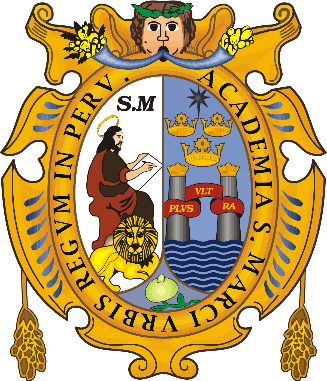
Año del Bicentenario del Perú: 200 años de Independencia



**UNIVERSIDAD NACIONAL MAYOR DE**

**SAN MARCOS**

**(Universidad del Perú, DECANA DE AMÉRICA)**

**FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

***Escuela Profesional de Ingeniería de Software***

Imagen que contiene Logotipo

Descripción generada automáticamente

**Diseño de Arquitectura**

**Curso:**  Gestión de la Configuración del Software

**Profesor:** Espinoza Robles, Armando David

**Grupo:** 6

**Integrantes:**

* Hugo Quispe, Railly Angelo
* Gomez Huanca, Elian Dalmiro
* Lizana Yauri, Alexander Daniel
* Ramos Salinas, Maicol
* Rivas Rodriguez, Catherine Lesly
* Rojas Rojas, Mauro Alexander
* Tarmeño Noriega, Carlos Daniel
* Ytusaca Quispe, Saúl Fernando

Ciudad Universitaria, 5 de junio del 2021

1. **Vista de escenario**
2. **Descripción del Negocio**

Foody es una aplicación web que posee un sistema para registro de cuentas, que principalmente permite almacenar recetas mediante libro de recetas, consideradas listas de recetas, de manera sencilla y rápida mediante su interfaz interactiva, además de poder gestionar un horario para organizar y poder cocinar las comidas registradas en tu libro de recetas.

1. **Modelo de procesos de negocios relevantes para la arquitectura**

Procesos de negocio relevante para el sistema

Se identificó 1 proceso de negocio:

● PN1: Gestión de solicitud de cuentas premium

El proceso de negocio más relevante para el sistema es Gestión de solicitud de cuentas premium. Identificamos este proceso y lo hemos dividido en 3 subprocesos; los cuales vienen a ser: gestión de pedido de una cuenta, gestión de envío de solicitud de una cuenta con mayores privilegios y/o beneficios (premium) y gestión de verificación y entrega de cuenta premium . El primer subproceso es cuando el usuario crea una cuenta en Foody . El segundo subproceso es cuando el usuario envía una solicitud para acceder a más funcionalidades (premium). Y el tercer subproceso es la verificación de la solicitud y el otorgamiento de esta nueva condición a la cuenta.

1. **Modelo de dominio de la aplicación**
2. **CUS relevantes organizados en paquetes**

1. **Descripción de los CUS relevantes para la Arquitectura**

|  |  |  |
| --- | --- | --- |
| **HU-01** | Mostrar la página de inicio | * Maquetar cuerpo principal de la página * Crear barra de navegación para acceder al resto de páginas de uso público * Enrutar las páginas públicas |
| **HU-02** | Mostrar la página de servicios | * Maquetar cuerpo de la página de servicios |
| **HU-03** | Mostrar la página de precios | * Maquetar cuerpo de la página de precios |
| **HU-04** | Mostrar opciones de registro y login | * Crear botón “ir a registro” * Crear botón “ir a inicio de sesión” |
| **HU-05** | Registrar usuario con correo y contraseña | * Crear formulario con campos correo y contraseña * Conectarse a firebase-auth para enviar los datos y almacenarlo en la base de datos |
| **HU-06** | Mostrar feedback visual en caso de éxito o error | * Crear toast de error para los siguientes cases   + Usuario no registrado   + Usuario ya registrado   + Contraseña muy corta   + Campo(s) vacío(s) |
| **HU-07** | Limitar acceso a usuarios autentificados | * Crear función usando el enrutador para identificar la dirección actual y redireccionar al usuario a la aplicación en caso esté autenticado |
| **HU-08** | Permitir navegar a página de inicio de sesión | * Crear diccionario de direcciones * Crear enlace dinámico para navegar al inicio de sesión |
| **HU-09** | Iniciar sesión con usuario con correo y contraseña | * Crear formulario con campos correo y contraseña * Conectarse a firebase-auth para enviar los datos y compararlos con los valores correctos. |
| **HU-10** | Permitir el inicio de sesión con Google | * Crear botón con la normativa de Google * Conectarse a firebase-auth para acceder al proveedor de Google. |
| **HU-11** | Permitir el inicio de sesión con Facebook | * Crear botón con la normativa de Google * Conectarse a firebase-auth para acceder al proveedor de Facebook. |
| **HU-12** | Mostrar feedback visual en caso de éxito o error | * Crear toast de error para los siguientes cases   + Contraseña incorrecta   + Usuario no registrado   + Muchos intentos fallidos   + Campo(s) vacío(s) |

* **Descripción de Caso de Uso**

|  |  |
| --- | --- |
| **Código: HU-01** | **Nombre:**Mostrar la página de inicio |
| **Descripción:**  **COMO:** Administrador  **QUIERO:** poder mostrar la página de inicio  **PARA:** ofrecer una vista de lo que nuestro producto ofrece | |
| **Criterios de Aceptación :**   * **C01:** Como usuario quiero ver una página de inicio, describiendo la función de la aplicación * **C02:** Como administrador quiero ver una página de inicio ordenada y amigable con el usuario y fácil de saber dónde están sus componentes | |

|  |  |
| --- | --- |
| **Código: HU-04** | **Nombre:** Mostrar opciones de registro y login. |
| **Descripción:**  **COMO:** Administrador  **QUIERO:** poder ver las opciones de registro y login en mi producto y saltar su respectiva interfaz  **PARA:** ingresar o registrarme | |
| **Criterios de Aceptación :**   * **C01:** Poder ver el botón de login de manera sencilla * **C02:** Poder ver el botón de registrarse de manera sencilla | |

|  |  |
| --- | --- |
| **Código: HU-05** | **Nombre:** Registrar usuario con correo y contraseña |
| **Descripción:**  **COMO:** Administrador  **QUIERO:** poder agregar a un nuevo usuario a través de la opción de registro  **PARA:** Crear una nueva sesión de usuario | |
| **Criterios de Aceptación :**   * **C01:** Ver si el correo ya está registrado o no * **C02:** Ofrecer la opción de registrarse mediante gmail. * **C03:** Verificar si el correo tiene su terminación correcta | |

|  |  |
| --- | --- |
| **Código: HU-06** | **Nombre:** Mostrar feedback visual en caso de éxito o error. |
| **Descripción:**  **COMO:**Usuario  **QUIERO:** poder ver si mi registro se ha realizado con éxito, o si fallo en algo saber en donde  **PARA:** asegurar al usuario que su cuenta ha sido creada, o que algún dato no ha sido validado correctamente | |
| **Criterios de Aceptación :**   * **C01:** Si ha fallado en algún campo , aparecerá una ventana con mensaje de error * **C02:** Si no hubo errores le aparecerá un mensaje de éxito en la creación de cuenta | |

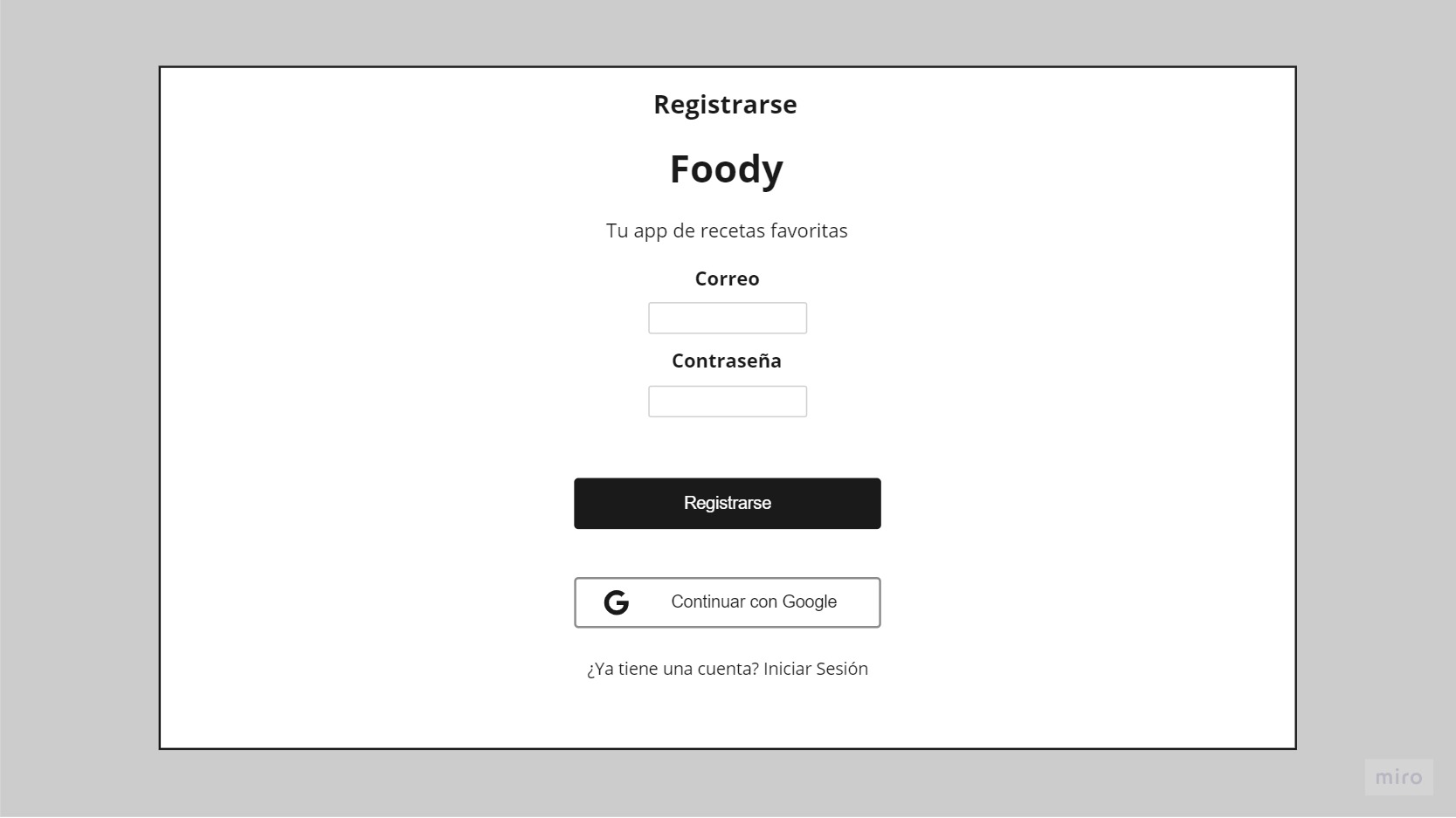
|  |  |
| --- | --- |
| **Código: HU-09** | **Nombre:** Iniciar sesión de usuario con correo y contraseña |
| **Descripción:**  **COMO:** Usuario  **QUIERO:** poder validar la existencia del correo  **PARA:** El ingreso al sistema | |
| **Criterios de Aceptación :**   * **C01:**Poder poner mi usuario o correo en la interfaz de login y que estén validados. * **C02:**Poder poner mi contraseña y que no sea visible, y esté validada. | |

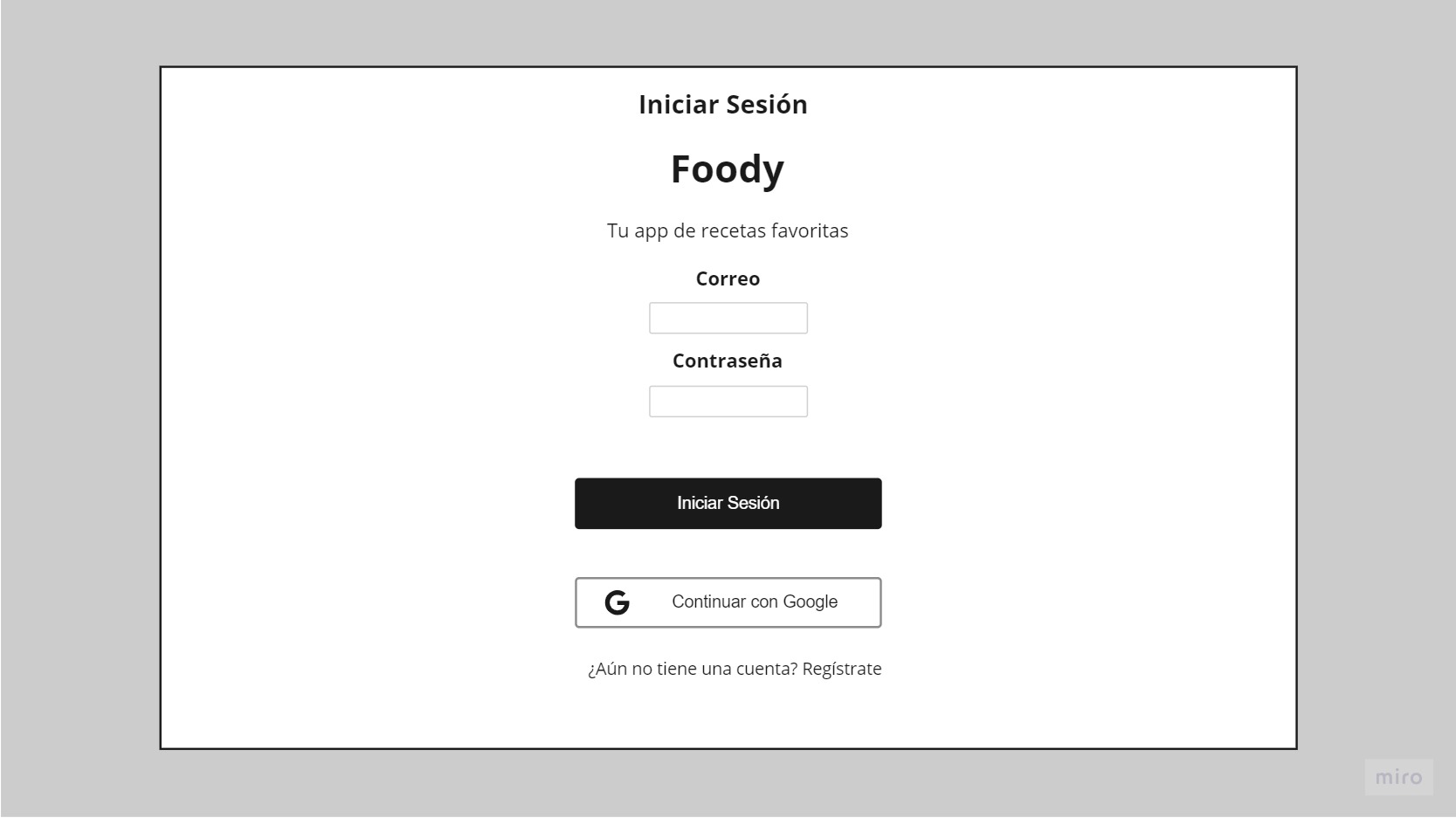
|  |  |
| --- | --- |
| **Código: HU-10** | **Nombre:** Permitir el inicio de sesión con Google. |
| **Descripción:**  **COMO:** Usuario  **QUIERO:** poder ingresar al sistema usando mi cuenta de google  **PARA:** facilitar mi ingreso al sistema. | |
| **Criterios de Aceptación :**   * **C01:** El usuario debe poder acceder al sistema vinculando su cuenta de google con nuestra plataforma. | |

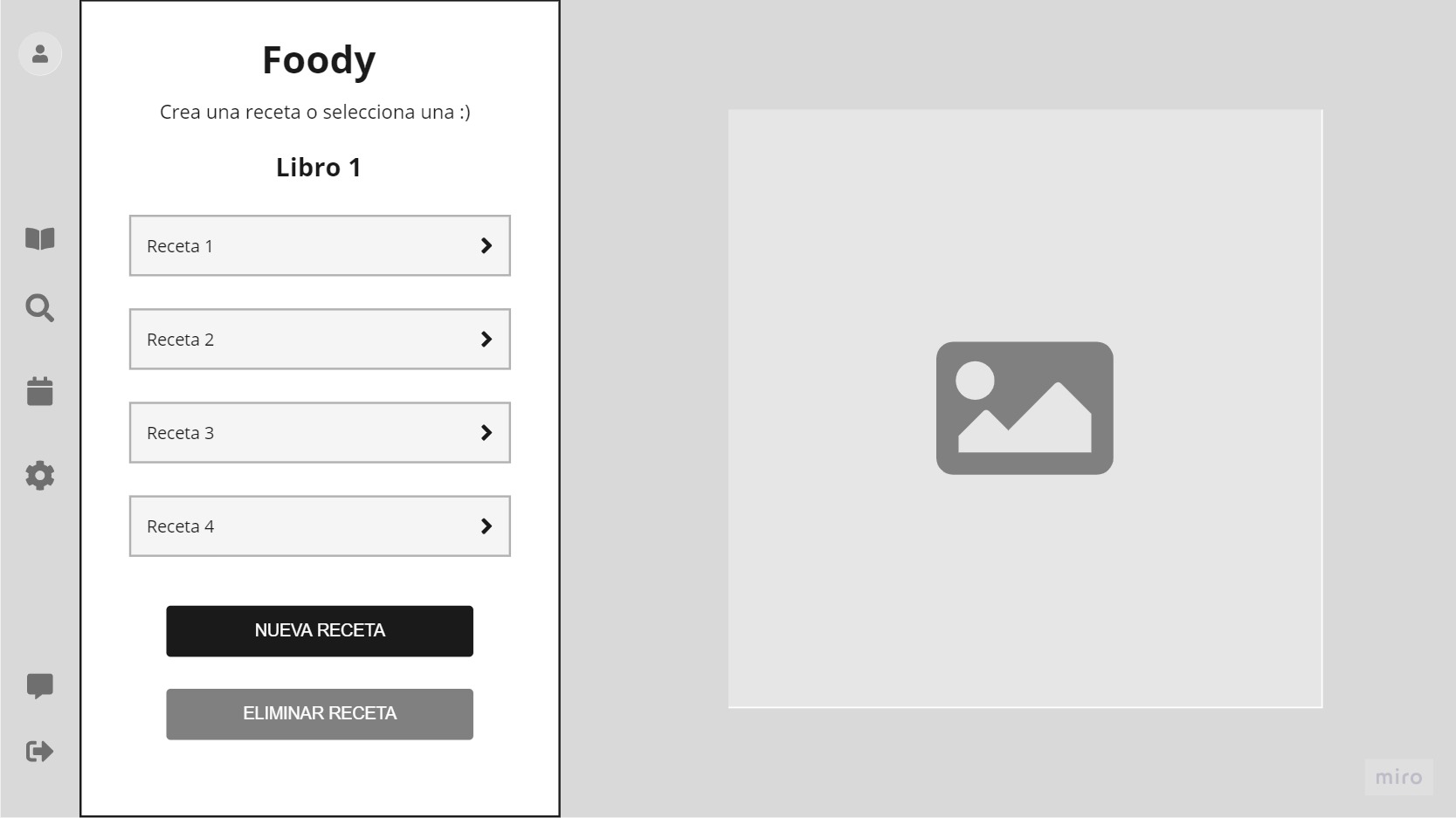
|  |  |
| --- | --- |
| **Código: HU-12** | **Nombre:** Mostrar feedback visual en caso de éxito o error. |
| **Descripción:**  **COMO:** Usuario  **QUIERO:** poder ser notificado en cualquier caso de error o éxito.  **PARA:** evitar futuros errores. | |
| **Criterios de Aceptación :**   * **C01:** El usuario debe ser notificado cuando se produce un error. * **C02:** El sistema debe mantener informado al usuario en caso de éxito o error. | |

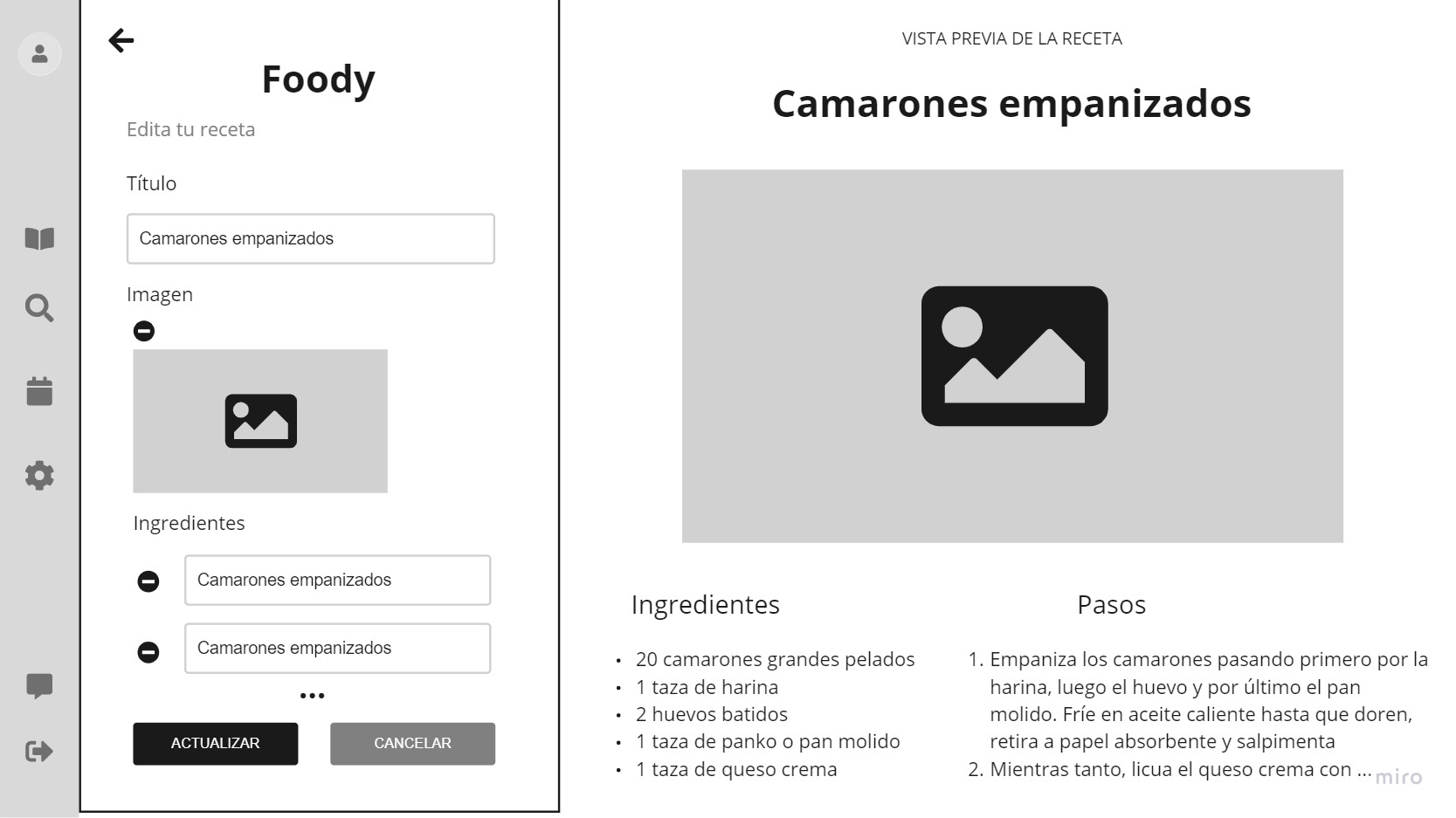
**f. Interfaz gráfica de usuarios de los Casos de Uso relevantes**

1. **Mostrar la página de inicio**
2. **Mostrar la página de registro**



1. **Mostrar la página de inicio de sesión**
2. **Mostrar página de libros de cocina**



1. **Mostrar página de edición de recetas**

**g. Sección de restricción**

**Normativas**

**● Licenciamiento**

No existe regulación de relevo para Foody . En cuanto al programa a emplear, no es forzoso lograr licencia para el uso del Node.js Framework, ya que es gratuito y sin restricciones de uso. Para el uso del MySQL también es un instrumento exento y gratuito sin restricciones de uso.

**Estándares**

**● UML**

Todos los entregables utilizados para la comunicación entre los miembros del equipo de desarrollo y los usuarios, y la documentación correspondiente requerida para desarrollar Foody**,** se basan en el Lenguaje Unificado de Modelado (UML).

**●**  Foody será desarrollado en el lenguaje de PHP , el cual se complementará con el entorno de desarrollo Node.js.

● El motor de base de datos a utilizar será el MySQL.

● Las herramientas de modelado para el desarrollo del sistema son el “IBM Rational Rose Enterprise Edition” y el “Bizagi Process Modeler” para el diagrama de actividades de los procesos.

**Soporte**

**●** Foody tendrá un sostenimiento escalonado en el cual se podrán llevar a cabo modificaciones en pro de alistar nuevas funcionalidades y/o eliminaciones las cuales estarán orientadas a mejorar las interacciones entre usuario-sistema.

**h. Sección de Calidad**

### Usabilidad

Las interfaces de Foody han sido desarrolladas para ser amigables con el usuario ya que incluyen gráficos y elementos interactivos para su mayor entendimiento en cada una de estas,pues nos vemos en la necesidad de ahorrar el tiempo de aprendizaje en el manejo del sistema para nuestros usuarios.

### Eficiencia

El sistema tendrá una respuesta inmediata a las acciones del usuario, ya que nuestro sistema no abarca demasiadas funcionalidades complejas que afecten la navegación por parte del usuario.

**Seguridad**

El sistema permitirá el uso de sus distintas funcionalidades dependiendo del tipo de cuenta con la que el usuario acceda al sistema, su ingreso se validará a través de su usuario y contraseña. Por lo tanto no puede haber filtro de información de un usuario a otro ya que cualquier acción realizada por uno de estos será registrada en el sistema como un historial junto con los datos personales.

### Confiabilidad

El sistema siempre validará los datos ingresados y mostrará mensajes indicando la posible solución en caso de presentar errores del sistema.

En cuanto a los formularios (de registro, login, creación de receta y organizador de eventos) se han restringido la digitación de ciertos caracteres para asegurar la validación de los datos a la hora de ser guardados en el sistema. En caso de que sucedan errores en el sistema, se enseñarán mensajes indicando los detalles de estos errores para que el usuario tome las medidas adecuadas ante estos.

### Mantenimiento

El mantenimiento estará regido de acuerdo a las necesidades de los usuarios y los posibles fallos que surjan y que no se hayan identificado. Debido a que el sistema no es de gran envergadura y está orientado a ser un sitio web por lo tanto su mantenimiento futuro no tendrá muchas dificultades.

**Estándares**

Se tendrá un estándar para el nombre de los formularios, clases, variables, métodos y códigos.

1. **Vista de Lógica**

**2.1 Arquitectura lógica de la aplicación.**

**Arquitectura de 3 capas**

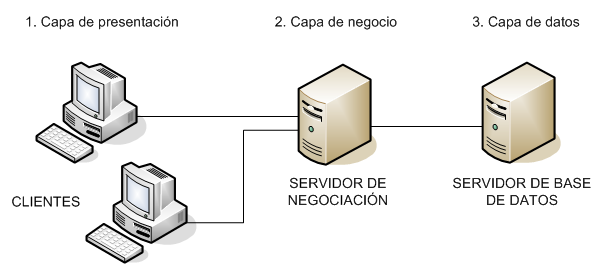
La aplicación WEB se basará en una Arquitectura de tres capas (presentación,

aplicación y persistencia). La utilización de esta arquitectura se debe a que los

distintos niveles son independientes unos de otros de manera que, por ejemplo,

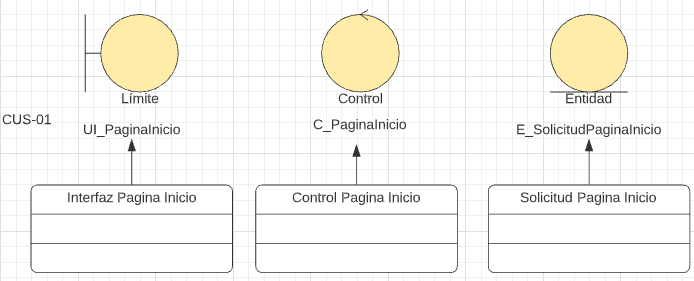
se puede cambiar fácilmente el comportamiento de las clases en el nivel de

aplicación sin que ello influya en las otras capas.

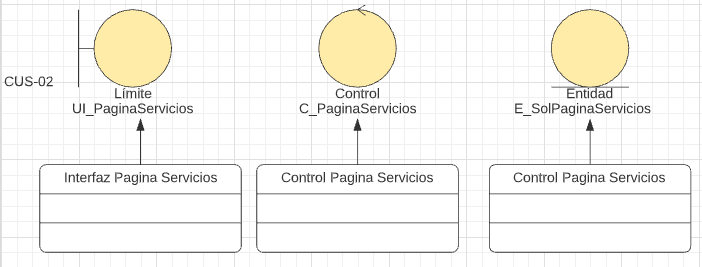
****

**2.2 Identificación de las clases de diseño por CUS**

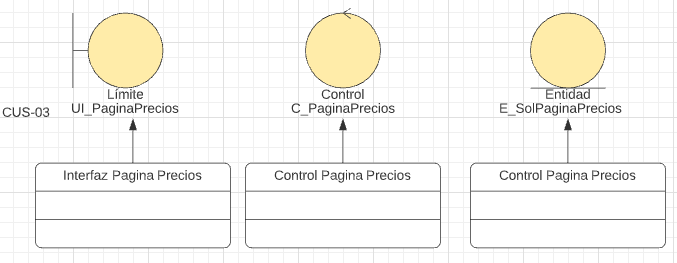
**CUS-01:** Mostrar página de inicio



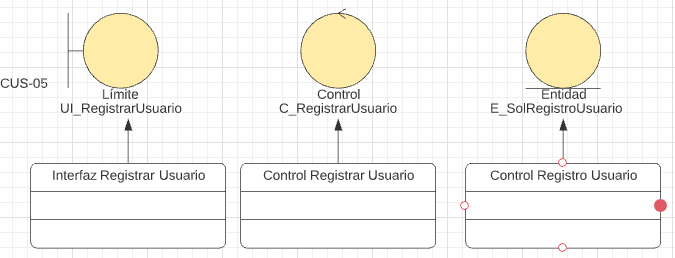
**CUS-02:** Mostrar página de servicios

****

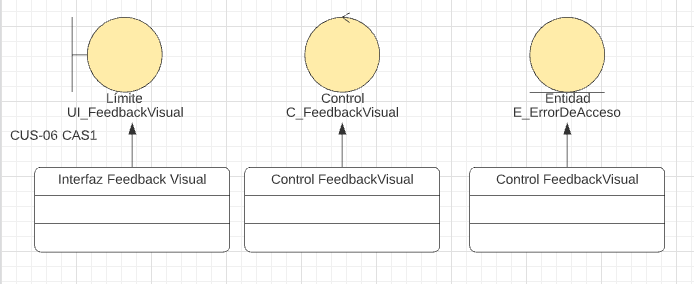
**CUS-03:** Mostrar página de precios

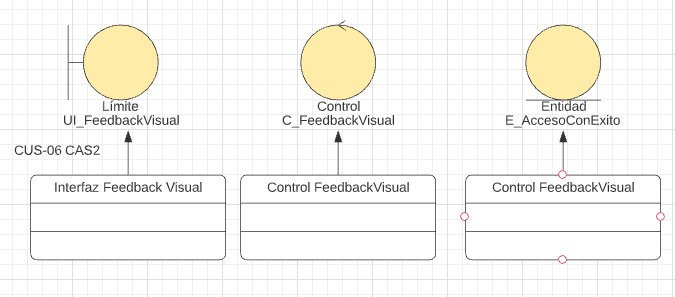


**CUS-05:** Registrar usuarios con correo y contraseña

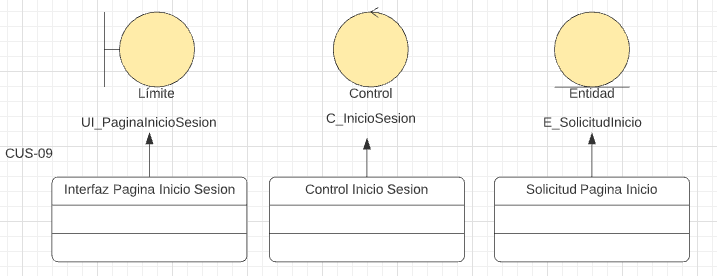


**CUS-06:** Mostrar feedback visual en caso de éxito o error

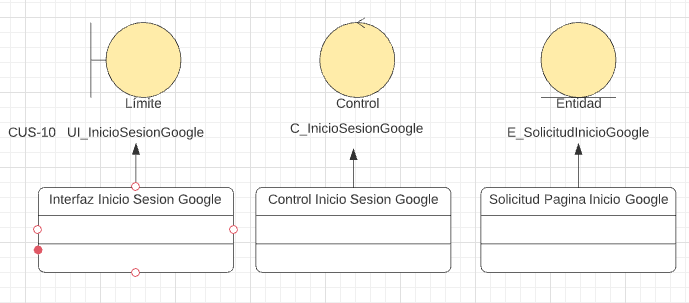




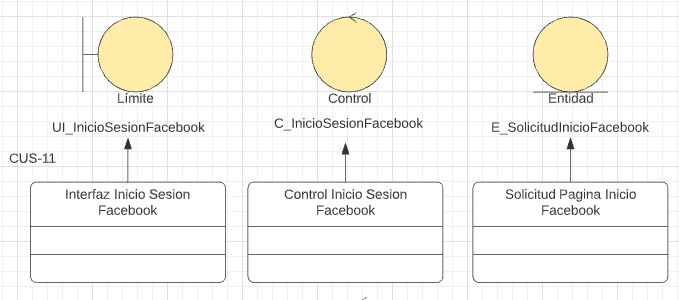
**CUS-09:** Iniciar sesión de usuario con correo y contraseña.



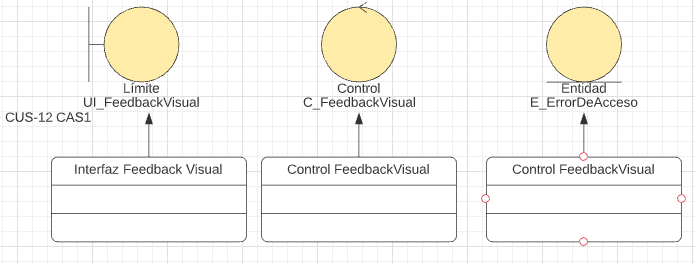
**CUS-10:** Permitir el inicio de sesión con Google.

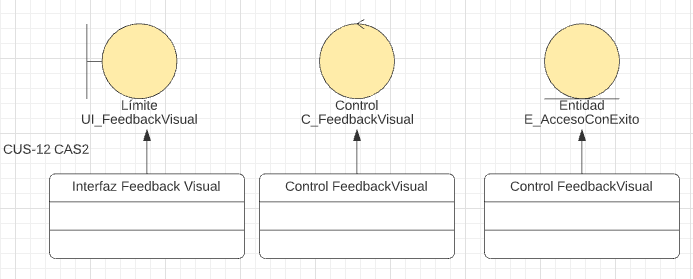


**CUS-11:** Permitir el inicio de sesión con Facebook.



**CUS-12:** Mostrar feedback visual en caso de éxito o error.





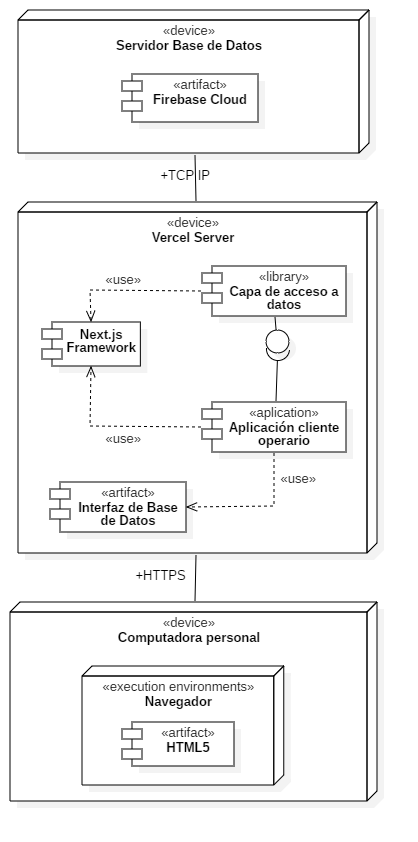
**2.3 Diagrama de Secuencias de las clases de Diseño por CUS**

**2.4 Diagrama de Subsistemas**

**2.5 Agrupación de clases por Subsistemas**

**2.6 Asignación de Operaciones a las clases de Diseño y Diagrama de clases de Diseño**

1. **Vista de Despliegue**
   1. **Diagrama de Despliegue**

****

* 1. **Características técnicas**
* **Computadora Personal**

Dispositivo que corresponde a un usuario, ya sea operario o cliente, el cual consumirá el servicio del aplicativo web de manera remota del servidor de Vercel.

1. **Vista de Implementación**

En esta vista de implementación se presenta el sistema en términos de componentes, es decir ficheros de código fuente. Nos enfocaremos en la organización de los módulos de software.

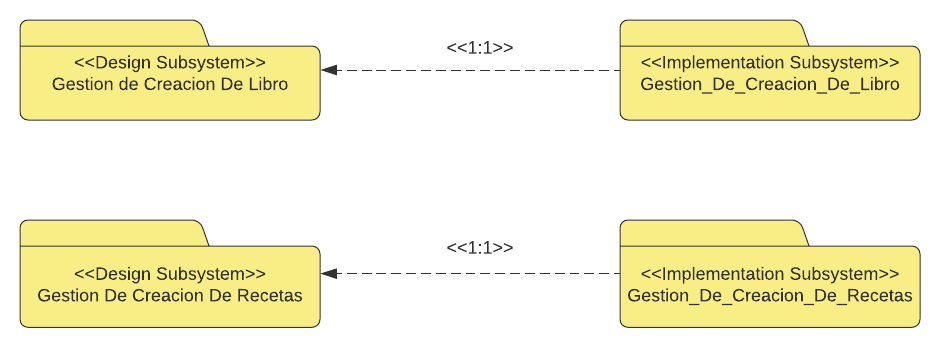
Se ha decidido separar en tres módulos Servicio al cliente (abarca cada una de las

actividades del paquete servicio al cliente), Gestión de producto (abarca cada una de las actividades del paquete de Gestión de producto) y Gestión de tareas de operario (abarca cada una de las actividades del paquete Gestión de tareas de operario).

En esta vista también se explicara cual es la relación entre los componentes y la clase de diseño de cada módulo.

1. **Clases agrupadas en componentes que forman parte de cada Subsistema de implementación**

Actividad implementar un subsistema

****

**Subsistema de Implementación: Gestión de Creación de Libro**

**Componente usuario**

* Usuario(titular)
* Interfaz usuario
* Interfaz registrar datos
* Interfaz seleccionar correo

**Componente Libro**

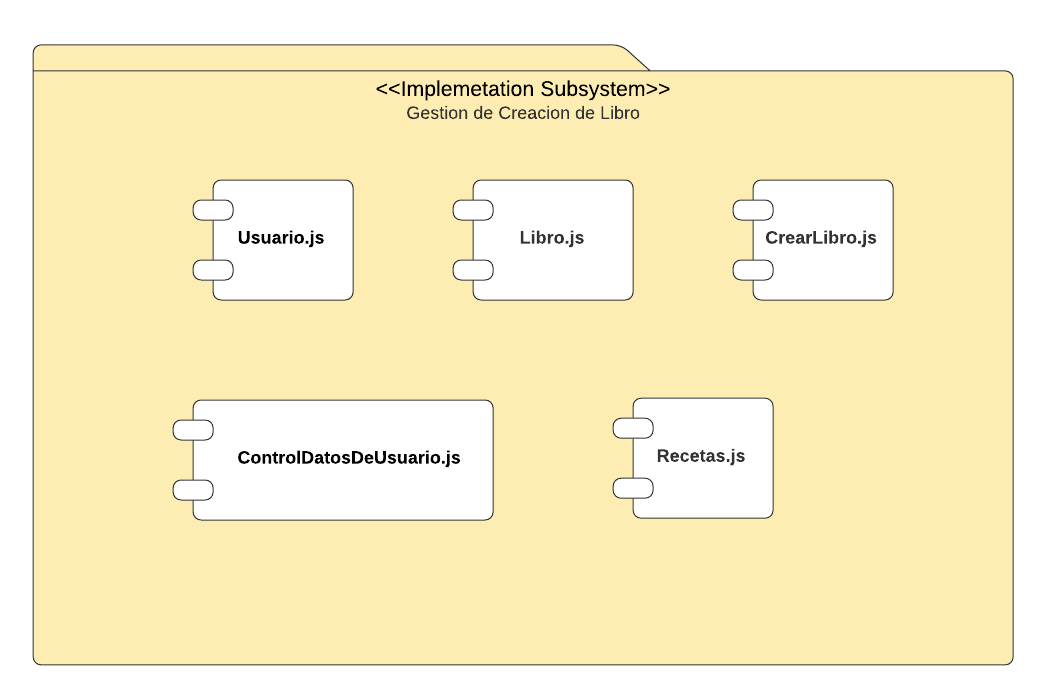
* Libro
* Interfaz nuevo libro
* Interfaz eliminar libro

**Componente Recetas**

* Recetas
* Interfaz nueva receta
* Interfaz eliminar receta

**Componente control datos usuario**

* Control login



**Subsistema de Implementación: Gestión de Creación de Receta**

**Componente Gestor completar solicitud**

* Control completar solicitud

**Componente Receta**

* Interfaz mostrar recetas
* Interfaz completar receta
* Interfaz eliminar receta
* Interfaz nuevo libro

**Componente Libro**

* Interfaz mostrar libro
* Interfaz nuevo libro
* Interfaz eliminar receta

**Componente Controlador de usuario**

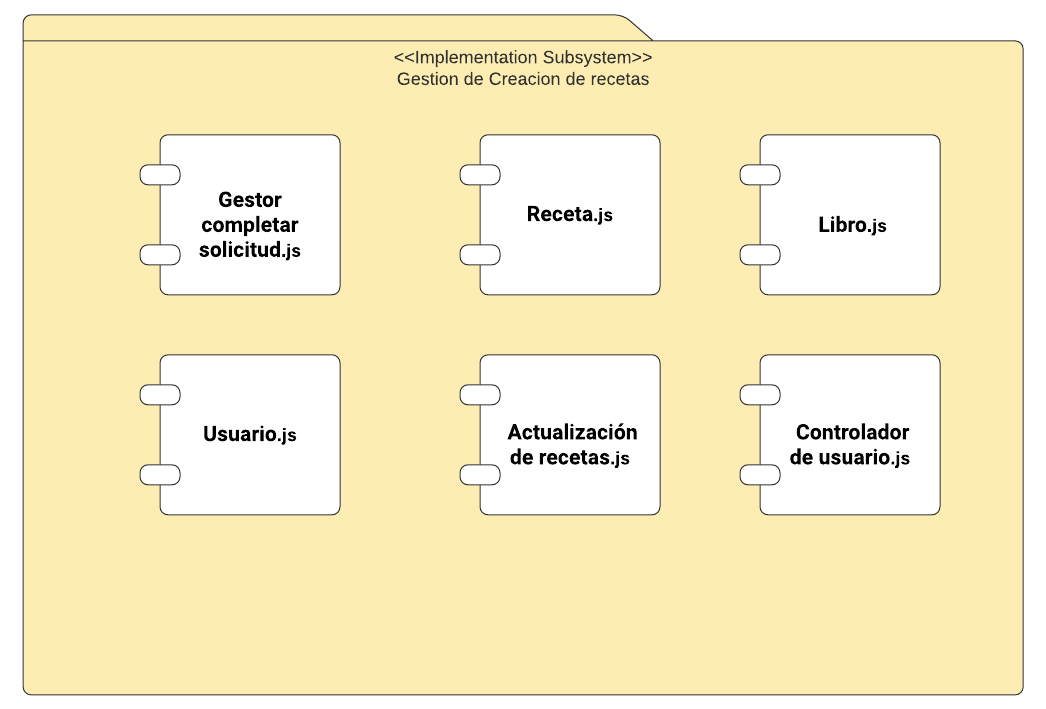
* Control login

**Componente Usuario**

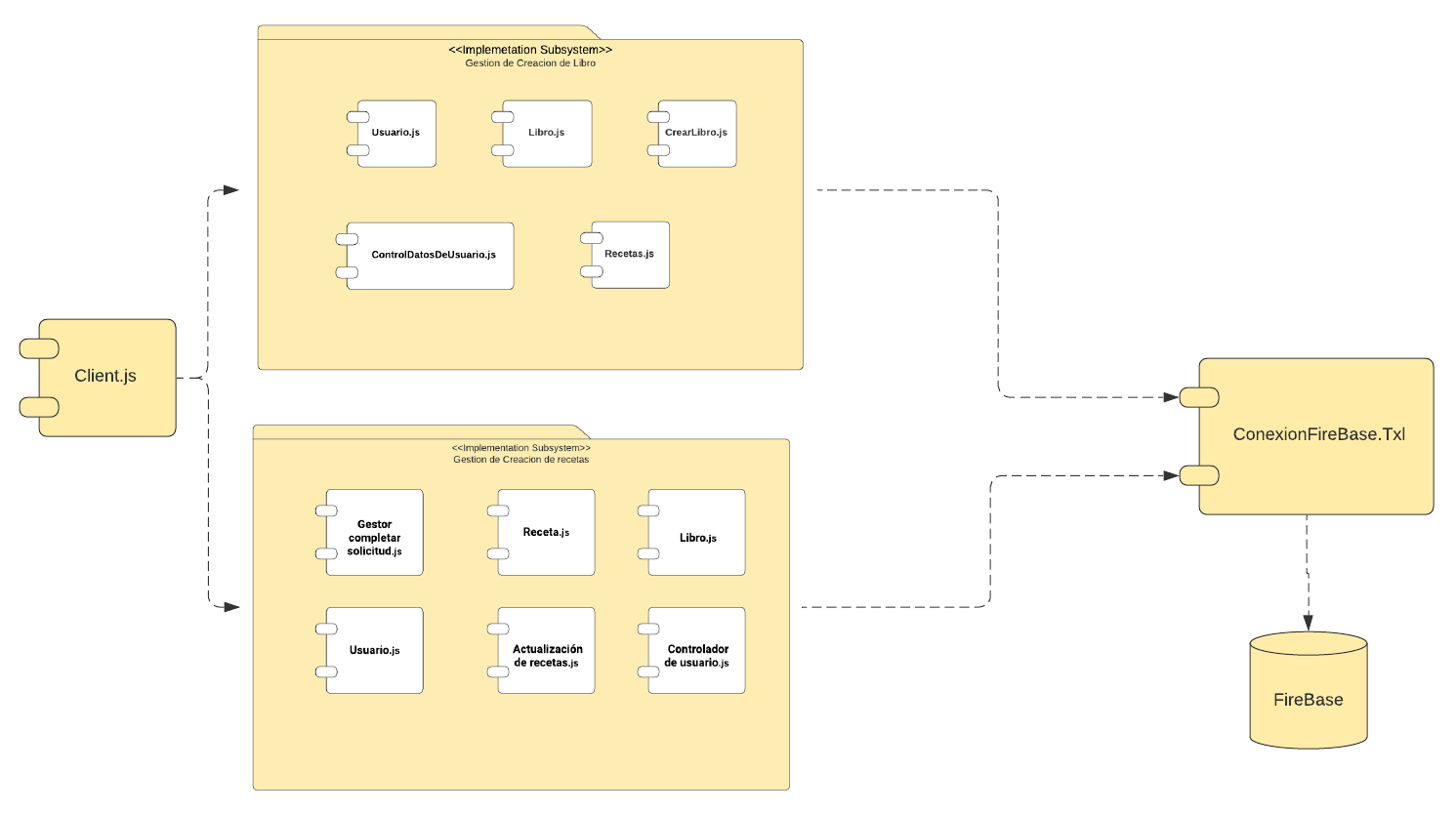
* Interfaz registrar datos

C**omponente Actualización de recetas**

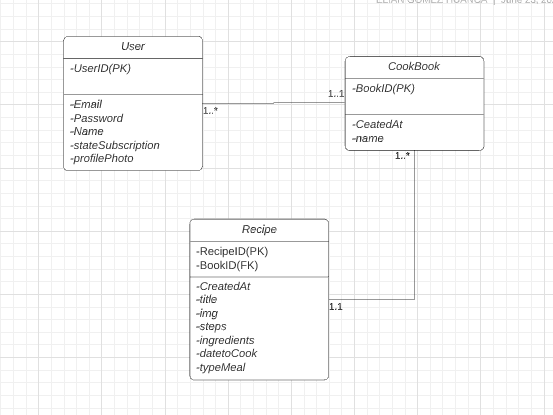
- Gestor confirmación de actualización de recetas



1. **Diagrama de Componentes.**

****

1. **Modelo relacional de BD.**

****

