**Documentación del proceso de desarrollo del software LoginApp**

**Nombre del software que se va a desarrollar: LoginApp**

**Tipo**: Aplicación web

**Encargado por**: Cliente

**Encargado a:** Juansedev2, desarrollador web

**Requerimientos**

Se requiere desarrollar una aplicación que permita el registro e inicio de sesión de usuarios, donde se manejara la siguiente información:

* Nombres y apellidos del usuario
* Correo
* Contraseña
* Rol (especificado más abajo)
* Género
* Edad
* Documento de identificación (tipo y número)
* Fecha de registro del usuario

El sistema debe contar con dos tipos de roles:

* Administrador: Este tipo de usuario debe de poder consultar, crear, actualizar y eliminar cualquier usuario del sistema, a la vez que debe de poder también modificar su propia información de manera completa.
* General: Este tipo de usuario solo puede consultar su información y actualizar su correo y contraseña.

También se necesita que el sitio web tenga vistas de información sobre el mismo, es decir una página acerca del sitio web y de contacto para los usuarios generales que requieran de contactarse con los administradores. Cualquier usuario podría ver estas páginas, sin embargo, si necesitará iniciar sesión para acceder al mismo y hacer uso.

**Restricciones y seguridad**

* Ningún usuario general debe ver información de otros usuarios y tampoco deberá modificar su información más que su correo y contraseña propios. En caso de que olvide sus credenciales, tendrá que enviar un correo a un administrador del sitio web y que el mismo realice la modificación.
* El usuario administrador es el único que puede ver información de otros usuarios y modificarla, a la vez que puede modificar los datos de su propio usuario (excepto el rol).
* El usuario general solo puede ver las opciones de cerrar sesión y ver datos personales, donde en esta podrá actualizar su correo y contraseña solamente.
* Debe de existir un usuario administrador auxiliar en caso de que por alguna razón se eliminen a todos los usuarios administradores o el principal pierda acceso a su cuenta por alguna razón, pero este solo podrá ser compartido con el administrador principal del sitio web (sus credenciales), para que este pueda acceder con el mismo y hacer las mismas funciones que un administrador.
* Las contraseñas de todos los usuarios deben estar cifradas.
* Todo usuario nuevo que quiera registrarse deberá hablar con el administrador de la aplicación para que pueda crear su usuario.

Toda la información de los usuarios debe de estar almacenada en una base de datos y debe estar acorde para guardar solo la información especificada en los requerimientos, además de que no puede haber alguna manera de que los usuarios directamente accedan a la base de datos y puedan manipularla.

En cuanto a las interfaces gráficas, estas serán presentadas al desarrollador luego de su creación para compartir la idea de cómo podría verse el sitio web, solicitando la adaptación de estas de la manera más fiel posible.

Ante cualquier novedad el desarrollador debe mantener contacto permanente con el cliente para realizar modificaciones de ser necesario.

**Fin de los requerimientos por parte del cliente como reunión inicial.**

**Producto analizado a desarrollar:**

Se concluye mediante los requerimientos que el cliente solicita una aplicación web que permita el registro e inicio de sesión de usuarios, teniendo en cuenta sus roles, restricciones y seguridad. Por lo tanto, se utilizarán tecnologías de desarrollo web tanto para el lado del cliente como el lado del servidor, las cuales en concreto serían:

* **Frontend:**
  + HTML
  + CSS
  + JavaScript
* **Backend**
  + PHP
  + MySQL derivado en MariaDB (Para base de datos)

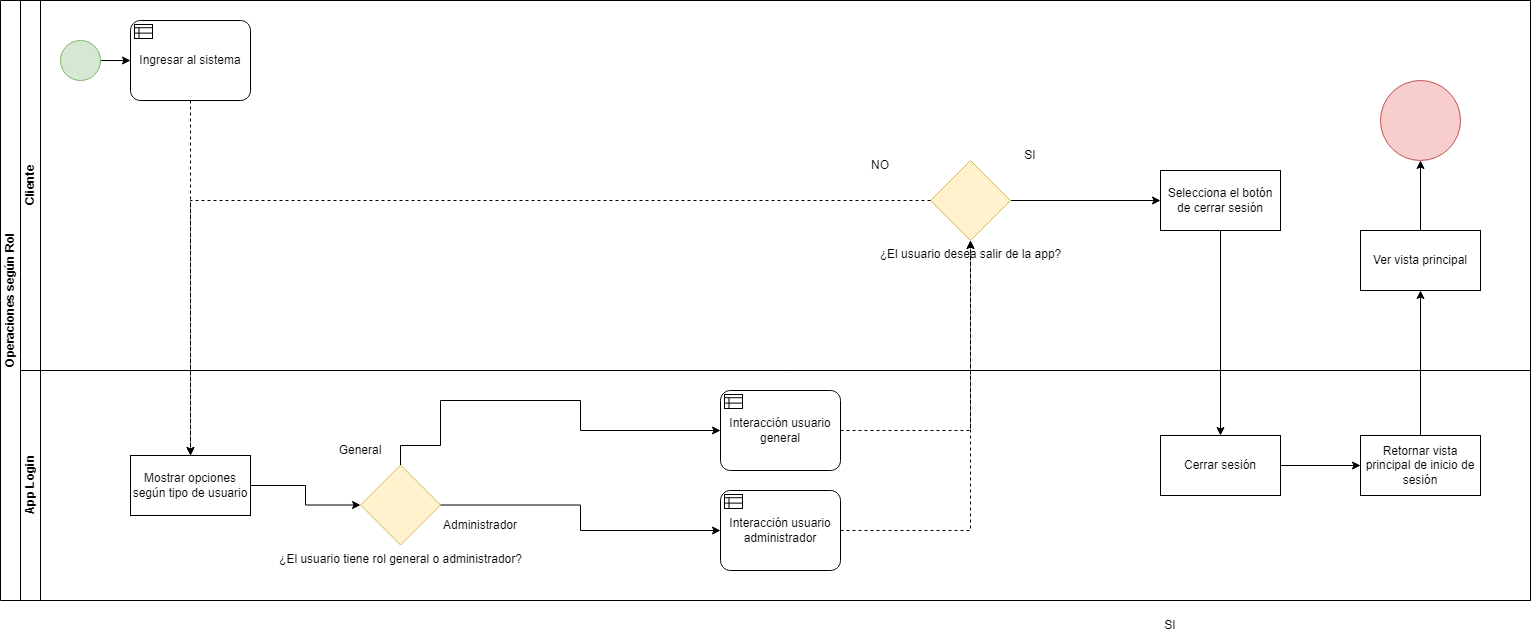
**Diseño del producto**

Se anexa a continuación el desarrollo lógico del desarrollo del software, el cual constará de las siguientes partes:

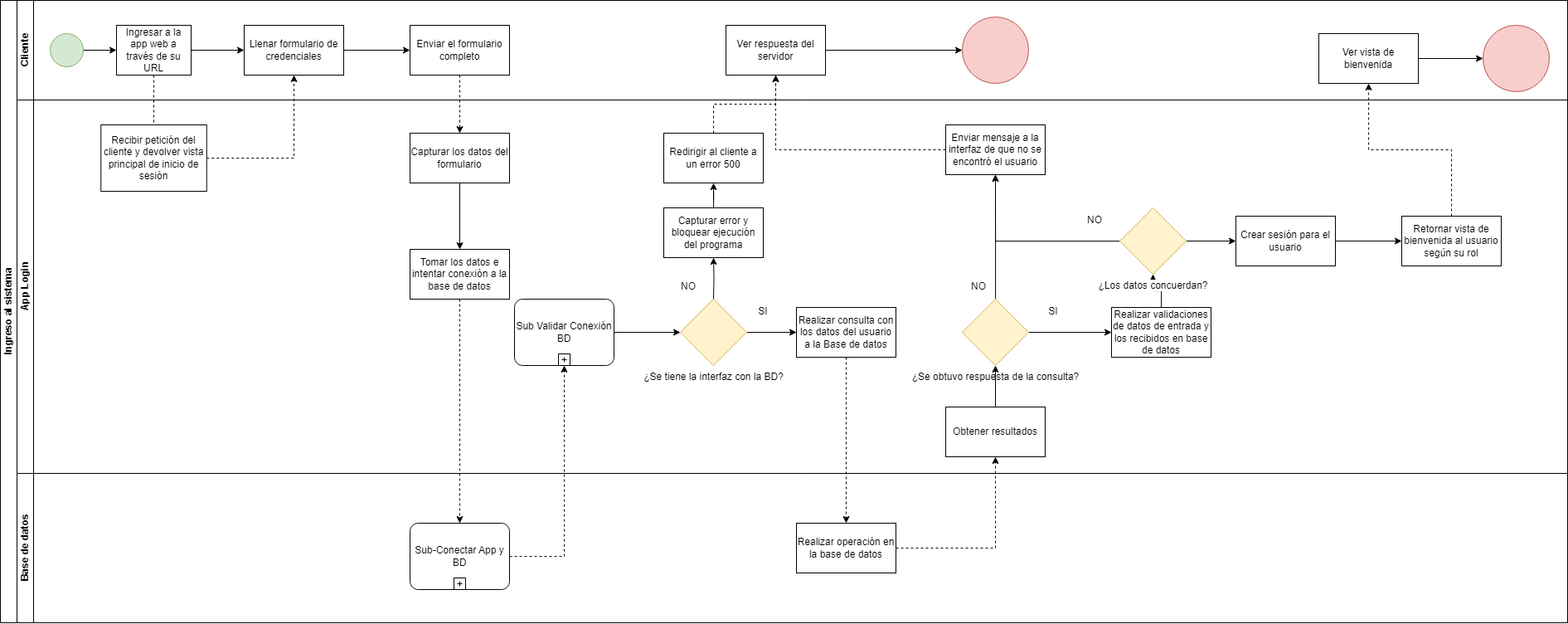
* Diagrama de procesos – lógica del sistema
* Diagrama de paquetes
* Diagrama de clases
* Diseño lógico de la base de datos (MER Y MR)
* Mockups de interfaces gráficas
* Consideraciones finales importantes

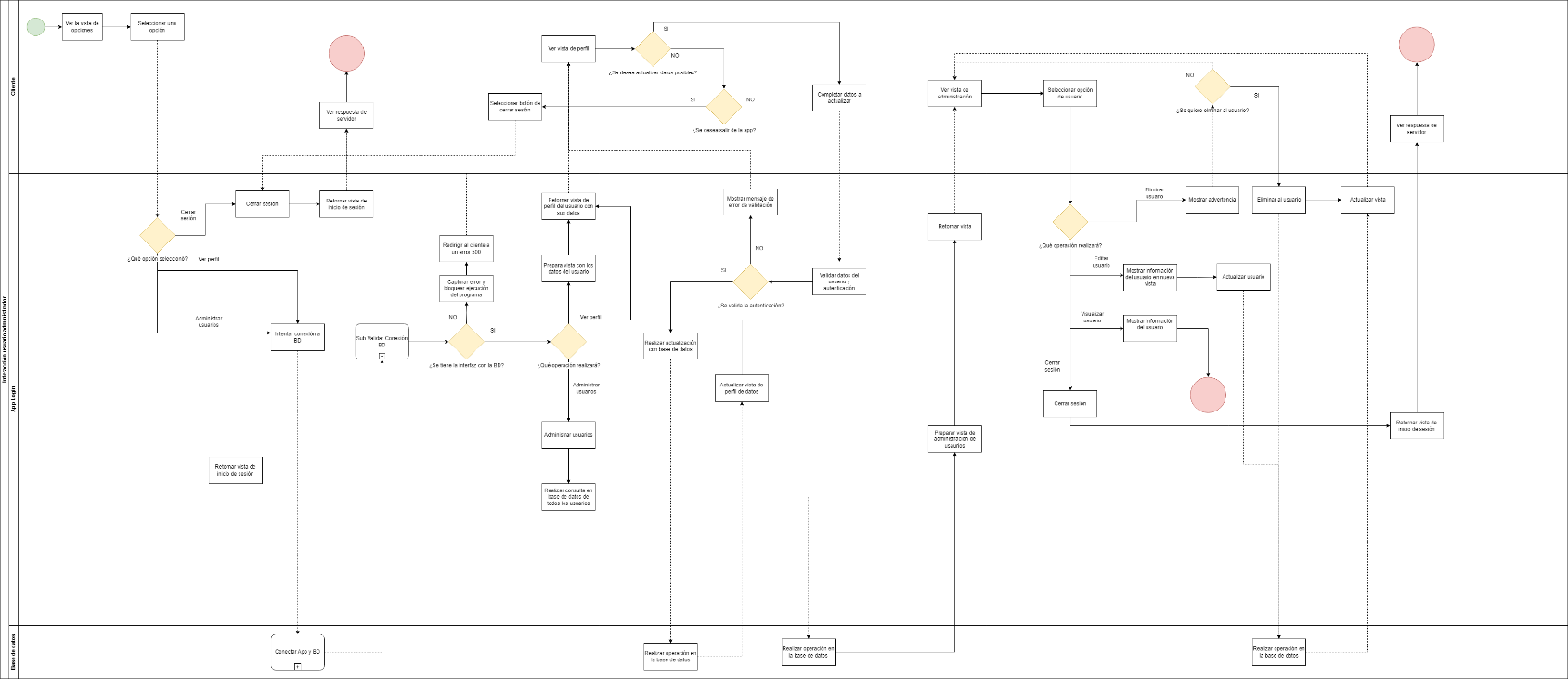
**Diagramas de procesos**

A continuación, se anexan las imágenes de los procesos diagramados de la lógica del sistema y sus procesos (si no pueden visualizarse correctamente, por favor dirigirse al directorio de “Diagramas img”, luego “Diagramas de procesos” y buscarlos por su respectivo título referenciado acá mismo).

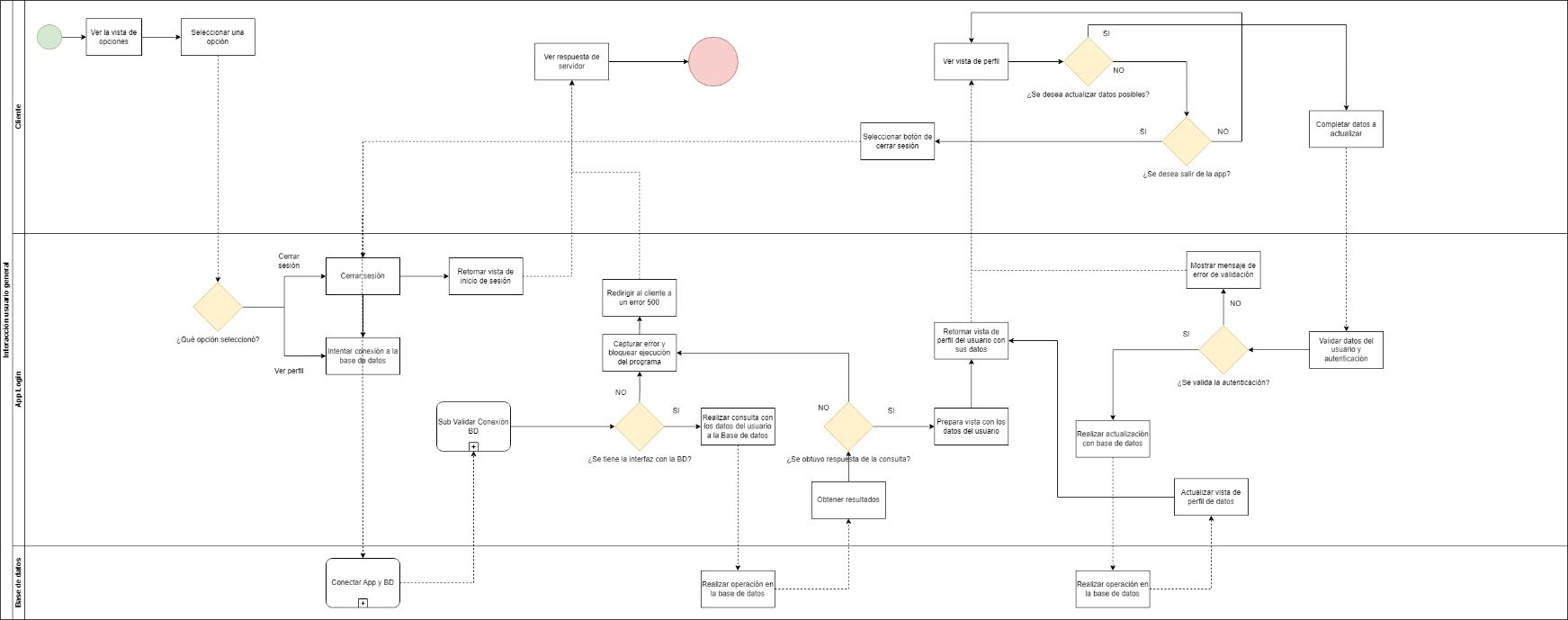
**Diagrama número 1: Interacción con la app**

**Diagrama número 2: Ingreso al sistema**

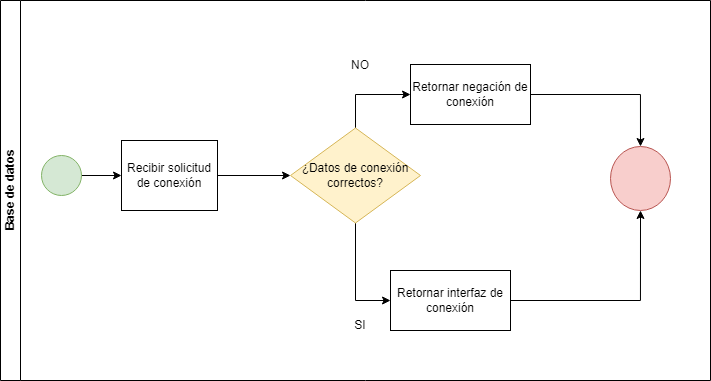
****

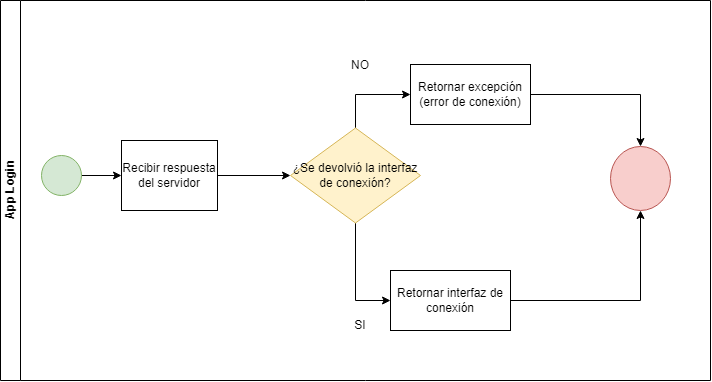
**Diagrama número 3: Interacción usuario administrador**

**Diagrama número 4: Interacción usuario general**

****

**Subprocesos:**

**Diagrama número 5: Sub Conectar App y BD**

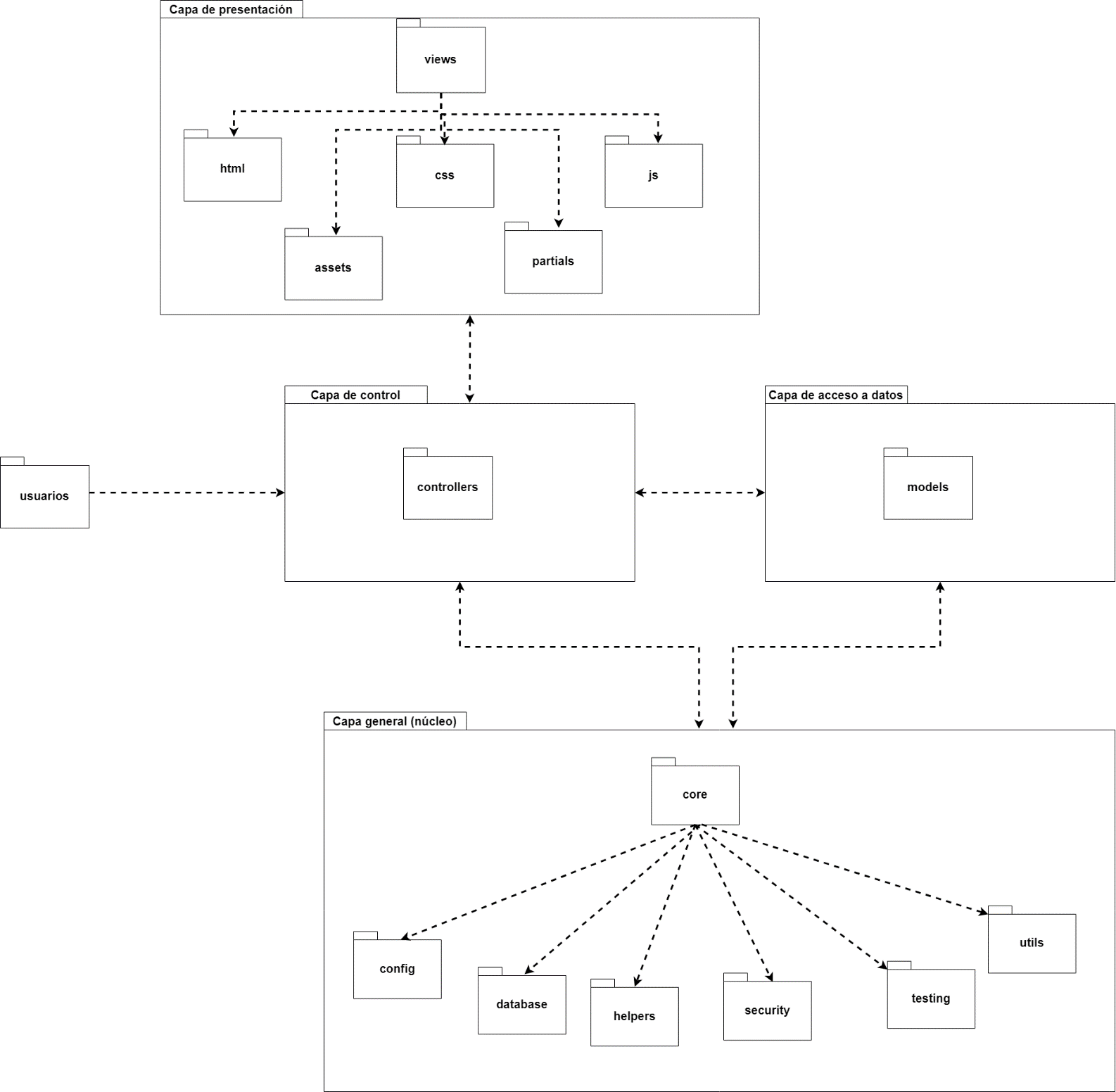
**Diagrama número 6: Sub Validar Conexión BD**

En este punto de la documentación, se tiene claro la lógica que debería de cumplir el sistema a desarrollar, así que por consiguiente se procede al siguiente diagrama de paquetes del sistema.

**Diagrama de paquetes**

**Diagrama número 7: Diagrama de paquetes**

Si no pueden visualizarse correctamente, por favor dirigirse al directorio de “Diagramas img” luego “Diagrama de paqeutes” y buscarlos por su respectivo título referenciado acá mismo).



El anterior diagrama de paquetes hace también referencia al patrón arquitectónico de la aplicación, donde se pueden pensar los directorios principales como las capas de la aplicación, es decir, partes específicas que tienen una responsabilidad que en conjunto forman la solución a brindar. Este patrón arquitectónico corresponde al denominado MVC (Modelo Vista Controlador o Model View Controller por sus siglas en inglés), el cual propone separar las aplicaciones en 3 capas de interacción y orden, las cuales en resumen se describen como:

* Capa de presentación o vista: Esta parte de la aplicación está destinada a almacenar y proporcionar la parte gráfica de la aplicación, es decir las interfaces gráficas que ve el cliente y en ella estarán involucrados todos los temas de diseño, forma de presentación de la información y aspecto requerido, a la vez que será el intermediario principal entre lo que ve y quiere realizar el cliente en el sistema.
* Capa de control o controladores: Esta parte de la aplicación está destinada a ser el intermediario entre las peticiones del cliente y el acceso a la información, es decir que estará encargado de responder a esas peticiones, solicitando manipular la información según corresponda lo que pida el cliente, temas de seguridad, protección de la información y conector entre la capa de presentación y la capa de acceso a datos, así que se encargará también de indicar que regresará al cliente como respuesta.
* Capa de acceso a datos o modelos: Esta parte de la aplicación está destinada a cargar con la responsabilidad de acceder a la información que solicita la capa de controlador y con ello hacer las operaciones necesarias sobre la misma, para que a su vez pueda regresarla o no, actualizarla, entre otras operaciones, además de que puede o no acceder a un sistema de información persistente según la lógica del negocio.

Estas tres capas responden a la lógica del negocio en conjunto para cumplir con las actividades que el sistema debe hacer, aunque se debe añadir que en este caso, según el diagrama, también hay una capa llamada “capa de negocios”, la cual en resumen es una abstracción de otras capas o partes de la aplicación que son necesarias, pero como son comunes para cumplir con las tareas de las demás capas de la aplicación, entonces se centralizan para que así, se tenga domino sobre los componentes base para que la aplicación funcione completamente, por ejemplo una capa de configuración general que necesita la aplicación, una capa de seguridad que ofrecerá todas las funcionalidades para tratamiento de la información y protección de la misma, capa de acceso a los datos como a una base de datos, capa de pruebas y demás.

Esto finalmente puede estar sujeto a ampliaciones y/o cambios que de todas forman respetarán la lógica y el sentido del patrón arquitectónico seleccionado.

**Diagrama de clases**

Se presenta a continuación el diagrama de clases de la aplicación (interacción y lógica entre clases de la aplicación), si no pueden visualizarse correctamente, por favor dirigirse al directorio de “Diagramas img”, luego “Diagramas UML” y buscarlos por su respectivo título referenciado acá mismo.

**Diagrama número 8: Diagrama de clases UML de la lógica de la aplicación**

**Diagrama

Descripción generada automáticamente**

**Diseño lógico de la base de datos (MER Y MR)**

A continuación, se presenta el diseño conceptual y lógico de la base de datos, obtenido luego del análisis de los requisitos, el cual comprende los diagramas de MER (Modelo Entidad Relación) como el diseño conceptual de la lógica de la base de datos de la aplicación y el modelo MR (Modelo Relacional) como diseño lógico de la base de datos, el cual ya expone la estructura e interacción de la base de datos en sus componentes llamados tablas.

Sustantivo: Usuario

Adjetivos: Nombres, apellidos, correo, contraseña, \_rol\_, género, edad, documento de identificacion, fecha de registro de usuario.

Roles pueden ser: Administrador, general.

Conclusión:

Tipos de entidad:

Usuario

Atributos y tipo:

Id\_usuario => número entero

Nombres (primer y segundo nombres y apellidos) => cadena de texto

Apellidos => cadena de texto

Correo => cadena de texto

Contraseña => cadena de texto

Rol => número entero

Género => cadena de texto (un solo caracter)

Edad => número entero

Documento de identificacion => cadena de texto

Fecha de registro de usuario => cadena de texto

(Ningún valor puede ser nulo)

Rol

Id\_rol => número entero

Nombre\_rol => cadena de texto

(Ningún valor puede ser nulo)

Tipo de identificación

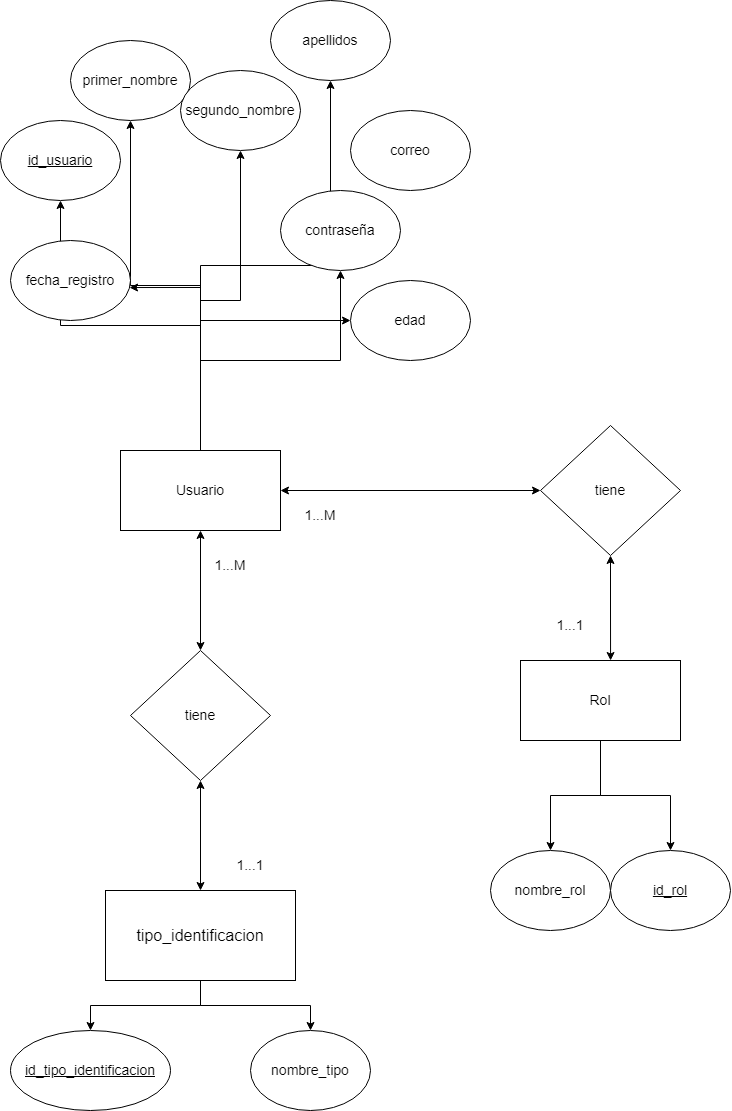
id tipo de identificacion => número entero

nombre del tipo => cadena de texto

Relaciones:

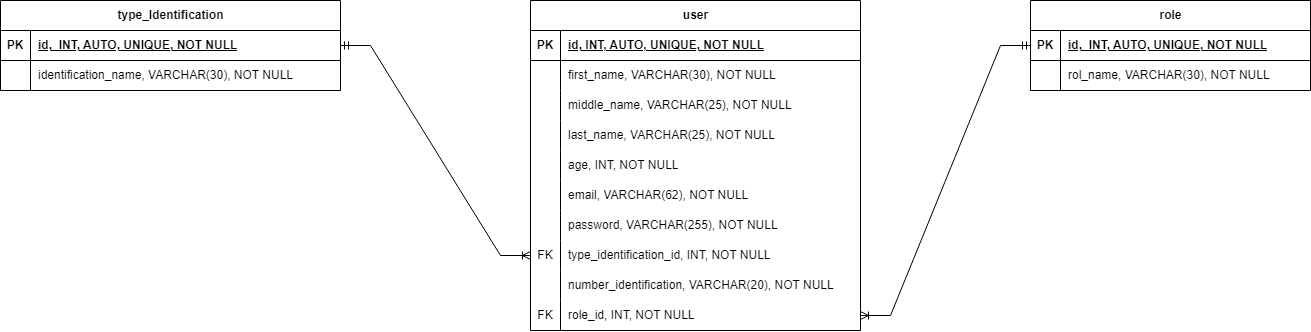
° Un usuario tiene un único rol y un rol puede ser tenido por uno o más usuarios

**Diagrama número 9: Modelo MER (Modelo Entidad Relación)**

****

Se presenta a continuación el modelo entidad relación (MER), si no puede visualizarse correctamente, por favor dirigirse al directorio de “Diagramas img”, luego “Diagramas BD” y buscarlos por su respectivo título referenciado acá mismo.

**Diagrama número 10: Modelo MR (Modelo Relacional)**

****

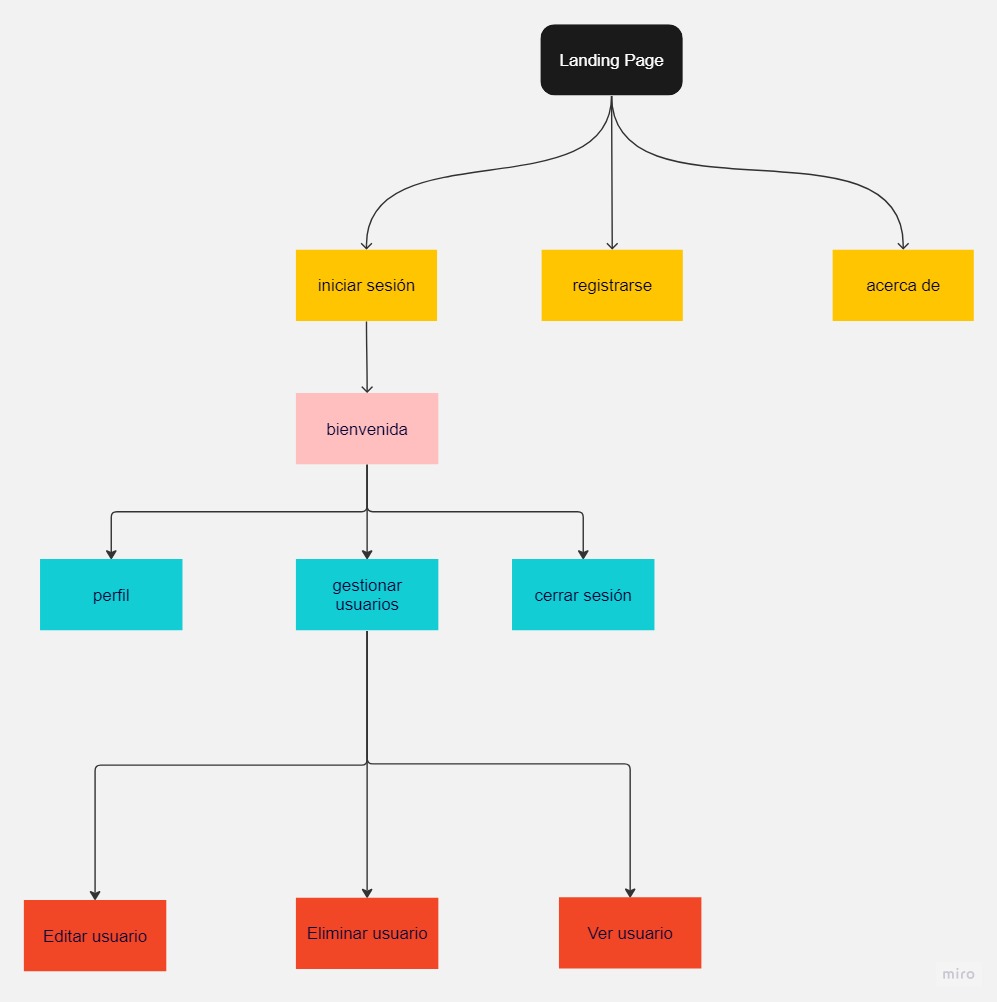
Se presenta a continuación el modelo relacional (MR), si no puede visualizarse correctamente, por favor dirigirse al directorio de “Diagramas img”, luego “Diagramas BD” y buscarlos por su respectivo título referenciado acá mismo.

Luego de finalizado el proceso de análisis y diseño lógico de la base de datos, se procede a realizar su diseño físico, el cual tendrá todo un script de creación añadido al directorio de documentos en el directorio sql, allí estará un script completo que contendrá las instrucciones de la base de datos, así que podrá ser creada en el SGBD que requiera y que sea compatible.

**Mapa de navegación y mockups de interfaces gráfica**

Se realiza a continuación tanto el mapa de navegación de la aplicación web, como el diseño de las ideas de aspecto que contendrán las interfaces gráficas de la aplicación (GUI), para así tener la idea de como debería verse la aplicación por parte del cliente y tener a la mano cualquier idea de corrección y/o mejora. Esta parte contempla tanto el mapa de navegación como los diseños de las GUI.

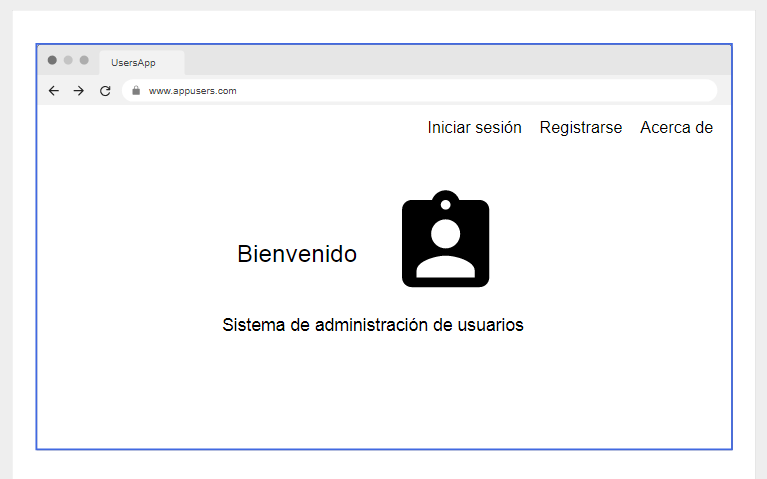
**Diagrama número 11: Mapa de navegación de la aplicación web**

****

Se presenta a continuación el mapa de navegaciín, si no puede visualizarse correctamente, por favor dirigirse al directorio de “Diagramas img”, luego “Mapa navegacion” y buscarlo por su respectivo título referenciado acá mismo.

A continuación, se mostrarán los mockups de las GUI diseñadas para la aplicación web y todas estarán almacenadas en formato imagen en el directorio de “Diagramas img” y luego “Mockups” para buscarlas por su respectivo título referenciado acá mismo por si no se visualiza correctamente.

**Mockup número 1: Landing Page (página principal)**

****

**Mockup número 2: Acerca de**

**Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente**

**Mockup número 3: Registro**

**Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente**

**Mockup número 4: Iniciar sesión**

**Interfaz de usuario gráfica

Descripción generada automáticamente**

**Mockup número 5: Bienvenida usuario (administrador)**

**Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente**

**Mockup número 6: Bienvenida usuario (general)**

**Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente**

**Mockup número 7: Perfil usuario (administrador)**

**Interfaz de usuario gráfica

Descripción generada automáticamente**

**Mockup número 8: Perfil usuario (general)**

**Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente**

**Mockup número 9: Gestionar usuarios**

**Tabla

Descripción generada automáticamente**

**Mockup número 10: Ver usuario**

**Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media**

**Mockup número 11: Editar usuario**

**Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media**

**Consideraciones finales importantes**

En este punto del desarrollo del proyecto, se ha finalizado el diseño lógico de la aplicación web, así que se procederá a realizar la implementación física de la misma, empezando con la creación del proyecto y de la base de datos. Puede presentarse casos de cambios a lo largo de su desarrollo y en caso de que sea así, se documentarán esos cambios, sean en la estructura de la base de datos, la lógica del negocio y la aplicación, las GUI, etc.

Todo el desarrollo de la aplicación para el servidor estará también documentado y seguido por un sistema de control de versiones llamado GIT, el cual tendrá todo el histórico del proceso de desarrollo del proyecto por si se llegara a necesitar.

**Fecha de inicio de desarrollo y finalización:**

**21 de Agosto - …**

**\*Al final:**

**Se entrega el producto desarrollado al cliente y se hace explicación de su uso, se hace validación de su funcionamiento y es recibido con éxito.\***