|  |
| --- |
| **MenuGo: Aplicación móvil para restaurantes** |
|  |
|  |
|  |

|  |
| --- |

**Integrantes:**

-Matías Cornejo

-Benjamín Gatica

-Joaquín González

-Diego Olmos

**Contenido**

[**1.- Introducción 3**](#_heading=h.6awp11t9r88w)

[**2.-Problemática 3**](#_heading=h.u2xvxd4e9rlh)

[**3.- Etapas del ciclo de vida del software 3**](#_heading=h.4ar7xgcapj3k)

[**4.- Alcances y objetivos del proyecto 3**](#_heading=h.7x5mwbm77w8b)

[**5.- Organización a la que pertenece la problemática 4**](#_heading=h.j5cqaejwhblw)

[**6.- Roles y responsabilidades de trabajo 4**](#_heading=h.mb2hannefdqw)

[**7.- Metodologías de gestión para el proyecto 4**](#_heading=h.3prqmzouzmhc)

[**8.- Requerimientos iniciales de alto nivel 5**](#_heading=h.rq6bv2922mv9)

[**9.- Requerimientos Funcionales y No Funcionales 5**](#_heading=h.i0dwv09m35q3)

[Requisitos Funcionales 5](#_heading=h.ntm4rzzh4ms3)

[Requisitos No Funcionales 7](#_heading=h.czd1cx6omtke)

[**10.- Desarrollo del proyecto con: 9**](#_heading=h.e06bgzitd8aw)

[**11.- Arquitectura y diseño del sistema (diagrama UML) 11**](#_heading=h.bbe4lxfj0oge)

[**12.- Diagramas 12**](#_heading=h.t58a8ze6i2b5)

[**13.- Normas a aplicarse para el diseño de software según necesidades del cliente 15**](#_heading=h.jvgr35dywrbz)

[**14.- Prototipo 16**](#_heading=h.h98ffgtxsgdc)

**Ficha del documento**

| **Fecha** | **Revisión** | **Autor** | **Modificación** |
| --- | --- | --- | --- |
| **17-05-25** | **✔️** | **Benjamin Gatica** | **Inicio del informe** |
| **19-05-25** | **✔️** | **Matías Cornejo** | **Revisión final** |

**1.- Introducción**

Este documento detalla la propuesta de desarrollo del sistema “MenuGo”, una aplicación móvil pensada para transformar digitalmente la experiencia en restaurantes. El objetivo es mejorar el servicio al cliente, agilizar procesos y facilitar la gestión por parte de los establecimientos gastronómicos.

**2.-Problemática**

Muchos restaurantes, especialmente medianos y pequeños, presentan sistemas de atención y pedido anticuados, lo que resulta en tiempos de espera largos, errores en pedidos y una experiencia de cliente deficiente. La falta de digitalización también limita su capacidad para competir en un mercado moderno y tecnificado.

**3.- Etapas del ciclo de vida del software**

* **Análisis:** Recolección de requisitos funcionales y no funcionales.
* **Diseño:** Arquitectura modular, diseño de UX/UI y modelado UML.
* **Desarrollo:** Implementación del backend y frontend, integración de pagos.
* **Testing:** Pruebas de funcionalidad, carga y seguridad.
* **Mantenimiento:** Soporte post-lanzamiento, actualizaciones, mejora continua.

**4.- Alcances y objetivos del proyecto**

**Alcances:**Desarrollar una app móvil multiplataforma (iOS/Android) que permite visualizar menús, realizar pedidos, pagar, seguir el estado del pedido y gestionar la experiencia del usuario y del restaurante.

**Objetivos:**

* Modernizar la interacción cliente-restaurante.
* Reducir tiempos de espera.
* Aumentar la eficiencia operativa.
* Mejorar la satisfacción del cliente.

**5.- Organización a la que pertenece la problemática**

La problemática pertenece al sector gastronómico, incluyendo restaurantes, cadenas de comida rápida y locales emergentes que buscan modernizar sus procesos sin grandes alteraciones. Estas organizaciones enfrentan dificultades como la gestión manual de pedidos e inventarios, baja eficiencia operativa y falta de acceso a tecnología accesible. El proyecto propone soluciones funcionales y actualizadas que les permitan optimizar sus operaciones y mejorar su competitividad sin comprometer su negocio.

**6.- Roles y responsabilidades de trabajo**

| **Nombre Integrante del Equipo** | **Rol Definido** |
| --- | --- |
| *Benjamin Gatica* | *Product Owner* |
| *Matias Cornejo* | *Scrum Master* |
| *Diego Olmos* | *Developer* |
| *Joaquin Gonzalez* | *QA* |
|  |  |

**7.- Metodologías de gestión para el proyecto**

Se utilizaron metodologías de trabajo en conjunto, tales como desarrollar un espacio en Miro para la correcta organización del equipo y realizar los trabajos de la forma más eficiente posible, como también se trabajó via Discord para mantener la comunicación al momento de ir desarrollando y avanzando en el proyecto, para que así cada integrante hiciera su labor correspondiente sin altercados entre los mismos.

**8.- Requerimientos iniciales de alto nivel**

Para responder a las necesidades actuales del sector gastronómico, se plantea una propuesta tecnológica que moderniza y optimiza la operación de los restaurantes. Esta solución incluye:

* **Disponibilidad multiplataforma**: acceso desde computadoras, tablets o smartphones para mayor comodidad.
* **Visualización y personalización de menús**: adaptación del contenido del menú según el negocio y preferencias del cliente.
* **Pedidos en tiempo real**: gestión rápida y precisa de órdenes para mejorar el servicio.
* **Métodos de pago digitales**: opciones seguras y variadas para facilitar las transacciones.
* **Administración por parte del restaurante**: control total sobre productos, pedidos y estadísticas operativas.

**9.- Requerimientos Funcionales y No Funcionales**

**Requisitos Funcionales**

**RF001** **Tipo:** Funcional  
 **Actor relacionado:** Administrador  
 **Descripción:** El sistema debe verificar si el usuario tiene alguna deuda pendiente o si ha superado los límites de libros prestados.  
 **Criterio de aceptación:** Usuario con impedimento recibe una notificación que indica que no se le puede prestar un libro.  
 **Estado:** Aprobado

**RF002** **Tipo:** Funcional  
 **Actor relacionado:** Administrador  
 **Descripción:** El sistema debe permitir verificar si un libro está disponible, reservado, en uso o dañado.  
 **Criterio de aceptación:** El sistema debe mostrar en tiempo real el estado del libro.  
 **Estado:** Aprobado

**RF003** **Tipo:** Funcional  
 **Actor relacionado:** Administrador  
 **Descripción:** La aplicación debe permitir rastrear la ubicación física del libro dentro de la biblioteca.  
 **Criterio de aceptación:** El sistema permite buscar la ubicación exacta del libro.  
 **Estado:** Solicitado

**RF004** **Tipo:** Funcional  
 **Actor relacionado:** Administrador  
 **Descripción:** Permitir a los administradores o al sistema cambiar el estado del libro en tiempo real.  
 **Criterio de aceptación:** El cambio de estado se refleja instantáneamente en la plataforma.  
 **Estado:** Aprobado

**RF005** **Tipo:** Funcional  
 **Actor relacionado:** Administrador  
 **Descripción:** El sistema debe vincular el ID del libro con el ID del usuario al momento del préstamo.  
 **Criterio de aceptación:** El seguimiento del préstamo se realiza mediante esta vinculación.  
 **Estado:** Aprobado

**RF006** **Tipo:** Funcional  
 **Actor relacionado:** Administrador  
 **Descripción:** Permitir el registro de usuarios y administradores con asignación de roles y permisos.  
 **Criterio de aceptación:** Registro correcto y asignación de permisos adecuada.  
 **Estado:** Aprobado

**RF007** **Tipo:** Funcional  
 **Actor relacionado:** Administrador  
 **Descripción:** Los administradores pueden registrar nuevos libros con detalles relevantes.  
 **Criterio de aceptación:** Registro inmediato de la información del libro.  
 **Estado:** Aprobado

**RF008** **Tipo:** Funcional  
 **Actor relacionado:** Administrador y usuarios  
 **Descripción:** Los usuarios pueden programar la devolución del libro, manual o automáticamente.  
 **Criterio de aceptación:** El sistema debe generar recordatorios automáticos.  
 **Estado:** Aprobado

**RF009** **Tipo:** Funcional  
 **Actor relacionado:** Administrador y usuarios  
 **Descripción:** Acceso al historial de préstamos para cada usuario.  
 **Criterio de aceptación:** El sistema muestra títulos, fechas de préstamo y devolución.  
 **Estado:** Aprobado

**RF010** **Tipo:** Funcional  
 **Actor relacionado:** Administrador  
 **Descripción:** Utilizar alertas y mensajes personalizados para notificar devoluciones.  
 **Criterio de aceptación:** El sistema genera recomendaciones de libros basadas en el historial.  
 **Estado:** Aprobado

**Requisitos No Funcionales**

**RNF001** **Tipo:** No Funcional  
 **Actor relacionado:** Administrador  
 **Descripción:** La aplicación debe responder en menos de 3 segundos.  
 **Criterio de aceptación:** Carga y acceso en menos de 3 segundos.  
 **Estado:** Aprobado

**RNF002** **Tipo:** No Funcional  
 **Actor relacionado:** Administrador  
 **Descripción:** Capacidad para soportar 1.000 usuarios simultáneamente.  
 **Criterio de aceptación:** Funcionamiento fluido sin pérdida de rendimiento.  
 **Estado:** Aprobado

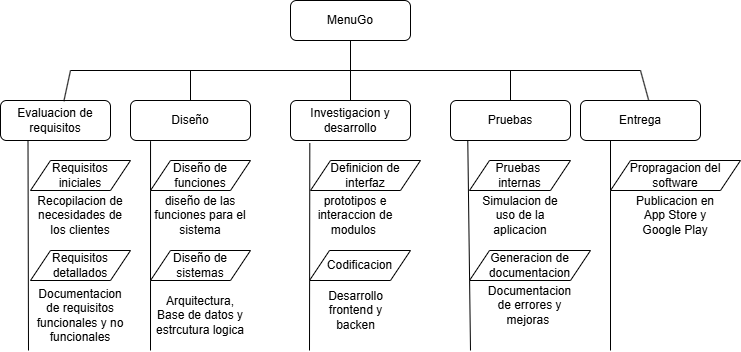
**RNF003** **Tipo:** No Funcional  
 **Actor relacionado:** Administrador y usuarios  
 **Descripción:** Disponibilidad del 99,9% del tiempo (máximo 24 horas de inactividad anual).  
 **Criterio de aceptación:** Se cumple el porcentaje de disponibilidad.  
 **Estado:** Solicitado

**RNF004** **Tipo:** No Funcional  
 **Actor relacionado:** Administrador  
 **Descripción:** Interfaz fácil de usar.  
 **Criterio de aceptación:** Navegación intuitiva y sencilla para todos los usuarios.  
 **Estado:** Aprobado  
 **RNF005** **Tipo:** No Funcional  
 **Actor relacionado:** Administrador  
 **Descripción:** Cumple con estándares internacionales de accesibilidad web.  
 **Criterio de aceptación:** Sin barreras para la interacción de los usuarios.  
 **Estado:** Aprobado

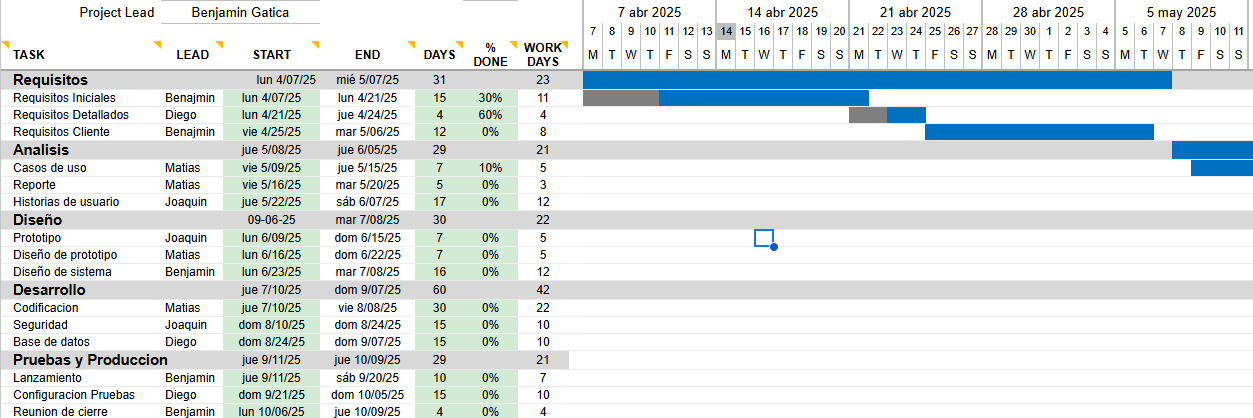
**RNF006** **Tipo:** No Funcional  
 **Actor relacionado:** Administrador  
 **Descripción:** Incluye manuales y tutoriales para los usuarios.  
 **Criterio de aceptación:** Documentación clara, accesible y efectiva.  
 **Estado:** Aprobado

**10.- Desarrollo del proyecto con:**

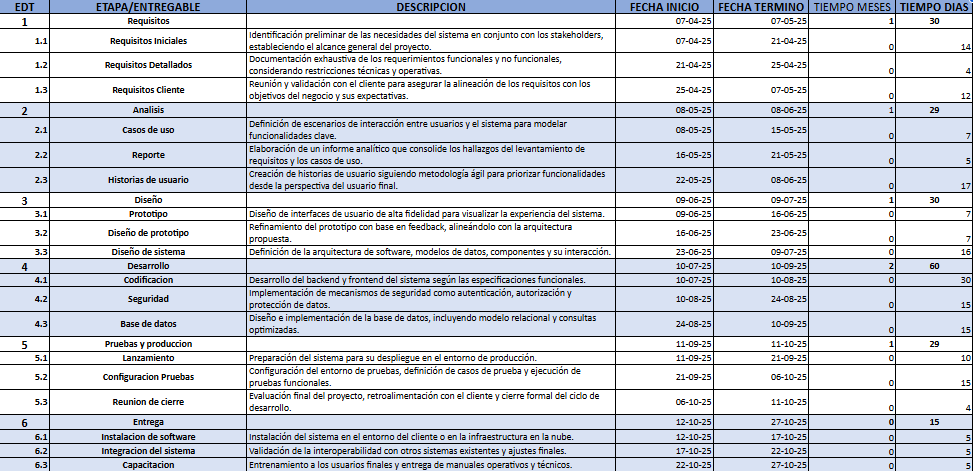
**-EDT**

****

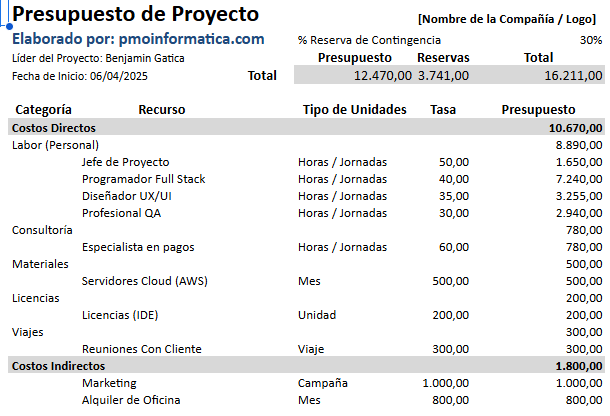
**-Carta Gantt**



**-Cronograma**



**-Recursos**



**11.- Arquitectura y diseño del sistema (diagrama UML)**

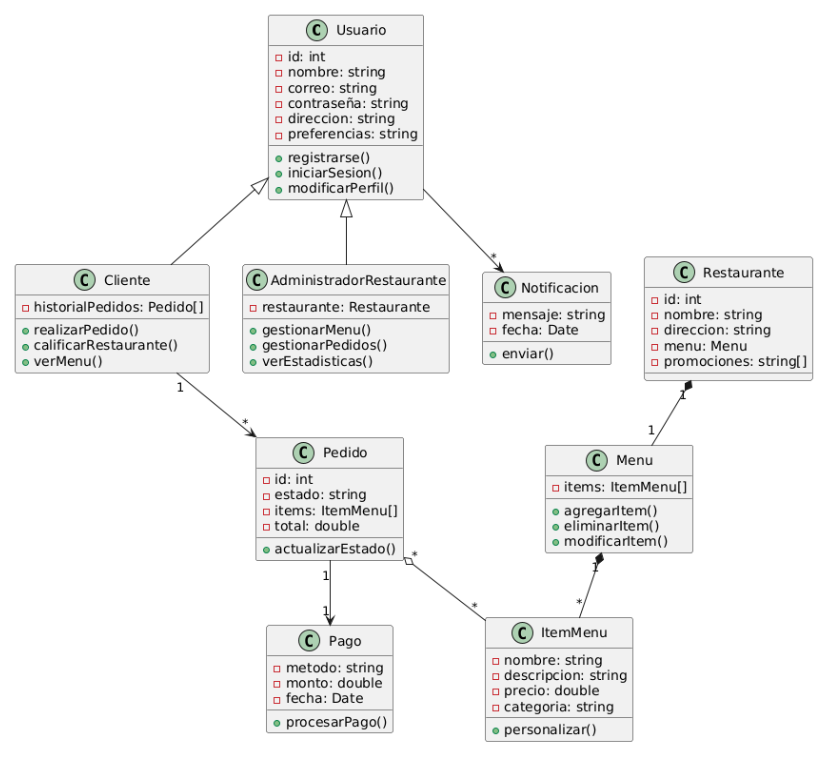
Arquitectura General

El sistema MenuGo se basa en una arquitectura cliente-servidor, donde la aplicación móvil (cliente) interactúa con una API RESTful centralizada que gestiona la lógica de negocio y los datos. La solución se divide en tres capas principales:

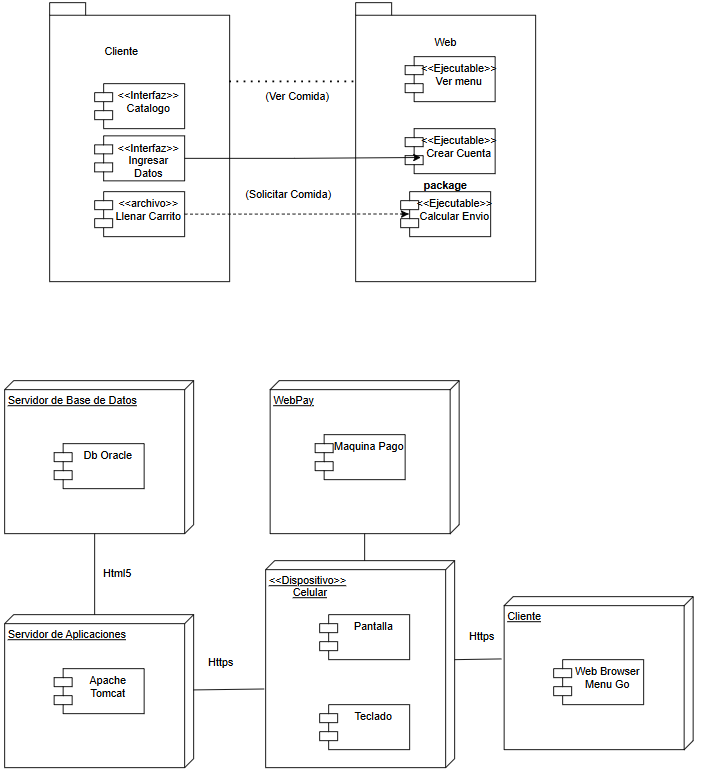
* Capa de presentación: Aplicación móvil híbrida para iOS y Android.
* Capa de negocio: Backend con microservicios para pedidos, menús, autenticación, pagos, y administración.
* Capa de datos: Base de datos relacional para almacenamiento persistente.

**12.- Diagramas**

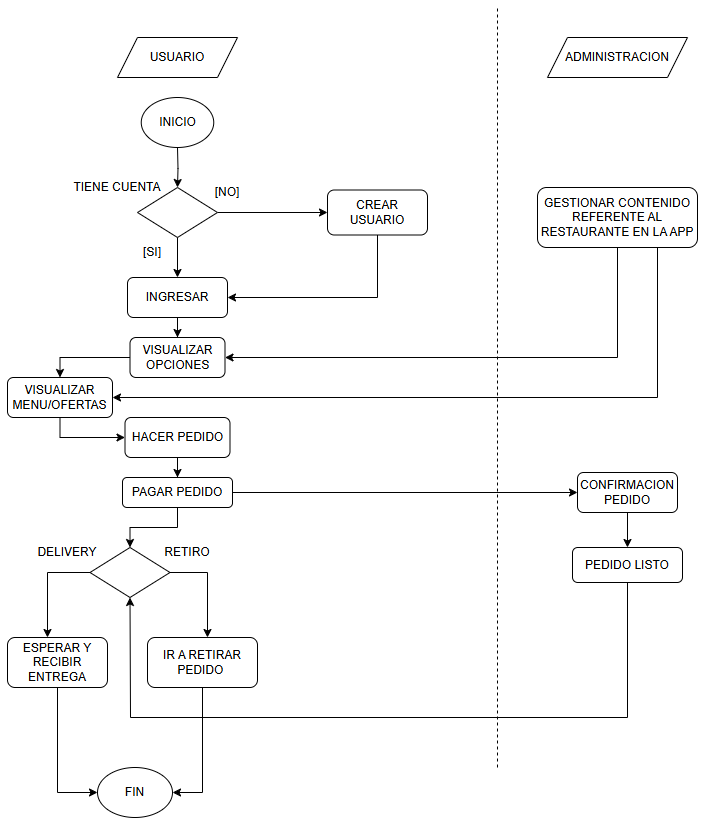
-Diagrama de Clases:

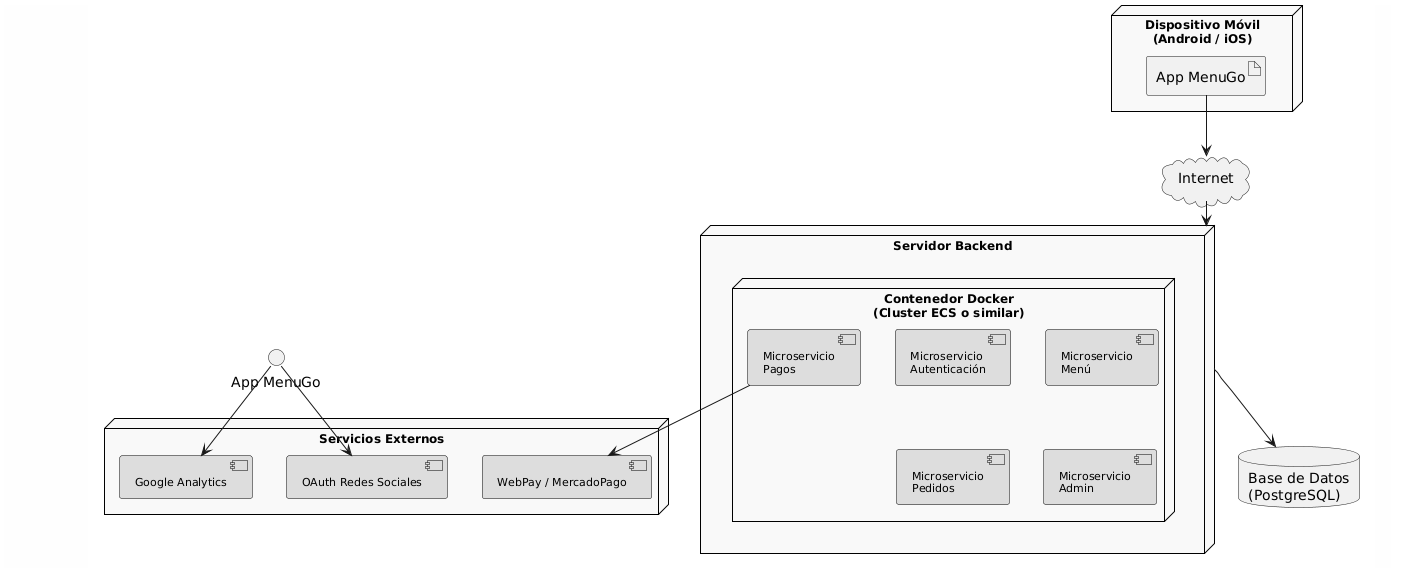


-Diagrama de componentes:

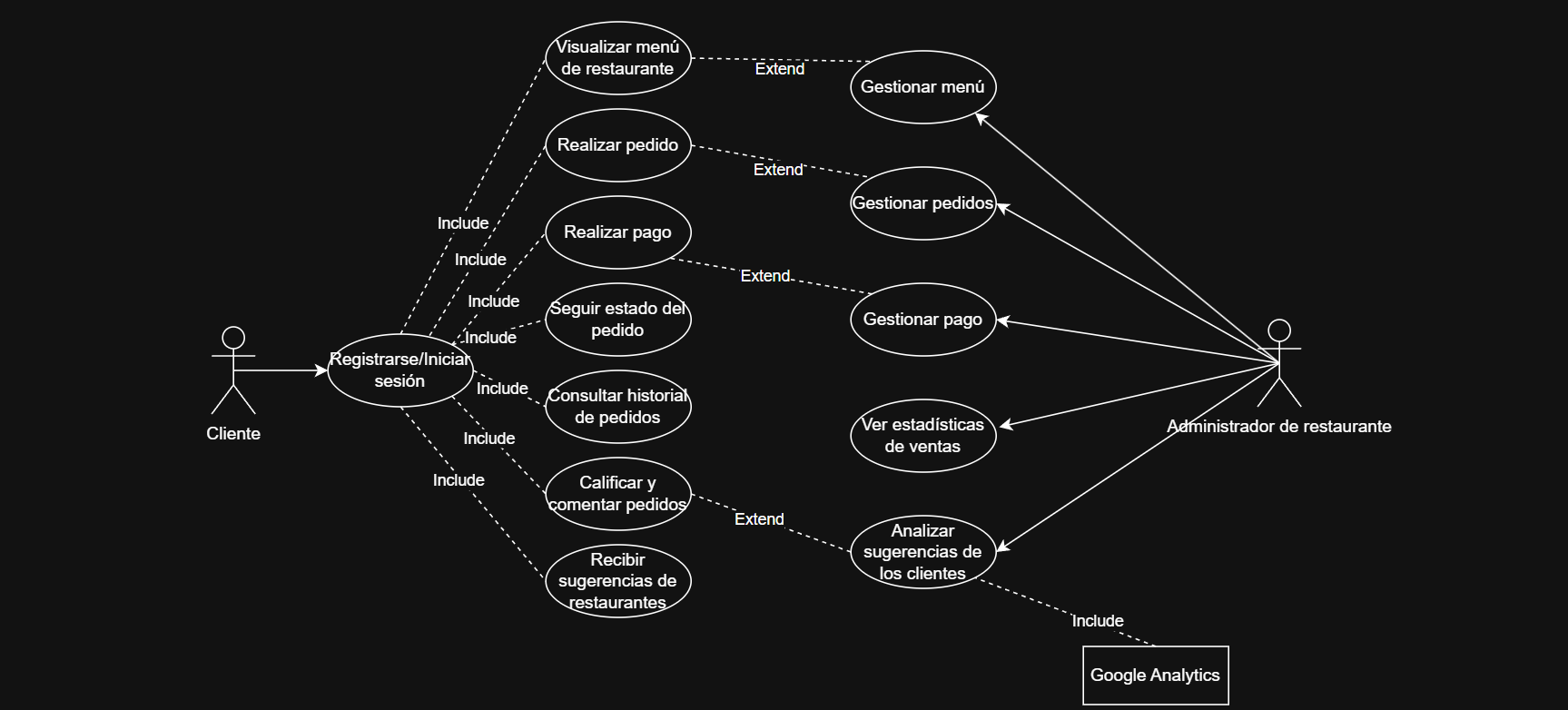


-Diagrama de actividad:



-Diagrama de despliegue:

-Diagrama de casos de uso:

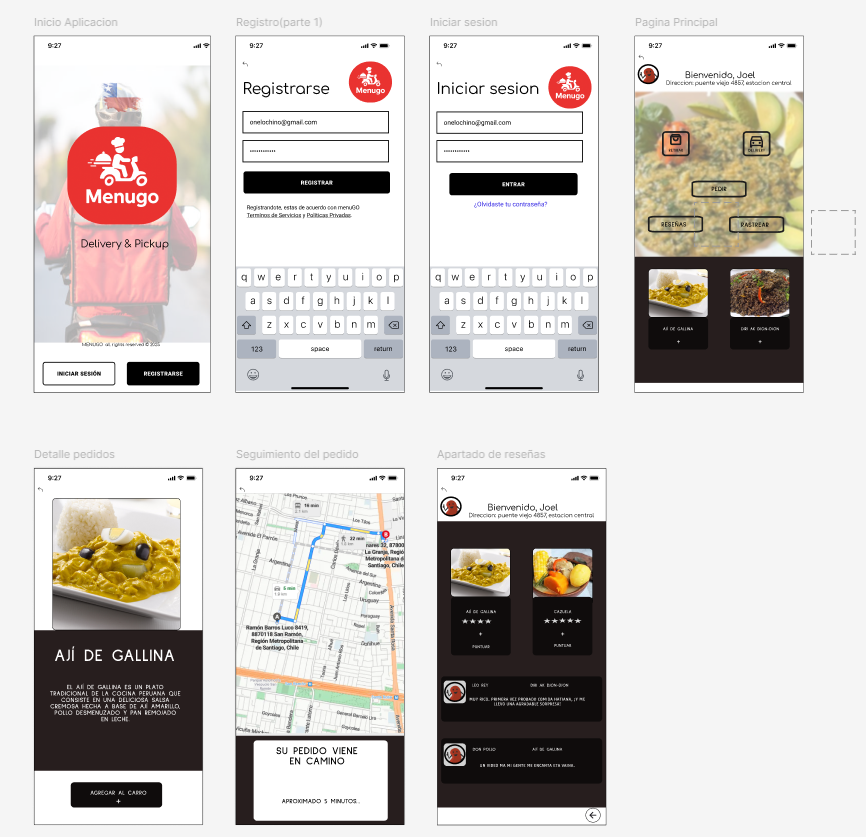


**13.- Normas a aplicarse para el diseño de software según necesidades del cliente**

Para asegurar que el software cumpla con lo que necesitan los clientes/restaurantes, se seguirán normas y buenas prácticas que garanticen que sea fácil de usar, seguro y funcione correctamente. Se tendrá en cuenta que el sistema pueda utilizarse en diferentes dispositivos, que sea accesible para todo tipo de usuarios y que proteja la información de los clientes, especialmente en los pagos. Además, se cuidará que el diseño sea claro, ordenado y práctico, permitiendo una buena experiencia tanto para los clientes como para quienes administran el restaurante.

**14.- Prototipo**

https://www.figma.com/proto/owqamFdv8KLl5LitZNfjBe/MENUGO?node-id=30-85&p=f&t=aN0tgiuqoe3OgiaK-1&scaling=scale-down&content-scaling=fixed&page-id=0%3A1&starting-point-node-id=30%3A85



**15.- Diseño Interactivo**

Hemos creado una simulación de cómo funciona la aplicación de MenuGo de manera interactiva , a través de la web de diseño de interfaz por figma, mostrando al menos 5 funcionalidades interactivas las cuales son las siguientes: