de entretenimiento de acuerdo a sus gustos y actitudes, permitiendo de este modo a las instituciones de cuidado de personas de tercera edad identificar actitudes y cualidades que permitan una mejor relación y estadía en la residencia. Para ello, estas actitudes y características van a ser almacenadas en un perfil donde se obtendrá una base de datos que nos permita brindarles opciones de entretenimiento de acuerdo a sus capacidades. También el sistema les permitirá a los usuarios favorecer actitudes o habilidades de estas mediante la participación, interacción y comunicación desarrolladas con otros usuarios.

El sistema también promueve el desarrollo de nuevas actitudes en el aprendizaje. Además el sistema tiene la capacidad de poder modificar las actividades favoritas ingresadas por el usuario, para poder tener nuevas instancias de actividades que pueda realizar y así poder tener una mejor comodidad en su estancia. Aparte el sistema también tomara y clasificara a los usuarios mediante sus características físicas es decir a sus capacidades y discapacidades que puedan realizar tanto en solitario como en compañía de otros usuarios que comparten mismo características.

También hay que recalcar que la implementación del sistema podrá ayudar al empleado de dicha casa o residencia para mayores a lo que sería las actividades diarias del usuario donde se podrá también tener una mejora de trabajo del empleado lo cual lo haría más factible y ordenado.

Por último, este sistema esta diseñado para los usuarios que ya hemos dicho y las entidades privadas o publicas pueden utilizarlo a cargo de sus empleados para poder llegar a implementarlo y tener un gran cambio, que es el objetivo. El sistema al momento de implementarse se espera que una gran cantidad de personas puedan utilizarlo.

# Especificaciones de software y hardware

## Selección de software

A partir de la revisión de los datos que deben ser ingresados en la base de datos y de los requisitos de acceso a los datos, se identificaron varios criterios que el software de la base de datos debe cumplir. Los requisitos mínimos para el software son:

* Para la implementación de este proyecto procederemos a utilizar MySQL que es un reconocido sistema de base de datos operacional. Por ser uno de los más utilizados en lo que se refiere al diseño y programación de base de datos de tipo relacional.
* Se utilizará Java para el manejo de la interface ya que facilitará el acceso por parte del usuario, así como también el mantenimiento de la base por parte del administrador.
* Sistema Operativo Linux, Windows, ya que, al utilizar los MySQL y Java, estos programas nos permiten que el sistema base de la plataforma en la cual estará la base de datos sea muy abierta a elección ya que cualquiera que esta sea es compatible con los sistemas antes descritos.

En base a estos requisitos, se determina el sistema operativo, el servidor web, el lenguaje de programación y el motor de base de datos.

## Requisitos de hardware

Cantidad de espacio de disco necesario, ya que el servidor de base de datos depende del número de máquinas de su entorno y del tamaño promedio de los archivos y análisis de exploraciones. Se puede calcular según la fórmula siguiente\*:

<El número de sistemas> x 1 MB + 6 GB de espacio de disco inicial

Por ejemplo:

10.000 sistemas

10 000 x 1 MB + 6 GB = 16 GB

100.000 sistemas

100.000 x 1 MB + 6 GB = 106 GB

La cantidad de espacio de disco que es necesario para que los registros de transacciones dependan del número de sistemas de su entorno, así como el número de sistemas de un grupo individual de exploraciones desde el que se procesarán los datos durante la importación.

Para reducir la cantidad de espacio de disco que es necesario para los registros de transacciones, distribuya las exploraciones a lo largo del tiempo de forma que se procesen durante varias importaciones de datos en lugar de una sola.

El tamaño de los registros de transacciones se puede calcular según la fórmula siguiente:

<El número de sistemas> x 1,2 MB + <el número de sistemas es el grupo de exploración de mayor tamaño> x 1,2 MB + 17 GB de espacio de disco inicial

Por ejemplo:

10.000 sistemas y 10.000 resultados de exploración

10 000 x 1.2 MB + 10 000 x 1.2 MB + 17 GB = 41 GB

100.000 sistemas y 15.000 resultados de exploración

100 000 x 1.2 MB + 15 000 x 1.2 MB + 17 GB = 155 GB

Comentarios tempdb base de datos que se utiliza para almacenar y gestionar objetos temporales.

El tamaño de tempdb se puede calcular según la siguiente fórmula:

<Número de sistemas> x 0,5 MB

Por ejemplo:

15.000 sistemas

15.000 x 0,5 MB = 7,5 GB

# Descripciones funcionales

**Proceso 1:** Buscar afinidad

* **Descripción:** Este proceso nos permite buscar la relación de un usuario con un grupo que tiene una actividad en común o afin en el sistema de acuerdo a un usuario ingresado (id\_usuario).
* **Nota:** Se validará la existencia de un usuario
* **Entrada:** id\_usuario.
* **Salida:** El sistema mostrara el grupo al que este usuario es afin y su actividad.

**Proceso 2:** Buscar Usuario por grupos

* **Descripción:** Este proceso nos permitirá buscar usuarios que formen parte de un grupo en el sistema de acuerdo a los parámetros ingresados.
* **Nota:** Se validará la existencia de un grupo.
* **Entrada:** nombre\_grupo.
* **Salida:** En el caso de encontrar coincidencias, el sistema mostrará un listado de los usuarios que forman parte del grupo ingresado caso contrario se mostrara un mensaje de no se encontraron usuarios en el grupo

**Proceso 3:** Crear usuario

* **Descripción:** Este proceso nos permitirá crear un usuario el cual sera añadido a la base de datos de acuerdo a los parámetros ingresados (id\_usuarios, nombres, apellidos, fecha\_nac, sexo, discapacidades, telefono).
* **Nota:** Se validará que el usuario no se encuentre ya registrado
* **Entrada:** id\_usuarios, nombres, apellidos, fecha\_nac, sexo, discapacidades, telefono.
* **Salida:** Muestra por pantalla un mensaje que dice que el usuario se agregó a la base de datos correctamente.

**Proceso 4:** Eliminar usuario

* **Descripción:** Este proceso nos permitirá eliminar a un usuario del sistema (id\_usuario).
* **Nota:** Se validará que el usuario exista en la base de datos.
* **Entrada:** id\_usuario.
* **Salida:** Muestra por pantalla un mensaje que dice que el usuario ha sido removido completamente de la base de datos.

**Proceso 5:** Modificar usuario

* **Descripción:** Este proceso nos permitirá modificar a un usuario del sistema de acuerdo a los parámetros ingresados (id\_usuarios, nombres, apellidos, fecha\_nac, sexo, discapacidades, telefono).
* **Nota:** Se validará que el usuario exista previamente.
* **Entrada:** id\_usuarios, nombres, apellidos, fecha\_nac, sexo, discapacidades, telefono.
* **Salida:** Muestra por pantalla un mensaje que dice que el usuario ha sido modificado correctamente en la base de datos.

**Proceso 6:** Crear Familiar\_usuario

* **Descripción:** Este proceso nos permitirá crear un familiar del usuario el cual será añadido a la base de datos de acuerdo a los parámetros ingresados (id\_usuario, nombre\_familiar, apellido\_familiar,dirección\_familiar,contacto\_familiar).
* **Nota:** Se validará que el familiar del usuario no se encuentre ya registrado
* **Entrada:** id\_usuario, nombre\_familiar, apellido\_familiar, dirección\_familiar, contacto\_familiar.
* **Salida:** Muestra por pantalla un mensaje que dice que el familiar del usuario se agregó a la base de datos correctamente.

**Proceso 7:** Eliminar Familiar\_usuario

* **Descripción:** Este proceso nos permitirá eliminar a un usuario del sistema de acuerdo a los parámetros ingresados (id\_usuario,contacto\_familiar).
* **Nota:** Se validará que el familiar del usuario exista en la base de datos.
* **Entrada:** id\_usuario, contacto\_familiar.
* **Salida:** Muestra por pantalla un mensaje que dice que el familiar del usuario ha sido removido completamente de la base de datos.

**Proceso 8:** Modificar Familiar\_usuario

* **Descripción:** Este proceso nos permitirá modificar al familiar de un usuario del sistema de acuerdo a los parámetros ingresados (id\_usuario, nombre\_familiar, apellido\_familiar,dirección\_familiar,contacto\_familiar).
* **Nota:** Se validará que el familiar del usuario exista previamente.
* **Entrada:** id\_usuario, nombre\_familiar, apellido\_familiar, dirección\_familiar, contacto\_familiar.
* **Salida:** Muestra por pantalla un mensaje que dice que el familiar del usuario ha sido modificado correctamente en la base de datos.

**Proceso 9:** Crear actividad

* **Descripción:** Este proceso nos permitirá crear una actividad la cual será añadida a la base de datos de acuerdo a los parámetros ingresados (id\_actividad, nombre, disponibilidad, puntuación, descripción, fecha\_act, calificacion).
* **Nota:** Se validará que la actividad no exista previamente.
* **Entrada:** id\_actividad, nombre, disponibilidad, puntuación, descripción, fecha\_act, calificación.
* **Salida:** Muestra por pantalla un mensaje que dice que la actividad se agregó a la base de datos correctamente.

**Proceso 10:** Modificar actividad

* **Descripción:** Este proceso nos permitirá modificar una actividad del sistema de acuerdo a los parámetros ingresados (id\_actividad, nombre, disponibilidad, puntuación, descripción, fecha\_act, calificacion).
* **Nota:** Se validará que la actividad ya exista previamente.
* **Entrada:** id\_actividad, nombre, disponibilidad, puntuación, descripción, fecha\_act, calificacion.
* **Salida:** Muestra por pantalla un mensaje que dice que la actividad ha sido modificada en la base de datos.

**Proceso 11:** Eliminar actividad

* **Descripción:** Este proceso nos permitirá eliminar una actividad del sistema (id\_actividad).
* **Nota:** Se validará que la actividad exista en la base de datos.
* **Entrada:** id\_actividad.
* **Salida:** Muestra por pantalla un mensaje que dice que la actividad ha sido removida completamente de la base de datos.

**Proceso 12:** Poder tomar actividad

* **Descripción:** Este proceso nos mostrara si el usuario puede tomar una actividad a la cual es afin (id\_usuario,id\_actividad)
* **Nota:** Se validará que la calificación muestre que la actividad es afin al usuario y que la disponibilidad le permita tomarla
* **Entrada:** id\_usuario,id\_actividad
* **Salida:** En el caso de encontrar las actividades disponibles mostrara una lista de actividades que puede participar según sus características y mostrara también el numero de acompañantes que puede participar esa actividad.

**Proceso 13:** Crear grupo

* **Descripción:** Este proceso nos permitirá crear un grupo el cual será añadida a la base de datos (id\_grupo, nombre\_grupo, id\_actividad).
* **Nota:** Se validará que el grupo no exista previamente.
* **Entrada:** id\_grupo, nombre\_grupo, id\_actividad.
* **Salida:** Muestra por pantalla un mensaje que dice que el grupo se agregó a la base de datos correctamente.

**Proceso 14:** Modificar grupo

* **Descripción:** Este proceso nos permitirá modificar un grupo del sistema de acuerdo a los parámetros ingresados (id\_grupo, nombre\_grupo, id\_actividad).
* **Nota:** Se validará que el grupo ya exista previamente.
* **Entrada:** id\_grupo, nombre\_grupo, id\_actividad.
* **Salida:** Muestra por pantalla un mensaje que dice que el grupo ha sido modificado en la base de datos.

**Proceso 15:** Eliminar grupo

* **Descripción:** Este proceso nos permitirá eliminar un grupo del sistema (id\_grupo).
* **Nota:** Se validará que la actividad exista en la base de datos.
* **Entrada:** id\_actividad.
* **Salida:** Muestra por pantalla un mensaje que dice que la actividad ha sido removida completamente de la base de datos.

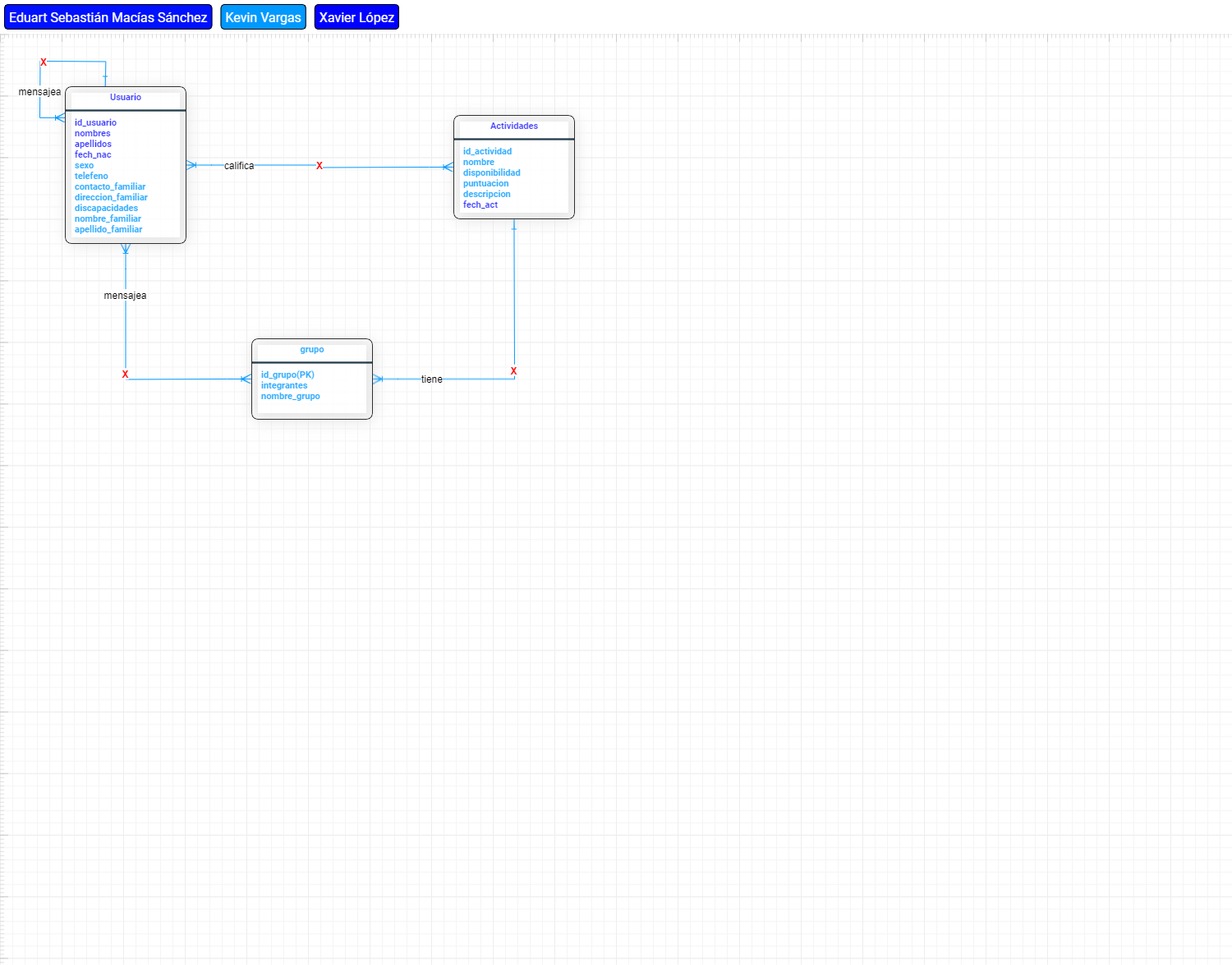
**Proceso 16:** Crear mensaje

* **Descripción:** Este proceso nos permitirá crear un mensaje el cual es enviado hacia otro usuario (id\_mensaje, emisor, receptor\_Usuario, receptor\_grupo, mensaje).
* **Nota:** Se validará que el mensaje tenga al menos 1 carácter
* **Entrada:** id\_mensaje, emisor, receptor\_Usuario, receptor\_grupo, mensaje.
* **Salida:** Muestra por pantalla un mensaje que dice “mensaje enviado”.

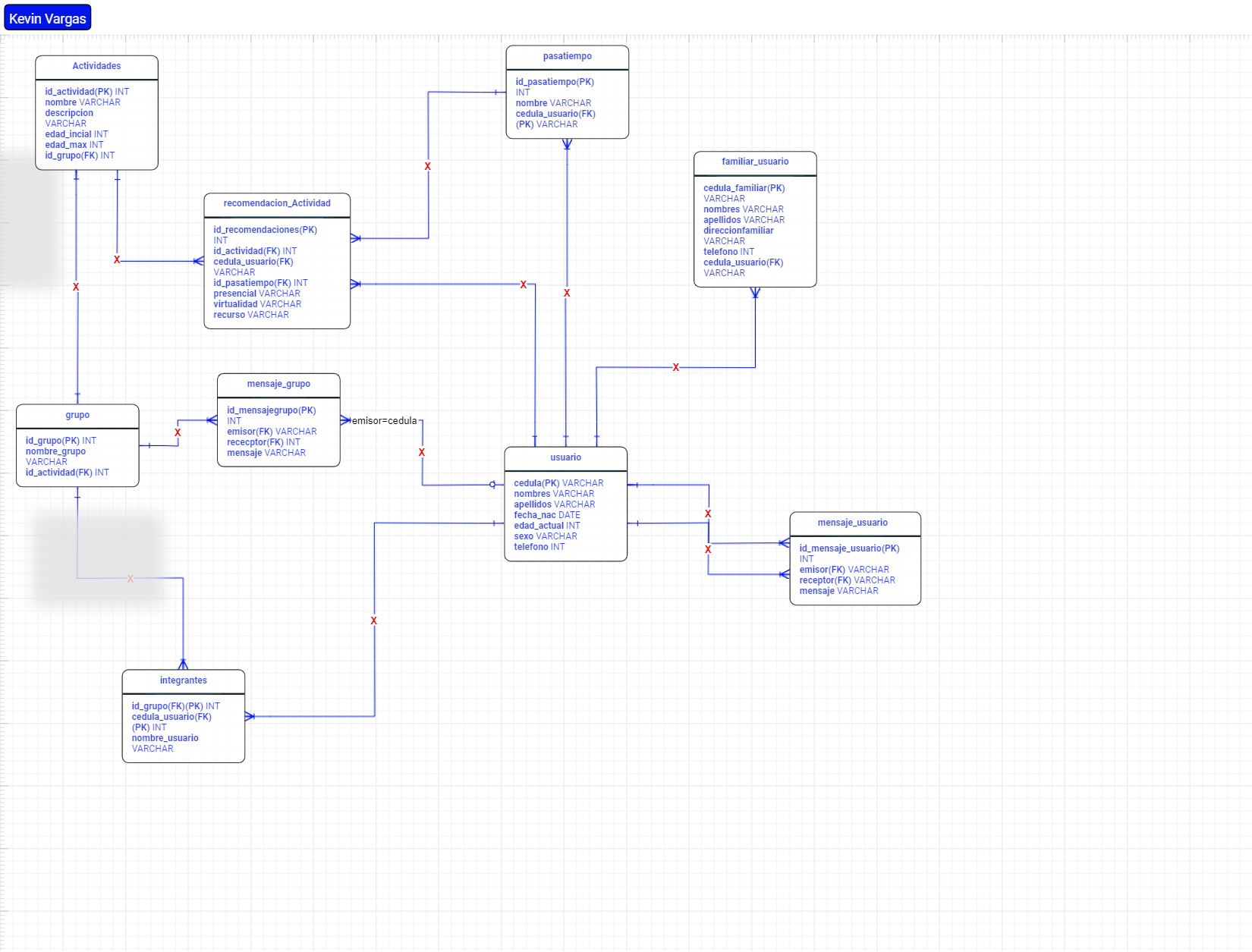
**Proceso 17:** Eliminar mensaje

* **Descripción:** Este proceso nos permitirá eliminar a un mensaje que fue enviado a otro usuario(id\_mensaje).
* **Nota:** Se validará que el mensaje a eliminar haya sido enviado .
* **Entrada:** id\_mensaje.
* **Salida:** Muestra por pantalla un mensaje que dice “mensaje eliminado”.

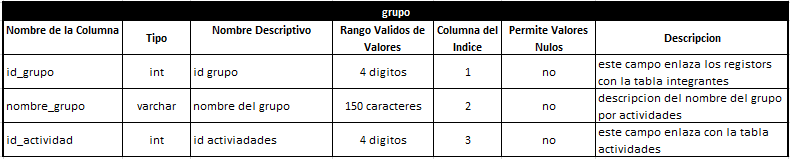
# Modelo conceptual

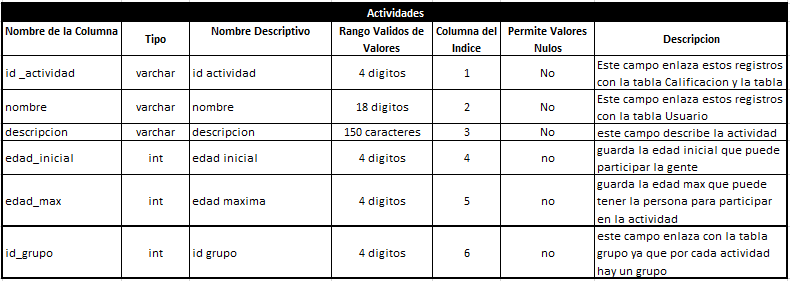


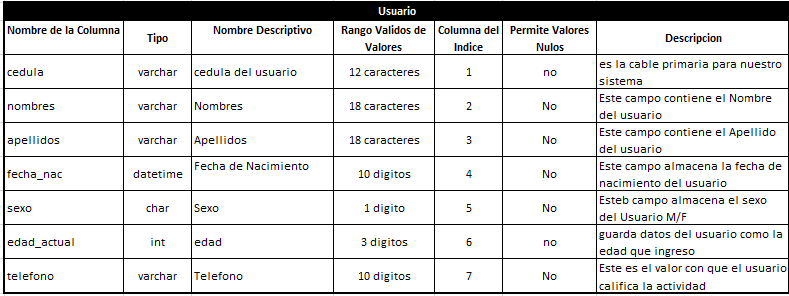
# Modelo Lógico

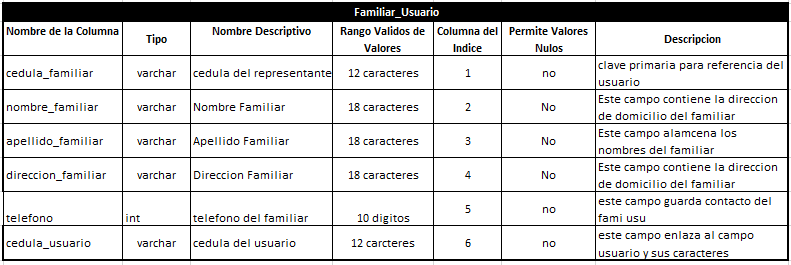


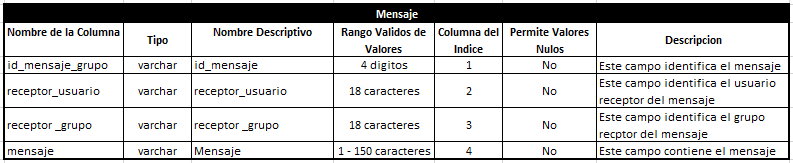
# Diccionario de Datos

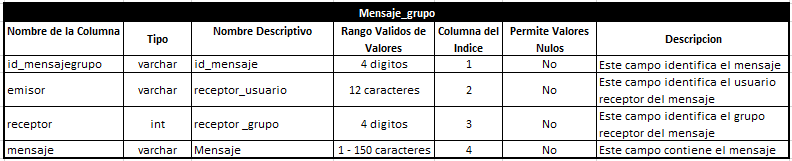


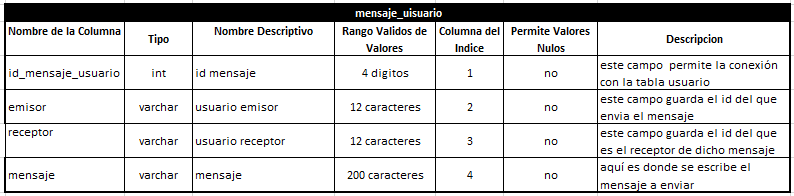


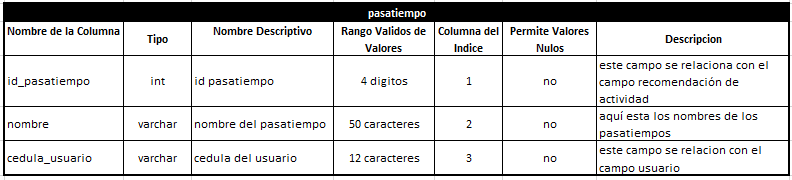




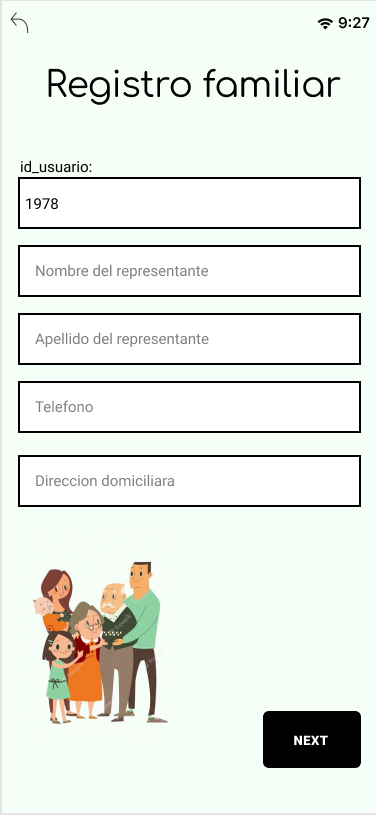






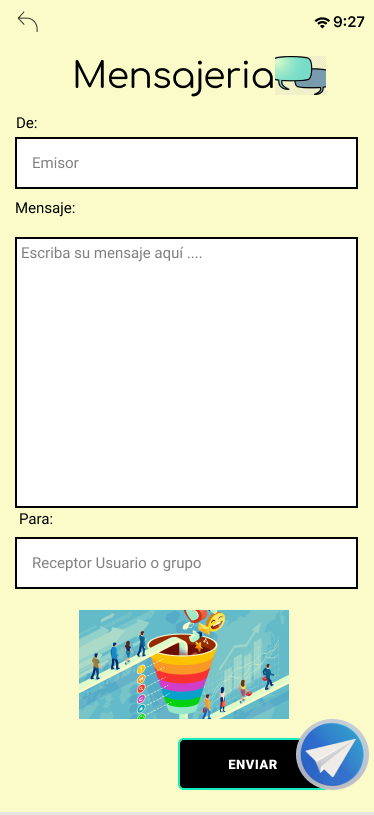
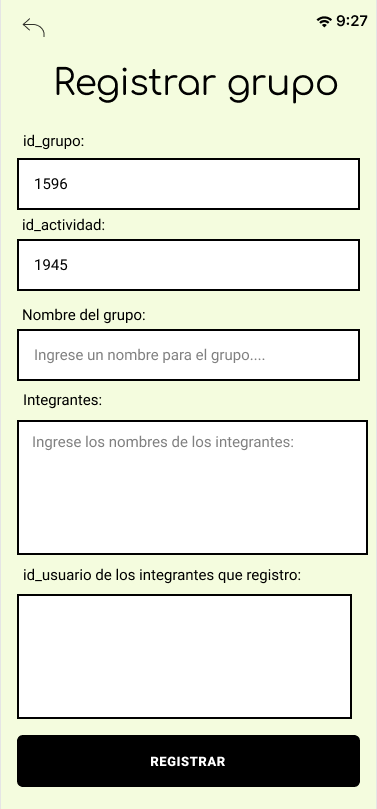


# Diseño de ventanas/vistas



AÑADIR





# Referencias

[Agregar las tres referencias de los sistemas similares revisados]