UNIVERSIDAD GALILEO

FACULTAD DE INGENIERÍA DE SISTEMAS, INFORMÁTICA Y CIENCIAS DE LA COMPUTACIÓN



**Documento de Requerimientos**

EDDY ALEXANDER CABRERA FUENTES - 17005241

KEVIN JOSÉ HERNÁNDEZ MARROQUÍN - 17001095

INGENIERÍA DE SOFTWARE

SEXTO SEMESTRE

GUATEMALA, 20 DE SEPTIEMBRE 2019

**Índice**

1. Prefacio 4

2. Introducción 5

2.1 Objetivo General 5

2.2 Objetivo Específico 5

2.3 Alcance del Producto 5

2.4 Definiciones 6

2.5 Resumen 7

3. Glosario 8

4. Requerimientos de Usuario 9

5. Arquitectura del Sistema 10

6. Requerimientos del Sistema 11

7. Modelos del Sistema 25

7.1 Diagramas de Casos de Uso 25

7.2 Diagramas de Secuencia 27

7.3 Diagrama de Clase 40

8. Evolución del Sistema 41

9. Apéndice 43

**Índice de Figuras**

Figura 1 10

Figura 2 25

Figura 3 26

Figura 4 27

Figura 5 28

Figura 6 29

Figura 7 30

Figura 8 31

Figura 9 32

Figura 10 33

Figura 11 34

Figura 12 35

Figura 13 36

Figura 14 37

Figura 15 38

Figura 16 39

Figura 17 40

**Índice de Tablas**

Tabla 1 11

Tabla 2 12

Tabla 3 13

Tabla 4 14

Tabla 5 15

Tabla 6 16

Tabla 7 17

Tabla 8 18

Tabla 9 19

Tabla 10 20

Tabla 11 21

Tabla 12 22

Tabla 13 23

Tabla 14 24

**Prefacio**

* **Versión 1.0**

Se comienza a realizar el documento de requerimientos, redactando todas las secciones indicadas en la presentación proporcionada en la clase.

El proyecto permite a los conectar a los financistas con emprendedores para desarrollar ideas que pueden tener un buen impacto y que necesitan de recursos para su implementación.

**Introducción**

* **Objetivo General:**

Definir oficialmente el sistema a implementar, dejando plasmado los requerimientos de usuario y sus respectivas traducciones a requerimientos de sistemas para poder comunicar qué funcionalidades, condiciones y restricciones debe cumplir dicho sistema.

* **Objetivo Específico:**

Detallar información adicional relacionado al funcionamiento del sistema.

* **Alcance del Producto**:

Al finalizar el proyecto, les producto provee las siguientes funcionalidades:

* + Permite a los usuarios registrarse y pueden optar por 3 roles distintos (puede escoger 1 o más roles): financista, emprendedor o negociante.
  + Los usuarios pueden configurar su información personal en cualquier momento y pueden cambiar de rol en cualquier momento.
  + Los usuarios con rol de emprendedor pueden publicar ideas de acuerdo con una categoría de ideas que pueden ser vistas por uno o varios financistas y pueden aceptar solicitudes de videochat.
  + Permite a los usuarios con rol de emprendedor ver qué usuarios de tipo financista están interesados en alguna idea que ha publicado.
  + Notifica a los usuarios con rol de emprendedor las solicitudes de videochat pendientes que tiene y puede aceptarlas o rechazarlas.
  + Permite filtrar las ideas publicadas y marcar aquellas que pueden ser consideradas como ‘Spam’ utilizando una inteligencia artificial.
  + Los usuarios con rol de financista pueden ver ideas publicadas por uno o varios emprendedores, pueden añadir dichas ideas a un bookmark que les permite una búsqueda ágil y pueden enviar una solicitud de videochat al autor de una idea para realizar sus respectivos negocios.
  + Los usuarios con rol de negociante pueden publicar recursos que no necesitan para realizar compras, intercambios o negocios con otros usuarios de tipo negociante.
  + Los usuarios con rol de negociante pueden ver recursos publicados de acuerdo con una categoría para realizar compras, intercambios o negocios con otros usuarios de tipo negociante.
* **Definiciones:**
  + **Usuarios con rol de** ***Emprendedor***:

Son aquellos usuarios que se registraron al sistema que poseen el derecho de publicar ideas en la aplicación web (de acuerdo a una categoría), observar detalles de todas las ideas que ha publicado y pueden darles mantenimiento.

* + **Idea:**

Es una noción o representación mental de algo

* + **Usuarios con rol de *Financista:***

Son aquellos usuarios que se registraron al sistema que pueden ver ideas publicadas por uno o varios usuarios con rol de emprendedores (filtrando por medio de una categoría).

* + **Usuarios con rol de *Negociante*:**

Son aquellos usuarios que se registraron al sistema que pueden publicar bienes que no necesitan y ver bienes que necesitan.

* + **Bien:**

Cualquier servicio o recurso que la persona posee o carece.

* + **Administrador:**

Es un tipo de usuario especial que posee privilegios sobre la base de datos y la aplicación web. Puede añadir, modificar o editar categorías y estados de ideas y puede administrar usuarios registrados.

* **Resumen:**

La aplicación web permite que los usuarios se puedan registrar al sistema y puedan escoger los roles que desean poseer.

Los servicios principales y novedosos que proporciona el sistema es que los usuarios pueden realizar videochat entre sí y que pueden publicar fotos y/o videos de acuerdo a un rol que escojan los usuarios.

De acuerdo con los roles, los usuarios tienen a su disposición diferentes servicios.

Los roles que se tienen son:

* + Emprendedor:

Los usuarios pueden publicar ideas y ver qué ideas han publicado a lo largo de todo el tiempo que han estado activos en la aplicación.

* + Financista:

Los usuarios con este rol pueden ver ideas que han sido publicados por emprendedores y pueden filtrar las ideas por categorías. Además, pueden marcar las ideas a un bookmark que les permite agilizar la búsqueda de alguna idea. Y por último, pueden enviar solicitud a un emprendedor para realizar videochat y realizar los negocios que deseen.

* + Negociante:

Los usuarios con este rol pueden publicar bienes que no necesiten y estén dispuestos a hacer negocios, y también pueden buscar bienes que necesitan y realizar los respectivos negocios. También dichos usuarios poseen la funcionalidad de mandar solicitudes de videochat y de publicar fotos y videos del bien que estén ofreciendo.

El objetivo del proyecto es conectar a las personas y que ellos hagan los negocios que tengan que hacer por su cuenta.

**Glosario**

* **Frontend:**

Es la parte de la aplicación web con el cual el cliente puede interactuar.

* **Backend:**

Es la parte de procesamiento que hay detrás en la aplicación.

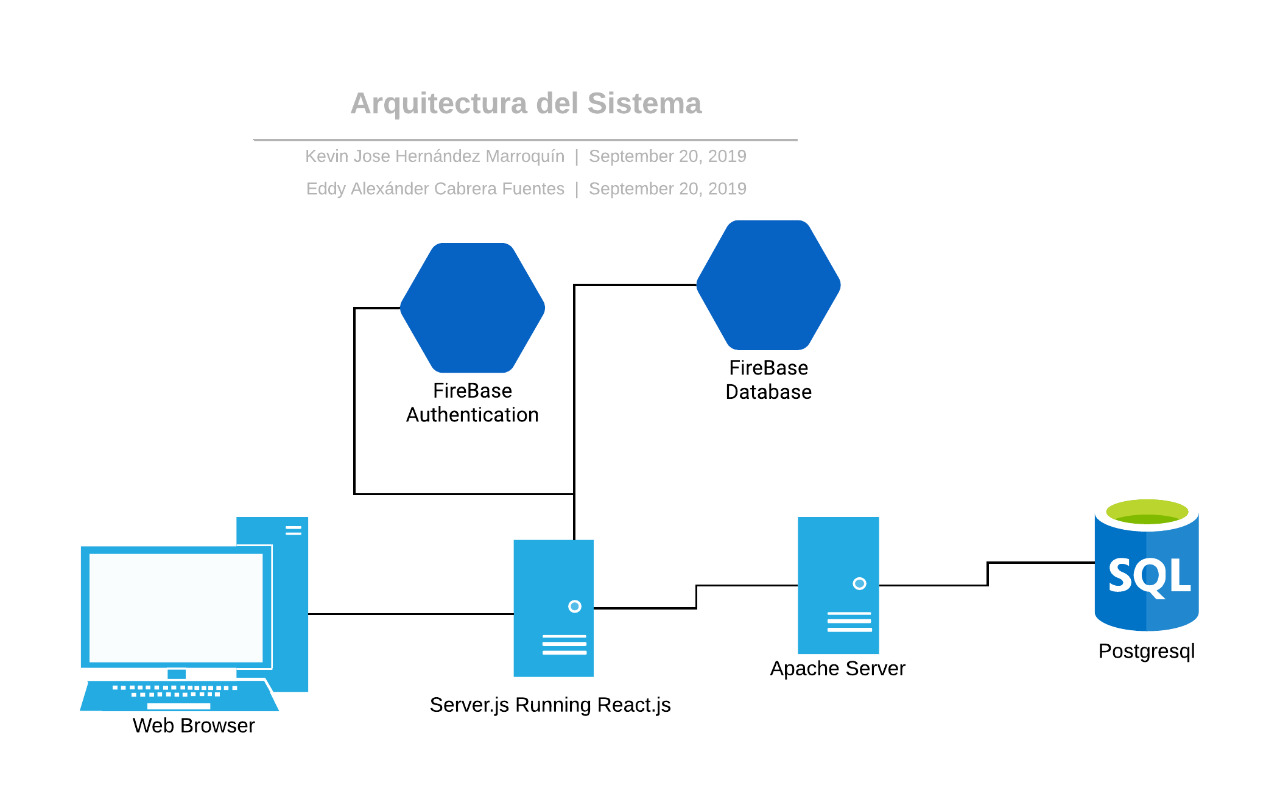
**Requerimientos de Usuario**

Servicios proveídos al usuario, requerimientos no funcionales, lenguaje natural, diagramas, definición de estándares de proceso o producto

* Solamente aquellas personas que se registren al sitio pueden acceder a todos los servicios que proporciona.
* El usuario debe escoger qué servicios desea.
* El usuario puede publicar ideas y dichas ideas pueden ser vistas por financistas.
* Al listar las ideas, se deben listar como 10 o 15 ideas.
* Al listar las ideas, el sistema no debe tardarse más de 10 segundos.
* Los financistas pueden guardar o marcar ideas en un libro para luego buscar de manera ágil la idea que desea.
* Los emprendedores pueden observar qué financistas les interesa alguna determinada idea.
* Se puede realizar videochat entre un financista y un emprendedor. El videochat no se guarda y debe ser estable.
* Eliminar ideas que no tengan sentido o evitar que se publiquen.
* Los usuarios pueden publicar recursos que no necesitan (pueden subir fotos y videos para aumentar la credibilidad).
* Los usuarios pueden buscar recursos que necesitan y realizar negocios.
* También se puede proveer el servicio de videochat entre los usuarios que publican recursos y aquellos que buscan recursos cuando quieran hacer negocios.
* Los usuarios pueden modificar su cuenta y habilitar o deshabilitar servicios.
* Los usuarios cuando ya no deseen utilizar la aplicación web pueden eliminar su cuenta.
* Se necesita un administrador que pueda añadir o eliminar las categorías de recursos e ideas.
* El administrador también puede administrar usuarios.
* Pueden implementar el proyecto usando Firebase y Amazon Web Services.
* Se necesita que la base de datos sea PostgreSQL.
* Pueden utilizar hasta 4 servidores.
* Se necesita que implementen Bootstrap para que se vea más bonito y profesional el sitio.
* Se necesita que exista un sistema de log in y sign up.
* Limitar el acceso de páginas de acuerdo al rol de usuario que la persona escogió.

**Arquitectura del Sistema**

La arquitectura utilizada para la implementación es una arquitectura web. El diagrama es el siguiente:



***Figura 1: Arquitectura del Sistema***

**Requerimientos del Sistema**

* **Requerimientos Funcionales:**

|  |  |
| --- | --- |
| **Función** | Registrar Usuario, Parte 1 |
|
| **Descripción** | Registrar el usuario al sistema asignándole el o los roles que dicho usuario escogió y determinar qué tipo de usuario le corresponde. |
|
|
|
| **Inputs** | Email, contraseña, rol y tipo de usuario. |
|
|
| **Outputs** | Mensaje de registro exitoso o registro rechazado. |
|
|
| **Acción** | Por medio de un formulario, el usuario ingresará todos los parámetros descritos en la sección de ***Inputs*** que luego será enviado a un servidor para extraer dichos parámetros y así construir la consulta SQL de tipo INSERT para ingresar el usuario a la tabla ***users***. |
|
|
|
| **Condiciones** | Es necesario que el usuario escoja obligatoriamente un rol. |
|
|
| **Efectos**  **Secundarios** | Dado que el usuario ha sido registrado, se le crea una carpeta asociada a su cuenta en donde se guardará las fotos o videos que suba. |
|
|

***Tabla 1: Registrar Usuario Parte 1***

|  |  |
| --- | --- |
| **Función** | Registrar Usuario, Parte 1.1 |
|
| **Descripción** | Registrar el usuario al tipo de usuario ‘Individuo’ si el usuario escogió que no representa una empresa. |
|
|
|
| **Inputs** | Nombres, apellidos, nacionalidad, biografía, organización afiliada, fecha de nacimiento y foto (imagen en formato png, jpg o jpge). |
|
|
| **Outputs** | Mensaje de registro exitoso o registro rechazado. |
|
|
| **Acción** | Por medio de un formulario, el usuario ingresará todos los parámetros descritos en la sección de ***Inputs*** que luego será enviado a un servidor para extraer dichos parámetros y así construir la consulta SQL de tipo INSERT para ingresar el usuario a la tabla ***Individual***. La foto ingresada por el usuario se tomará como la foto de perfil que será renombrada como ***profile***. |
|
|
|
| **Condiciones** | Es necesario que el usuario haya escogido que no representa a una organización en el formulario. |
|
|
| **Efectos**  **Secundarios** | El usuario será capaz de modificar luego los datos asociados a su cuenta de tipo ***Individual***. |
|
|

***Tabla 2: Registrar Usuario Parte 1.1***

|  |  |
| --- | --- |
| **Función** | Registrar Usuario, Parte 1.1 |
|
| **Descripción** | Registrar el usuario al tipo de usuario ‘Organización’ si el usuario escogió que no representa una empresa. |
|
|
|
| **Inputs** | Nombre, descripción, país, lugar de ubicación y logo. |
|
|
| **Outputs** | Mensaje de registro exitoso o registro rechazado. |
|
|
| **Acción** | Por medio de un formulario, el usuario ingresará todos los parámetros descritos en la sección de ***Inputs*** que luego será enviado a un servidor para extraer dichos parámetros y así construir la consulta SQL de tipo INSERT para ingresar el usuario a la tabla ***Organization***. La foto ingresada por el usuario se tomará como la foto de perfil que será renombrada como ***profile***. |
|
|
|
| **Condiciones** | Es necesario que el usuario haya escogido que su cuenta representa a una organización en el formulario. |
|
|
| **Efectos**  **Secundarios** | El usuario será capaz de modificar luego los datos asociados a su cuenta de tipo ***Individual***. |
|
|

***Tabla 3: Registro de Usuario Parte 1.1***

|  |  |
| --- | --- |
| **Función** | Emprendedor publica idea. |
|
| **Descripción** | Un usuario de rol de tipo emprendedor publica una idea. |
|
|
|
| **Inputs** | Título, descripción, categoría, estado de la idea y el ID del emprendedor que publica. |
|
|
| **Outputs** | Mensaje de ingreso exitoso o registro rechazado. |
|
|
| **Acción** | Por medio de un formulario, el usuario ingresará todos los parámetros descritos en la sección de ***Inputs*** que luego será enviado a un servidor para extraer dichos parámetros y así construir la consulta SQL de tipo INSERT para ingresar el usuario a la tabla ***Idea***. |
|
|
|
| **Condiciones** | Es necesario que el usuario posea el rol de emprendedor. |
|
|
| **Efectos**  **Secundarios** | La cantidad de usuarios interesados se inicializa con 0 por default y el estado de la idea siempre es ‘activo’ por default. |
|
|

***Tabla 4: Emprendedor publica una idea***

|  |  |
| --- | --- |
| **Función** | Enlistar ideas de un emprendedor. |
|
| **Descripción** | Enlistar ideas que un emprendedor haya publicado y que muestre la cantidad de personas interesadas en las respectivas ideas. |
|
|
|
| **Inputs** | ID del emprendedor. |
|
|
| **Outputs** | Todas las ideas publicadas relacionadas con el ID del emprendedor. |
|
|
| **Acción** | En la página principal para emprendedores, el usuario puede hacer click en un botón que le permita ver todas las ideas publicadas. Dicho botón envía el ID del emprendedor y luego es enviado al servidor para construir la consulta SQL de tipo ***SELECT*** y así extraer todas las ideas asociadas al ID del emprendedor que se encuentran en la tabla ***Idea.*** |
|
|
|
| **Condiciones** | Las ideas deben ser mostradas en una tabla, cuyo número de filas debe ser un número que es mandado como parámetro al servidor. |
|
|
| **Efectos**  **Secundarios** | El usuario será capaz de ver las ideas en grupos pequeños para facilitar la visualización. |
|
|

***Tabla 5: Emprendedor enlista ideas publicadas***

|  |  |
| --- | --- |
| **Función** | Borrar una idea de un emprendedor |
|
| **Descripción** | El usuario de tipo emprendedor puede borrar ideas ya publicadas. |
|
|
|
| **Inputs** | ID de la idea. |
|
|
| **Outputs** | Mensaje de eliminación exitosa o eliminación rechazada. |
|
|
| **Acción** | Por medio de un formulario, las ideas son recopiladas (que están asociadas al usuario) y el usuario puede escoger la que quiere eliminar. Dicha idea tiene un identificador que es enviado al servidor para que construya una consulta SQL de tipo ***DELETE*** para eliminar la idea seleccionada. |
|
|
|
| **Condiciones** | Es necesario que la idea exista en la base de datos. |
|
|
| **Efectos**  **Secundarios** | La idea será borrada en todos los bookmark de financistas que tienen dicha idea registrada. |
|
|

***Tabla 6: Emprendedor borra una idea ya publicada***

|  |  |
| --- | --- |
| **Función** | Editar idea de un emprendedor. |
|
| **Descripción** | El usuario de tipo emprendedor puede modificar una idea y puede cambiar su información. |
|
|
|
| **Inputs** | ID de la idea y campo seleccionado por el usuario (título, descripción, categoría o estado). |
|
|
| **Outputs** | Idea modificada. |
|
|
| **Acción** | Por medio de un formulario, el usuario puede seleccionar alguna idea de todas de las que ha publicado y puede modificar un campo que desea. |
|
|
|
| **Condiciones** | Es necesario que la idea ya exista en la base de datos. |
|
|
| **Efectos**  **Secundarios** | El sistema tendrá la idea actualizada. |
|
|

***Tabla 7: Emprendedor edita una idea ya publicada***

|  |  |
| --- | --- |
| **Función** | Enlistar ideas a un financista por categoría. |
|
| **Descripción** | Mostrar al usuario financista todas las ideas relacionadas a una categoría. |
|
|
|
| **Inputs** | Categoría. |
|
|
| **Outputs** | Lista de todas las ideas relacionadas a la categoría. |
|
|
| **Acción** | Por medio de un formulario, el usuario ingresará la categoría de ideas que desea observar. El formulario será enviado a un servidor y dicho servidor extraerá el valor de la categoría y construirá la consulta SQL de tipo ***SELECT*** que extraerá todas las ideas de la tabla ***Idea*** que están relacionadas con la categoría. |
|
|
|
| **Condiciones** | Es necesario que la categoría exista. |
|
|
| **Efectos**  **Secundarios** | Ninguno. |
|
|

***Tabla 8: Financista enlista ideas por categoría***

|  |  |
| --- | --- |
| **Función** | Bookmark de una idea |
|
| **Descripción** | El financista puede marcar una idea y registrarla en su bookmark para luego buscarla de manera ágil. |
|
|
|
| **Inputs** | ID de la idea y el ID del usuario. |
|
|
| **Outputs** | Mensaje de registro exitoso o registro rechazado. |
|
|
| **Acción** | Tras haber enlistado todas las ideas de una categoría específica al financista, el usuario puede seleccionar una idea que le gustó y hacer click a un botón de “add to bookmark” que le permita registrar la idea en el bookmark asociado a su cuenta. El servidor obtiene los ID de la idea y del usuario y luego construye la consulta SQL de tipo INSERT para insertar la idea en la tabla ***FinBook*** |
|
|
|
| **Condiciones** | Es necesario que la idea exista en la categoría en la que buscó el usuario. |
|
|
| **Efectos**  **Secundarios** | El usuario de tipo financista puede buscar de manera más sencilla a la idea haciendo click en el botón “bookmark” que se encuentra en su página principal de financista. |
|
|

***Tabla 9: Financista agregar una idea a su bookmark***

|  |  |
| --- | --- |
| **Función** | Borrar una idea de un bookmark de un financista. |
|
| **Descripción** | El financista borra una idea que marcó en su bookmark. |
|
|
|
| **Inputs** | ID de la idea y el ID del usuario. |
|
|
| **Outputs** | Mensaje de eliminación exitosa o eliminación rechazada. |
|
|
| **Acción** | Tras enlistar las ideas del bookmark asociado al financista, el usuario puede escoger la idea que desea eliminar de su bookmark. Tras hacer click en “eliminar”, el servidor recibe el ID de la idea y el ID del usuario que con ellos genera una consulta SQL de tipo ***DELETE*** que borra la idea de la tabla ***FinBook*** que tiene asociada ambos IDs. |
|
|
|
| **Condiciones** | Es necesario que el usuario haya escogido una idea de su bookmark. |
|
|
| **Efectos**  **Secundarios** | El financista no volverá a ver la idea en su bookmark y tendrá que buscarla otra vez. |
|
|

***Tabla 10: Financista borra ideas de su bookmark***

|  |  |
| --- | --- |
| **Función** | Publicación de un bien de un negociante. |
|
| **Descripción** | El negociante publica un bien que está dispuesto a realizar negocios con él. |
|
|
|
| **Inputs** | ID del usuario, descripción, categoría, fotos y videos. |
|
|
| **Outputs** | Mensaje de ingreso exitoso o ingreso rechazado. |
|
|
| **Acción** | El usuario debe llenar un formulario llenando los parámetros descritos en la sección de ***Inputs***. Dicho formulario se envía al servidor para que obtenga los parámetros y construya la consulta SQL de tipo ***INSERT*** que pueda insertar el bien en la tabla ResGive, asociándolo con el ID del usuario. |
|
|
|
| **Condiciones** | Es necesario que el usuario sea de rol de negociante. |
|
|
| **Efectos**  **Secundarios** | El bien estará disponible públicamente para todos aquellos usuarios con rol de negociante. |
|
|

***Tabla 11: Negociante publica un bien***

|  |  |
| --- | --- |
| **Función** | Enlistar bienes de acuerdo con una categoría. |
|
| **Descripción** | El negociante puede ver todos los bienes asociados a una categoría que él escoja. |
|
|
|
| **Inputs** | Categoría. |
|
|
| **Outputs** | Lista de todos los bienes asociados a la categoría. |
|
|
| **Acción** | El usuario debe seleccionar la categoría con la que quiere filtrar los bienes. Este parámetro es enviado al servidor que construye la consulta SQL de tipo ***SELECT*** para obtener todos los bienes. |
|
|
|
| **Condiciones** | Es necesario que la categoría del bien exista. |
|
|
| **Efectos**  **Secundarios** | Ninguno. |
|
|

***Tabla 12: Negociante enlista bienes por categoría***

|  |  |
| --- | --- |
| **Función** | Administrador cree una nueva categoría de ideas. |
|
| **Descripción** | El administrador de la aplicación web puede crear nuevas categorías de ideas. |
|
|
|
| **Inputs** | Nombre |
|
|
| **Outputs** | Mensaje de ingreso exitoso o ingreso rechazado. |
|
|
| **Acción** | El administrador debe seleccionar “añadir categoría” o “crear nueva categoría” y luego se le presenta un formulario que le permite ingresar el nuevo nombre de la categoría. Dicho formulario se envía al servidor y este extrae los parámetros para construir una consulta SQL de tipo ***INSERT*** que añade la nueva categoría a la tabla ***CategoryIdea.*** |
|
|
|
| **Condiciones** | Es necesario que el usuario sea un administrador. |
|
|
| **Efectos**  **Secundarios** | Los emprendedores pueden publicar ideas en esa categoría nueva. |
|
|

***Tabla 13: Administrador crea una nueva categoría de ideas***

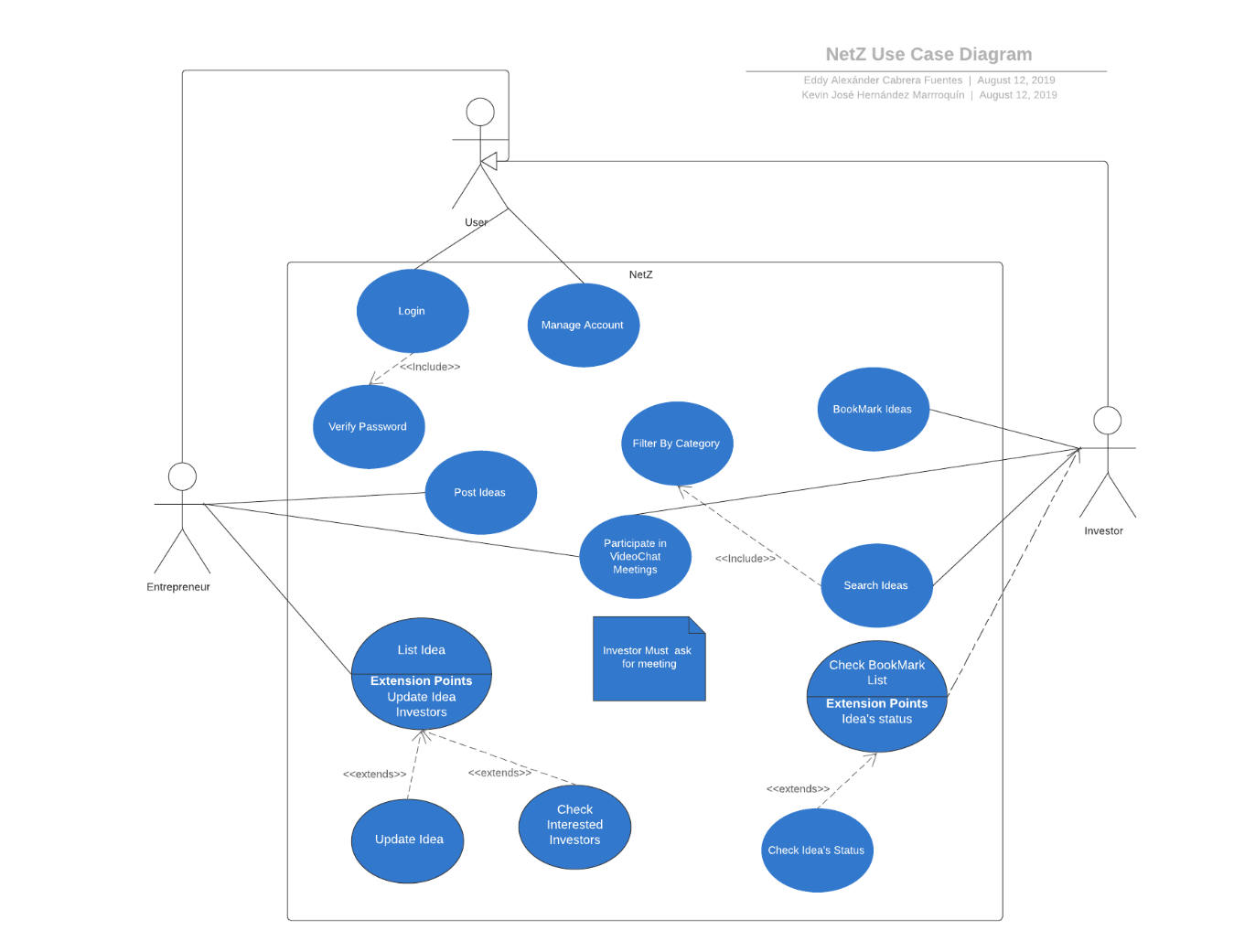
|  |  |
| --- | --- |
| **Función** | Administrador cree una nueva categoría de ideas. |
|
| **Descripción** | El administrador de la aplicación web puede crear nuevas categorías de recursos. |
|
|
|
| **Inputs** | Nombre. |
|
|
| **Outputs** | Mensaje de ingreso exitoso o ingreso rechazado. |
|
|
| **Acción** | El administrador debe seleccionar “añadir categoría” o “crear nueva categoría” y luego se le presenta un formulario que le permite ingresar el nuevo nombre de la categoría. Dicho formulario se envía al servidor y este extrae los parámetros para construir una consulta SQL de tipo ***INSERT*** que añade la nueva categoría a la tabla ***CategoryRes.*** |
|
|
|
| **Condiciones** | Es necesario que el usuario sea un administrador. |
|
|
| **Efectos**  **Secundarios** | Los negociantes pueden publicar ideas en esa categoría nueva. |
|
|

***Tabla 14: Administrador crea una nueva categoría de ideas***

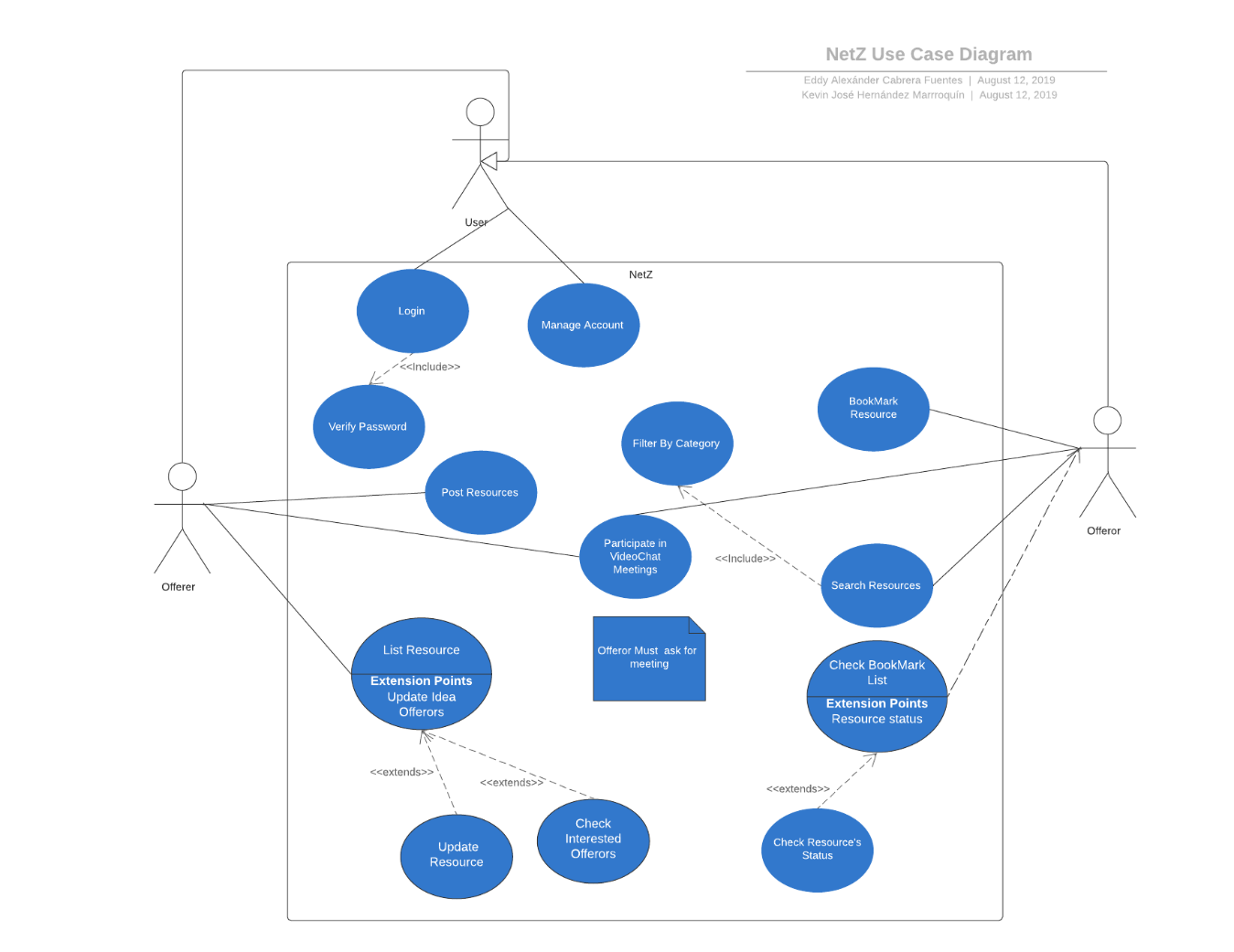
**Modelos del Sistema**

**Diagramas UML:**

**Casos de Uso:**

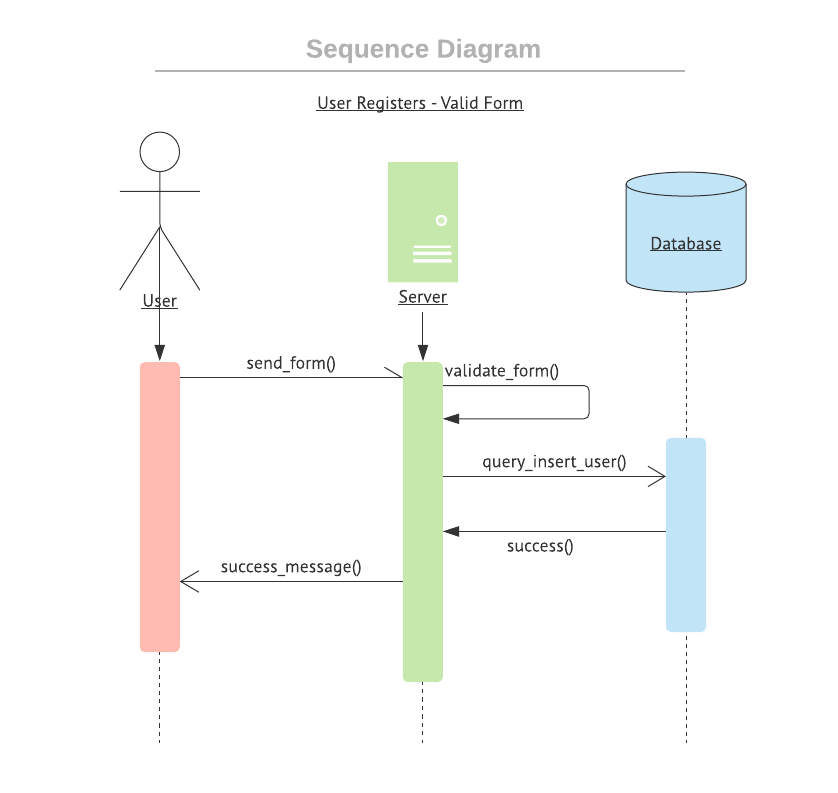


***Figura 2: Diagrama Caso de Uso de Emprendedor-Financista***

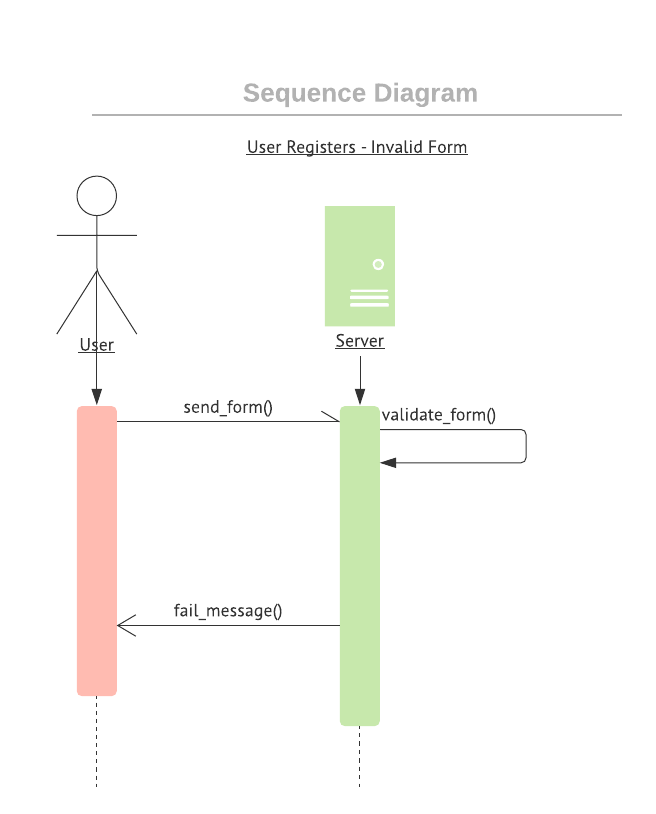


***Figura 3: Diagrama Caso de Uso de Recursos***

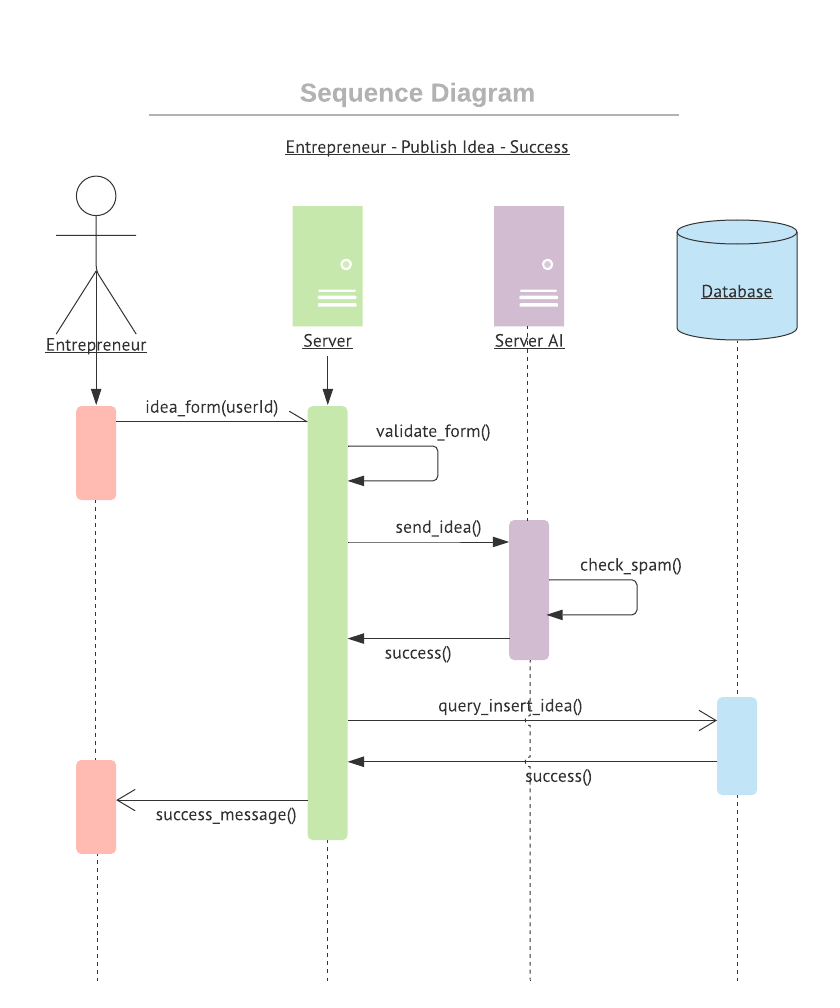
**Diagramas de Secuencia:**



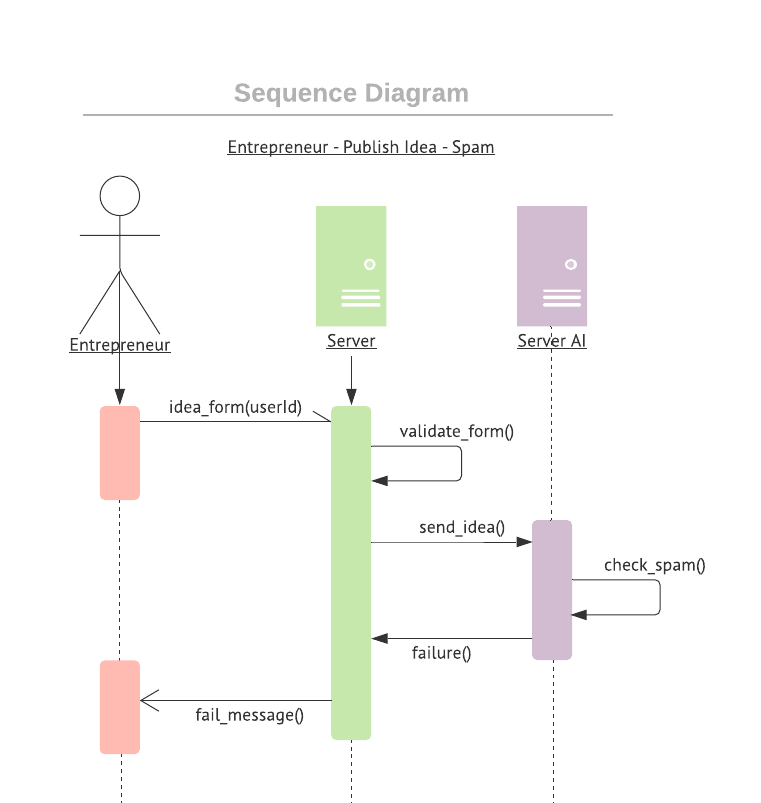
***Figura 4: Diagrama de Secuencia de Usuario – Registro Exitoso***



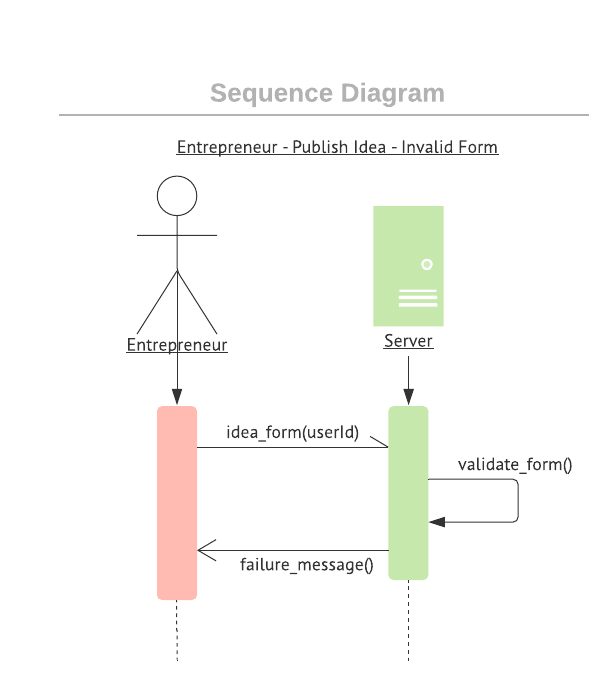
***Figura 5: Diagrama de Secuencia de Usuario – Registro Rechazado***



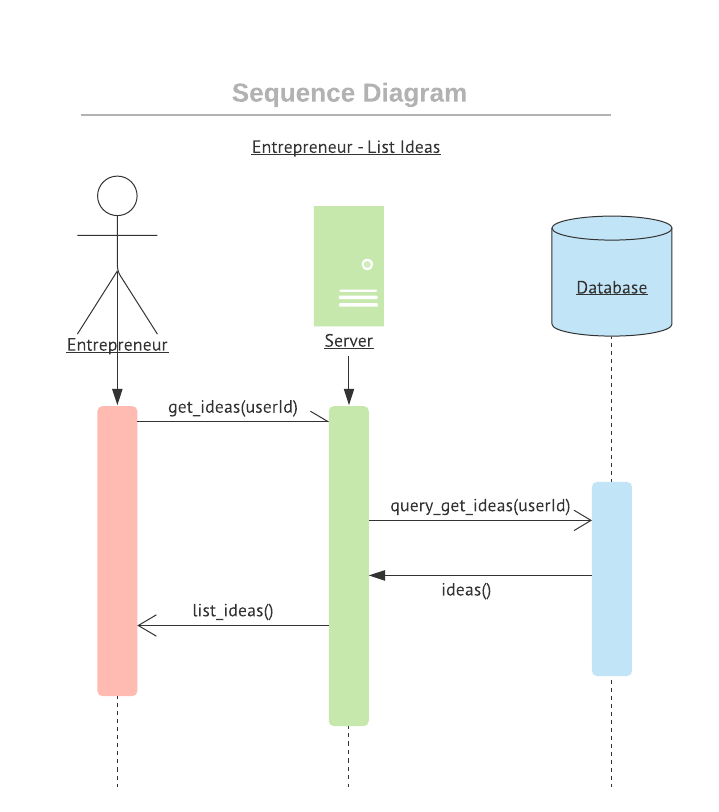
***Figura 6: Diagrama de Secuencia de Emprendedor - Publicando idea y es exitoso***



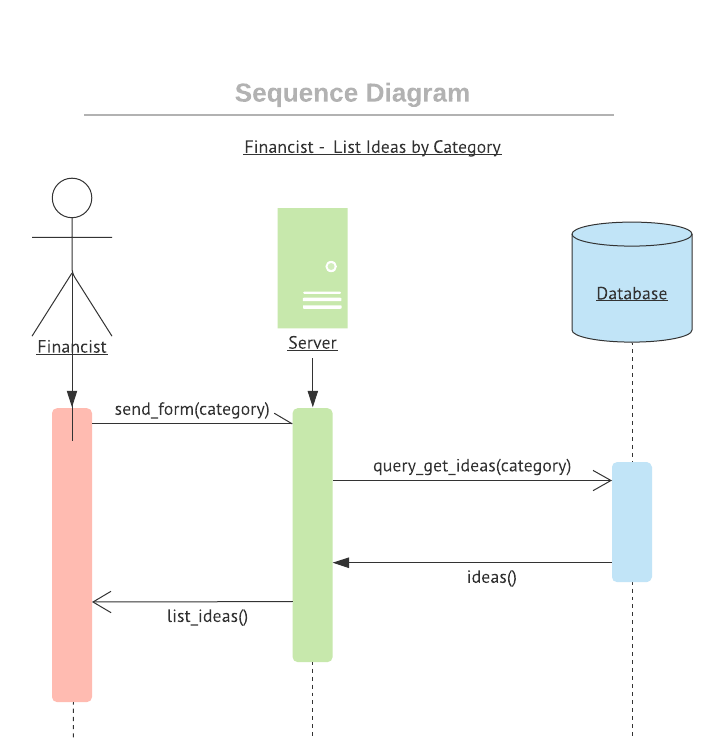
***Figura 7: Diagrama de Secuencia de Emprendedor – Idea es filtrada como Spam***



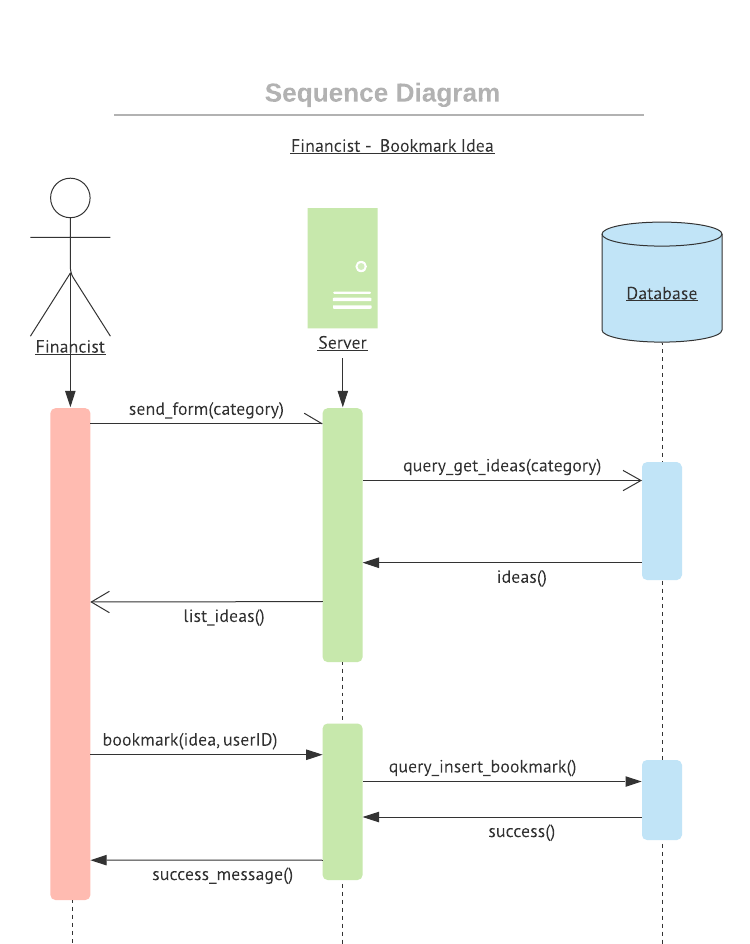
***Figura 8: Diagrama de Secuencia de Emprendedor – Formulario Inválido***



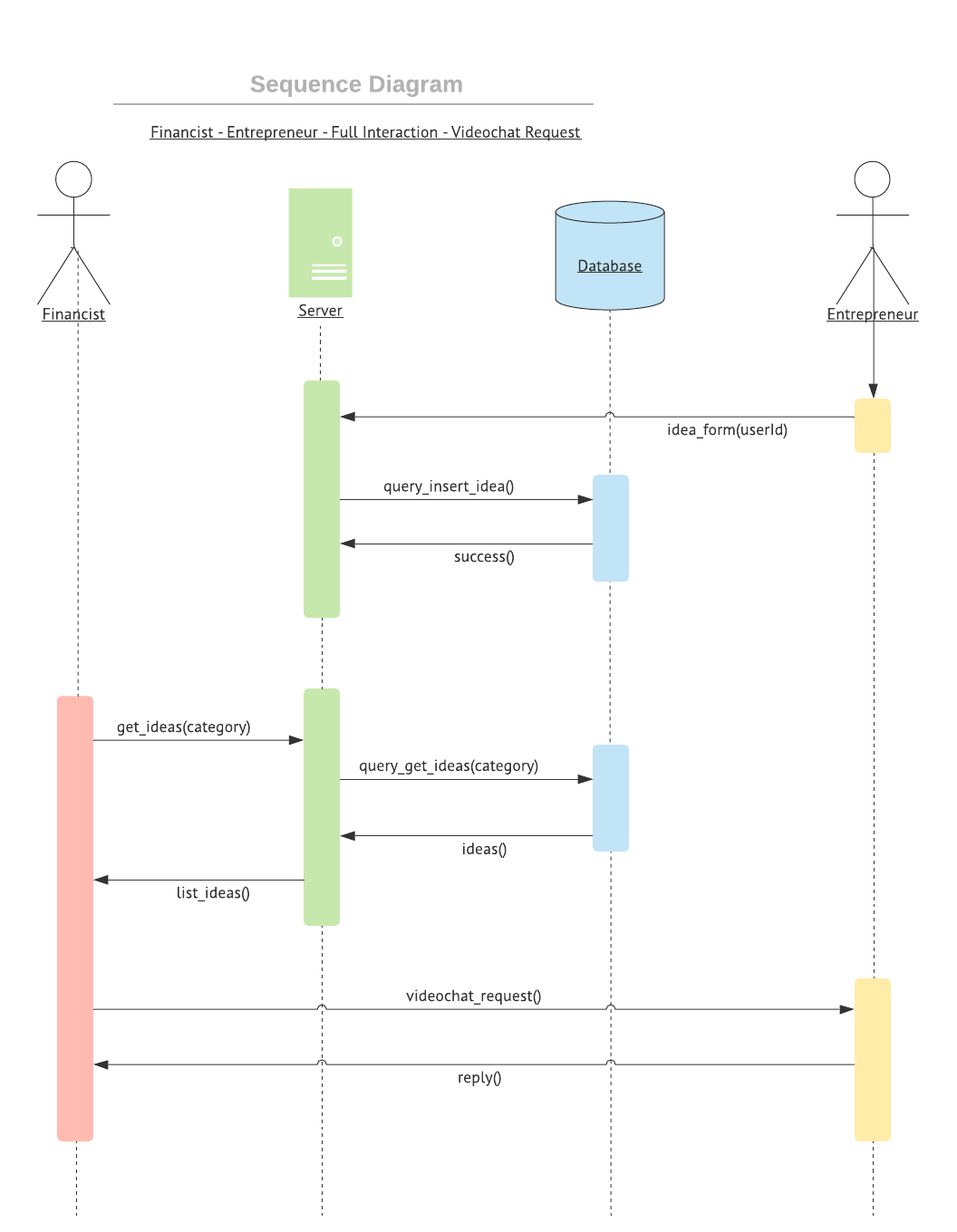
***Figura 9: Diagrama de Secuencia de Emprendedor – Enlistar ideas Publicadas***



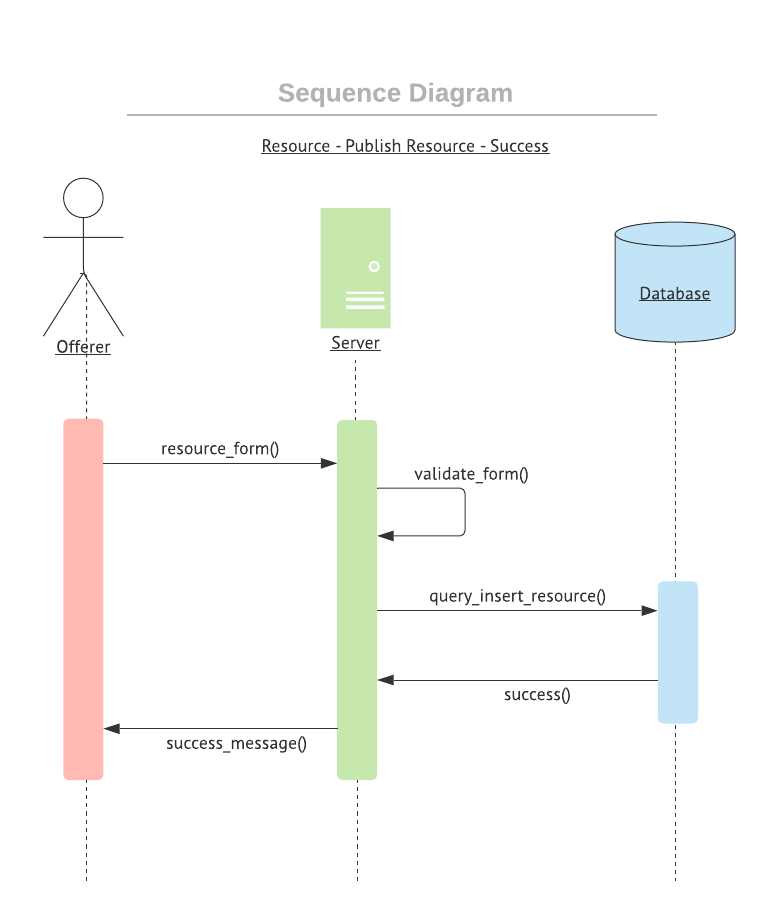
***Figura 10: Diagrama de Secuencia de Financista – Enlistar Ideas por Categoría***



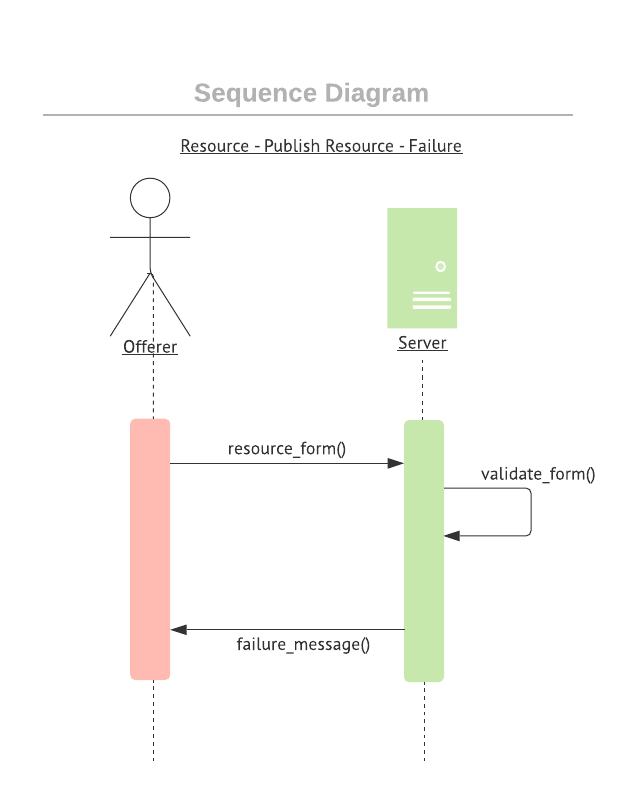
***Figura 11: Diagrama de Secuencia de Financista – Marcar idea en Bookmark***



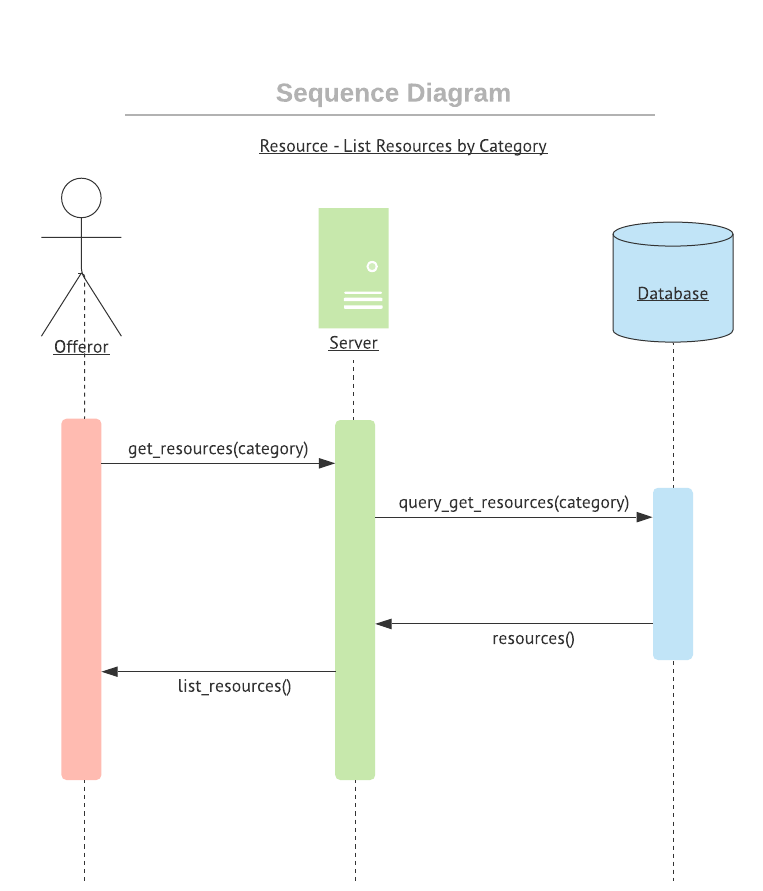
***Figura 12: Diagrama de Secuencia de Financista y Emprendedor – Interacción completa del Sistema***



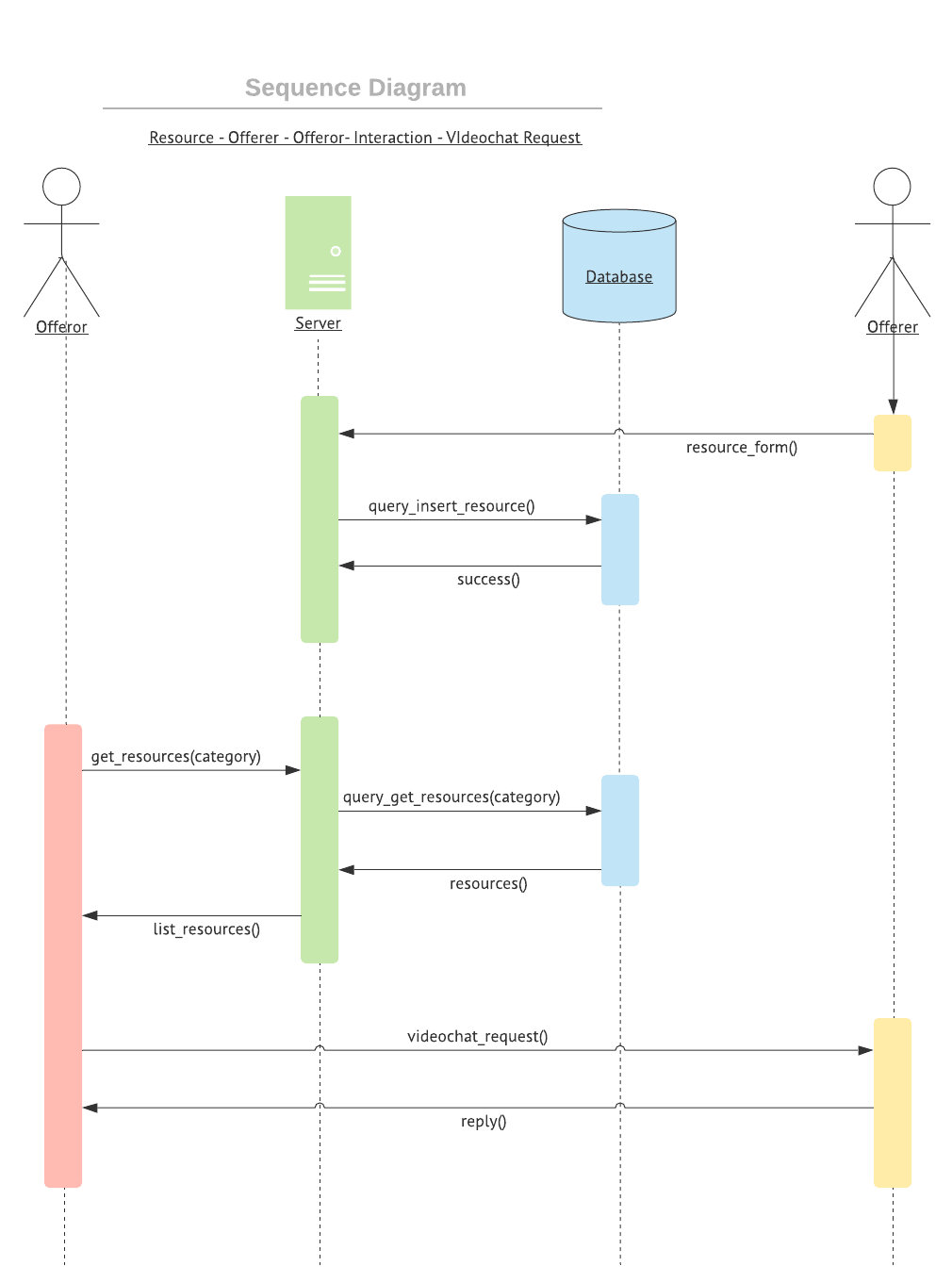
***Figura 13: Diagrama de Secuencia de Negociante – Publicar recurso exitosamente***



***Figura 14: Diagrama de Secuencia de Negociante – Publicar Recurso es Rechazado***

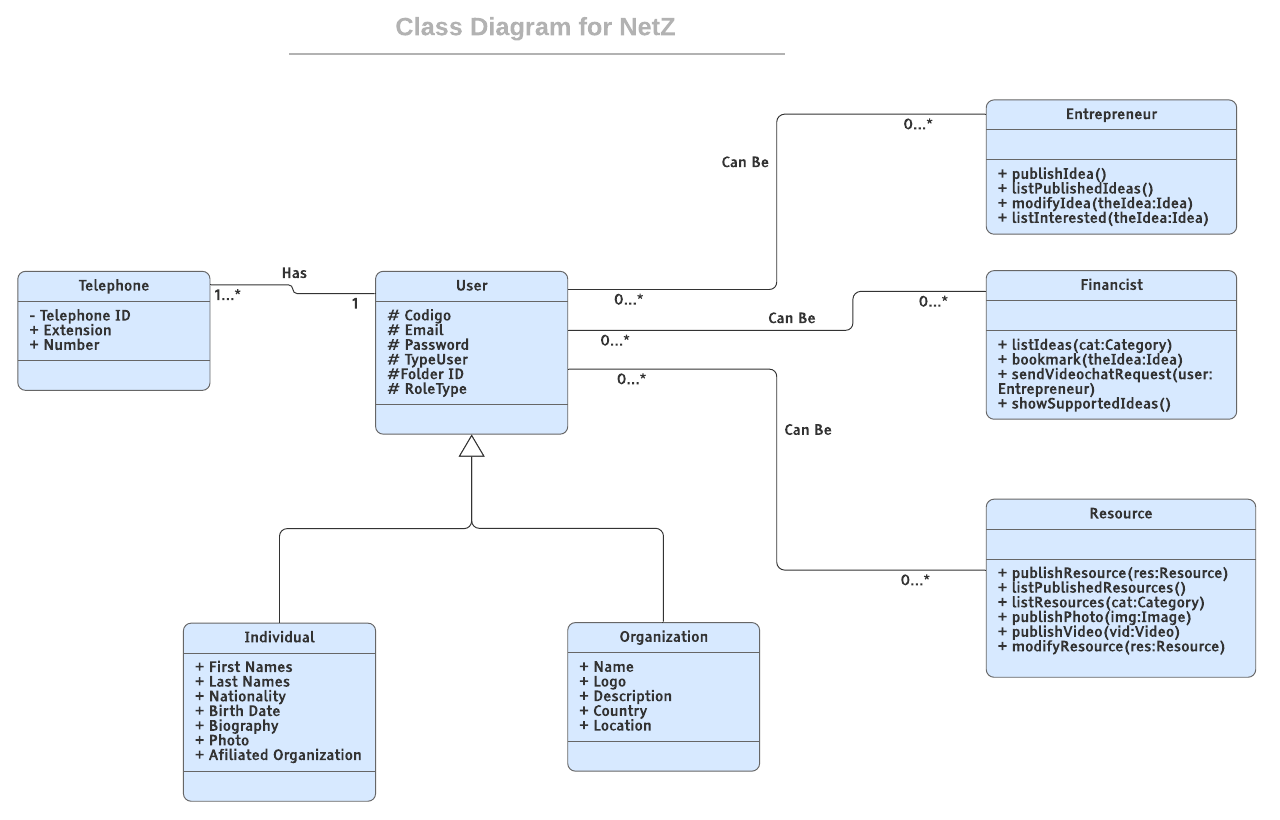


***Figura 15: Diagrama de Secuencia de Negociante – Enlistar Recursos por Categoría***



***Figura 16: Diagrama de Secuencia de Negociantes – Interacción Completa del Sistema***

**Diagramas de Clase:**



***Figura 17: Diagrama de Clases del Sistema***

**Evolución del Sistema**

* **Cambios muy probables:**
  + Interfaz gráfica:
    - Páginas de usuario:
      * Página para cambiar contraseña y correo electrónico.
      * Página para ver detalles de la cuenta.
      * Página para configurar el rol del usuario.
    - Páginas de Emprendedor:
      * Página para enlistar ideas ya publicadas y los interesados en cada respectiva idea.
      * Página para publicar ideas de acuerdo con una categoría.
    - Páginas de Financista:
      * Página para enlistar ideas de acuerdo con una categoría.
      * Página para enlistar ideas del bookmark.
    - Páginas de Negociante:
      * Página para enlistar bienes de acuerdo con una categoría.
      * Página para enlistar bienes ya vendidos o intercambiados.
  + Esquemas:
    - Tabla de Usuario:
      * Agregar más campos.
    - Tabla de Idea:
      * Agregar más campos.
    - Tabla de Emprendedor:
      * Agregar más campos.
    - Tabla de Financista:
      * Agregar más campos.
    - Tabla de Negociante:
      * Agregar más campos.
  + Hardware:
    - Cambiar el servidor por uno más rápido y efectivo.
    - Discos duros con mucha más capacidad.
    - Agregar más servidores para manejar concurrencia.

**Apéndice**

* **Información adicional de la aplicación:**
  + Pueden existir varios administradores que pueden gestionar la aplicación web.
  + Existe una inteligencia artificial que analiza las ideas publicadas por los emprendedores y los filtra ya sea por spam o no.
  + El videochat se puede realizar únicamente cuando el financista y el emprendedor interesado están de acuerdo.
  + Existe un chat debajo del videochat que puede ser usado para tomar notas o apuntar cosas importantes.
  + El videochat no se guardará en ninguna parte.
  + El chat no se guardará en ninguna parte.
* **Requerimientos de Hardware Mínimos:**
  + Discos duros con capacidad de hasta 5TB.
  + Procesadores Intel Core i7 octava generación.
  + Memorias RAM de 16 GB.
  + Tarjetas gráficas NVIDIA GTX Geforce de 4GB.
  + 3 Servidores.
* **Descripción de Hardware:**
  + Servidores para montar los servicios web y la base de datos.
* **Descripción de la Base de Datos:**
  + Motor: PostreSQL
  + Versión: 11.2
  + 64 bits