

APLICACIÓN MAC RESTAURANT PARA RESTAURANTES DE LA CIUDAD DE POPAYÁN. DEFINICIÓN DE LA ARQUITECTURA.

*Autores:
Kevith Felipe Bastidas
Andrés Ordoñez
Whalen Stiven Caicedo
Juan David Muñoz Pasquel*

INGENIERÍA DE SOFTWARE II
UNIVERSIDAD DEL CAUCA
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES
PROGRAMA DE INGENIERÍA DE SISTEMAS

2021

Tabla de contenido

1. Resumen	1
2. Propósito	1
3. Representación de la arquitectura	1
4. Vista de requerimientos	2
4.1. Historias épicas de importancia arquitectural	3
4.2. Atributos de calidad de importancia arquitectural	4
4.3. Escenarios de calidad de importancia arquitectural	4
5. Resumen de las decisiones de arquitectura	5
6. Modelo de contexto	6
7. Vista lógica	
7.1. Vista de contenedores y componentes	8
7.2. Vista módulos	10
7.3. Vista de componentes y conectores	11
8. Vista de implementación	12
9. Anexos	13

1. Resumen

Las plataformas digitales son ahora una nueva opción para los restaurantes de tomar parte en la considerada transformación digital, Mac Restaurant es una nueva opción para los restaurantes de la ciudad de Popayán, esta aplicación se desarrollará con el fin de permitir a las personas de la ciudad de Popayán hacer pedidos de sus platos favoritos y disfrutar de ellos desde la comodidad de su casa, disponiendo en esta de los mejores platos de los restaurantes de la ciudad, con esta aplicación además de ofrecer los mejores platos de cada restaurante manejaremos el control de los pedidos, de las entregas, el dinero recaudado en cada día y la gestión de los platos del restaurante.

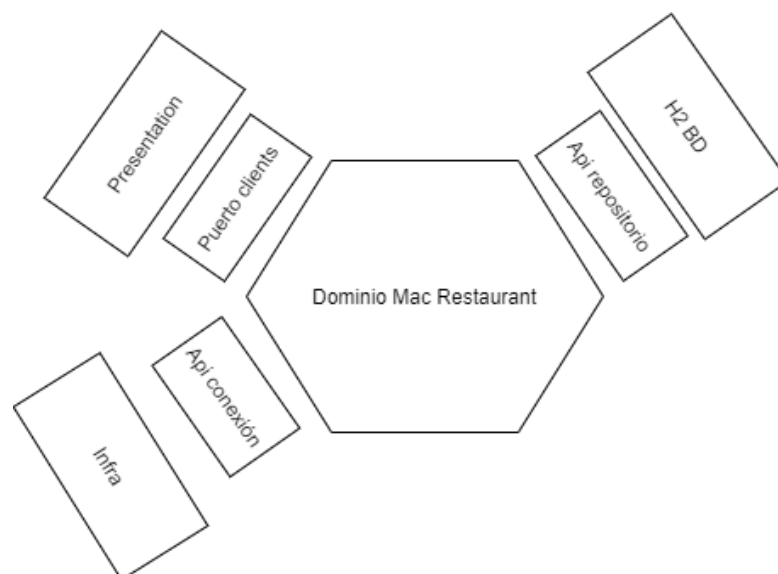
Partiendo de una necesidad común y unos requerimientos específicos; en este documento se plantea la descomposición de la arquitectura (teniendo en cuenta y/o priorizando algunas de las vistas más importantes para la descomposición y el entendimiento del sistema)

2. Propósito

El modelado y la descomposición de la aplicación Mac Restaurant, para restaurantes en Popayán han realizado en diferentes documentos; hasta el momento se cuenta con tres de ellos: Documento de Requerimientos y Documento de Descomposición de arquitectura, pensados para los diferentes tipos de público que pudieran necesitar la documentación respectiva a sus diferentes áreas competentes. El propósito de este documento es exponer la descomposición realizada de la arquitectura, priorizando algunas de las vistas e incluso, omitiendo algunas vistas que más adelante, en siguientes versiones de los documentos, podrían aparecer.

3. Representación de la arquitectura

La arquitectura utilizada fue la Hexagonal la cual se centra en mejorar la arquitectura de tres capas colocando en el centro de la aplicación el dominio o la lógica del negocio y haciendo que las demás capas sean exteriores y dependan de la lógica de negocio las cuales serán conectadas por medio de un puerto o interfaz. Haciendo que la aplicación sea fácil de modificar y la lógica de negocio no incluya nada de la lógica de presentación al igual que nada de la lógica de acceso a datos ya que si estas están separadas nos permitirán hacer cambios en la aplicación de manera más fácil, permitiéndonos tener más cohesión, mejor acoplamiento y mayor facilidad para realizar pruebas unitarias ya que la lógica de negocio no incluye lógica del acceso a datos y eso hace mucho más fácil realizar pruebas unitarias para esta capa.



A continuación, se denotan las vistas que se trabajarán a lo largo de este documento, como parte de la primera versión de descomposición de la arquitectura.

- a. **Vista de requerimientos:** La vista de requerimiento es casi obligatoria, puesto que, conforme a los requerimientos que se tengan, se tomarán las decisiones correspondientes acerca de qué patrones arquitecturales se utilizarán en base a los atributos priorizados. Es importante tener en cuenta que este documento es de descomposición arquitectural, por ende, esta vista indicará única y exclusivamente los requerimientos de importancia arquitectural.
- b. **Vista lógica:** Al hablar de subsistemas, la vista lógica se hace presente casi de inmediato, puesto que esta vista permitirá conocer una descomposición de los módulos, contenedores y componentes que presente el subsistema que se esté analizando. En este documento se hará un análisis de los diferentes subsistemas que se tocarán como parte de la primera descomposición de la arquitectura.
- c. **Vista de implementación:** Una vez se tiene claridad sobre los módulos, contenedores y componentes de los diferentes subsistemas, es posible tener una idea mucho más clara del funcionamiento de estos y su relación con el sistema en general, a través de la vista de implementación, donde se hará posible tener una idea de las dependencias necesarias a nivel de código.
- d. **Vista de despliegue:** La vista de despliegue permite tener claridad sobre la instalación que se hará, en las diferentes máquinas necesarias, de cada uno de los componentes de la aplicación.

4. Vista de requerimientos

Este documento tiene como propósito presentar la descomposición de la arquitectura, por ende, sólo se indicarán, en esta vista, algunos de los requerimientos, atributos de calidad y escenarios relevantes a la hora de tomar decisiones arquitecturales, partiendo de la existencia de atributos priorizados, historias épicas y de usuario que los respalden y una modificación de la red de atributos.

Prioridad	Código	Historia épica	Contexto
1	HE01	Yo como administrador del restaurante necesito registrar los componentes del almuerzo ejecutivo para posteriormente ofrecer a través de la plataforma el almuerzo de un día determinado.	Un componente del almuerzo ejecutivo debe tener un id, nombre, tipo (entrada, principio, proteína o bebida). Ejemplos de componentes son: Sopa de verduras, sopa de carantanta, frijoles, lentejas, pollo frito, pollo sudado, limonada, jugo de tomate etc.
2	HE02	Yo como administrador del restaurante necesito registrar el almuerzo	El restaurante puede ofrecer uno o más componentes por

		ejecutivo del día en el sistema para que los clientes sepan que ofrece el restaurante como entrada, principio, proteína y bebida.	entrada, principio, proteína y bebida. Por ejemplo, un cliente puede elegir de principio: frijoles, arvejas o lentejas; de entrada: Sopa de verduras, sopa de maíz o frutas. Cuando algún componente se agote, el administrador lo puede quitar de la oferta para que los clientes no lo soliciten más. Se debe además poder seleccionar y agregar una imagen del almuerzo ejecutivo. Además, se puede agregar una ligera descripción del plato ejecutivo que motivará al cliente a realizar el pedido.
3	HE03	Yo como administrador del restaurante necesito registrar los platos a la carta que ofrece el restaurante para que los clientes los puedan conocer y pedir a través del sistema.	Cada plato a la carta debetener un nombre, una descripción y una imagen. Ejemplos de plato especiales son: pechugas rellenas con tocino y queso, bandeja paisa, italian sausage and peppers, cheesy broccoli stuffed chicken breast, etc.
4	HE04	Yo como visitante necesito entrar a la plataforma y visualizar los restaurantes que tiene la ciudad para elegir uno en particular y visualizar los platos ejecutivos y a la carta que ofrece para ese día.	Los usuarios siempre quieren visualizar todos los platos, filtrar los platos de supreferencia y mirar el contenido que ofrece el restaurante sin hacer ningún tipo de registro o inicio de sesión.
9	HE09	Yo como súper	Un restaurante tiene

		<p>administrador de la plataforma necesito registrar un nuevo restaurante para que el restaurante pueda posteriormente ofrecer sus platos, pedidos a través de la plataforma.</p>	<p>un nit, nombre, eslogan, propietario, dirección, teléfonos, ciudad, administrador y fotografía. El listado de restaurantes se debe mostrar en una tabla que permita filtrar de acuerdo a su nombre, nit o el nombre del administrador.</p>
--	--	---	---

Tabla 1 Historias épicas de importancia arquitectural

4.1. Historias épicas de importancia arquitectural

Prioridad	Código	Historia épica	Contexto
1	HE01	Yo como administrador del restaurante necesito registrar los componentes del almuerzo ejecutivo para posteriormente ofrecer a través de la plataforma el almuerzo de un día determinado.	Un componente del almuerzo ejecutivo debe tener un id, nombre, tipo (entrada, principio, proteína o bebida). Ejemplos de componentes son: Sopa de verduras, sopa de carantanta, frijoles, lentejas, pollo frito, pollo sudado, limonada, jugo de tomate etc.
2	HE02	Yo como administrador del restaurante necesito registrar el almuerzo ejecutivo del día en el sistema para que los clientes sepan que se ofrece como entrada, principio, proteína y bebida.	El restaurante puede ofrecer uno o más componentes por entrada, principio, proteína y bebida. Por ejemplo, un cliente puede elegir de principio: frijoles, arvejas o lentejas; de entrada: Sopa de verduras, sopa de maíz o frutas. Cuando algún componente se agote, el administrador lo puede quitar de la oferta para que los clientes no lo soliciten más. Se debe además poder seleccionar y agregar una imagen del almuerzo ejecutivo. Además, se puede agregar una ligera 2 descripción del plato ejecutivo que motivará al

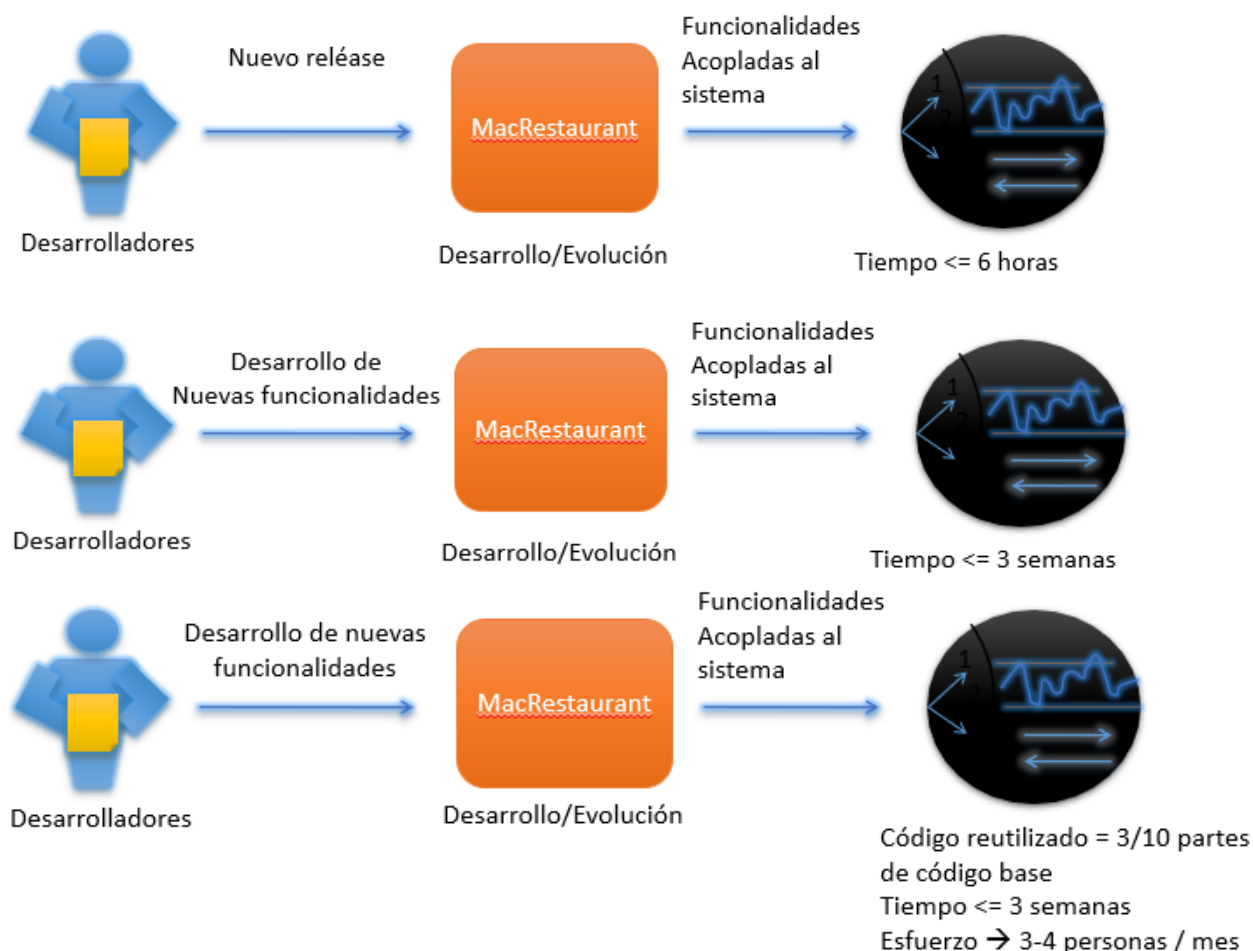
			cliente a realizar el pedido.
3	HE03	Yo como administrador del restaurante necesito registrar los platos a la carta que ofrece el restaurante para que los clientes puedan conocerlos y pedir a través del sistema.	Cada plato a la carta debe tener un nombre, una descripción y una imagen. Ejemplos de plato especiales son: pechugas rellenas con tocino y queso, bandeja paisa, italian sausage and peppers, cheesy broccoli stuffed chicken breast, etc.
9	HE09	Yo como súper – administrador de la plataforma necesito registrar un nuevo restaurante para que el restaurante pueda posteriormente ofrecer sus platos y atender pedidos a través de la plataforma.	Un restaurante tiene un nit, nombre, eslogan, propietario, dirección, teléfonos, ciudad, administrador y fotografía. El listado de restaurantes se debe mostrar en una tabla que permita filtrar de acuerdo a su nombre, nit o el nombre del administrador.

Tabla 2 Requerimientos de importancia arquitectural

4.2. Atributos de calidad de importancia arquitectural

Teniendo en cuenta los requerimientos que se presentan para el proyecto (especificados en el documento de requerimientos) se tiene la priorización de los siguientes atributos de calidad: **Usabilidad, Escalabilidad, Fiabilidad, Compatibilidad, Mantenibilidad, Portabilidad**; de los cuales, se encontró una mayor necesidad e inclinación hacia los siguientes: **Escalabilidad, Usabilidad y Mantenibilidad**. Esto permite poner atención especial en algunos atributos y posponer algunos otros, como lo son la Compatibilidad y la Portabilidad. En las siguientes secciones (específicamente en la sección “Resumen de las decisiones de arquitectura”) se aclarará qué importancia tiene esta preselección a nivel arquitectural; por ahora, se debe tener en cuenta que el proyecto tiene como prioridad los seis atributos anteriormente mencionados y, de ellos, tres son de mayor importancia.

4.3. Escenarios de calidad de importancia arquitectural



5. Resumen de las decisiones de arquitectura

Tomando como base los atributos que se han priorizado (mencionados en la sección 4.2 de este documento) y de acuerdo a los requerimientos (aquellos seleccionados como de importancia arquitectural – Mencionados en la sección 4.1 de este documento) es posible presentar las siguientes decisiones – arquitecturalmente hablando.

- Estilo arquitectónico:** Los requerimientos mostrados de importancia arquitectural indican que cada restaurante tiene un tipo de plato (hasta el momento pueden ser platos ejecutivos o platos a la carta) que con el pasar del tiempo, puede escalar (ya no serían solamente platos ejecutivos y platos a la carta, sino que puede incluirse comida rápida, por ejemplo), esto permitirá que muchos restaurantes más se hagan aliados a la plataforma. Uno de los puntos fuertes de los microservicios es, precisamente, el favorecer la escalabilidad, seguido de brindar facilidad en el mantenimiento de cada subsistema, además de permitir la reutilización de código a la hora de un nuevo reléase, favoreciendo así uno de los atributos priorizados (Mantenibilidad). Por otro lado, una de las grandes ventajas de los microservicios es que permite especializar muy bien ciertas tareas (algo que puede ayudarnos a la hora de, por ejemplo, crear un nuevo tipo de plato – es posible tener microservicios que se especialicen en la gestión de cada tipo de plato) favoreciendo otro de

los atributos de calidad priorizados anteriormente (Adecuación funcional). A pesar de todo lo mencionado, la decisión que se ha tomado es optar por una arquitectura basada en servicios, en donde se comparte una única base de datos para todos los subsistemas; en esta medida, se aprovecha lo mejor de microservicios y lo mejor de una arquitectura en monolito.

La arquitectura hexagonal es utilizada con el fin de mejorar la arquitectura de tres capas colocando en el centro de la aplicación el dominio o la lógica del negocio y haciendo que las demás capas sean exteriores y dependan de la lógica de negocio. De tal que la aplicación será fácil de modificar y nos permitirán hacer cambios en la aplicación de manera más fácil, permitiéndonos tener más cohesión, mejor acoplamiento.

- **Lenguaje de programación:** Se utilizará el lenguaje de programación Java EE, debido a que el grupo de desarrollo del proyecto cuenta con experiencia y un grado de experticia en el mismo. Además de esto, Java EE tiene varias especificaciones de API, como JDBC (brindando una alternativa para la conexión a base de datos), e-mail (necesaria para la interfaz y la interacción del repartidor). Por último, pero no menos importante, Java se ejecuta en una máquina virtual, brindando la opción de trabajarlo en múltiples plataformas.
- **Framework:** Para este desarrollo se hará uso de Spring Boot, framework partidario de SpringFramework. Este framework facilita tanto la tarea de desarrollo como la tarea de despliegue en un servidor. Spring Boot se enfoca en facilitar las tareas que no son de desarrollo, para que cada desarrollador se encargue y se enfoque únicamente en la tarea de desarrollar, ganando tiempo y esfuerzo.
- **Tipo de aplicación:** Lo adecuado para una aplicación de esta categoría, serían aplicaciones web, sin embargo, el equipo de desarrollo cuenta con falta de experiencia sobre las mismas y sobre aplicaciones Android. Por lo mismo, se toma la inclinación hacia **aplicación de escritorio**, con una posibilidad de complementarse muy bien, en un futuro y en versiones posteriores, con páginas y aplicaciones web, además de aplicaciones para smartphone también.

6. Modelo de contexto

En la figura se muestra, como punto referencial, la aplicación que se va a construir “Mac Restaurant”, la cual se presenta en relación con cinco actores principales y tres sistemas adicionales o de soporte a la aplicación. Los actores principales que se han identificado hasta el momento son: repartidor, visitante, cliente, administrador del restaurante y súper administrador. Por otro lado, los sistemas que actúan en conjunto a la aplicación principal “RlxdFood” son: GPS system, E-mail system, online payment system (por el momento no se ha tomado decisión sobre el sistema específico o el api de terceros que se utilizará). Cada actor y sistema juega un rol importante dentro de la aplicación, los cuales se ven a continuación.

2. Diagrama disponible en:

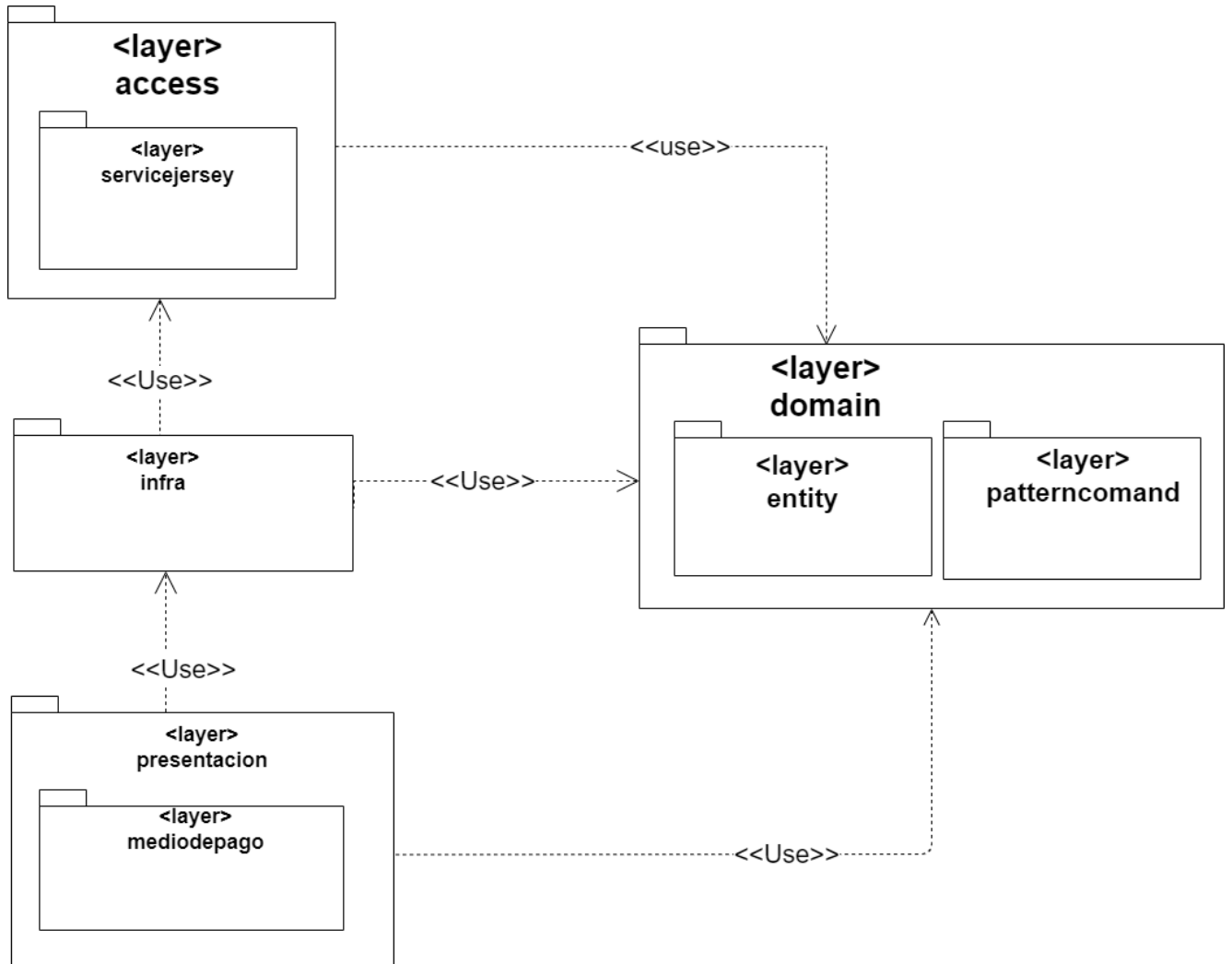
https://drive.google.com/file/d/12G9ttxV_cWGGWpPNu5ZZOt9sB2iiDNbY/view?usp=sharing

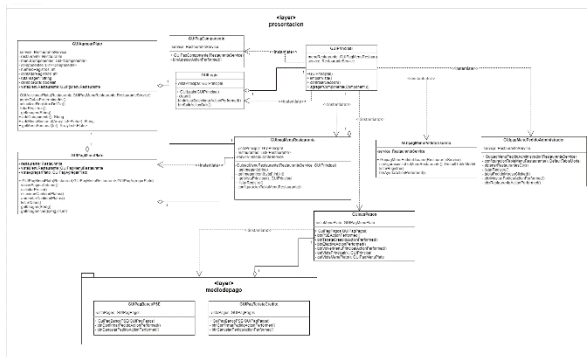
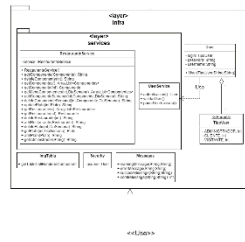
El link dispuesto para el diagrama y la figura 1 contienen la información de una versión anterior del

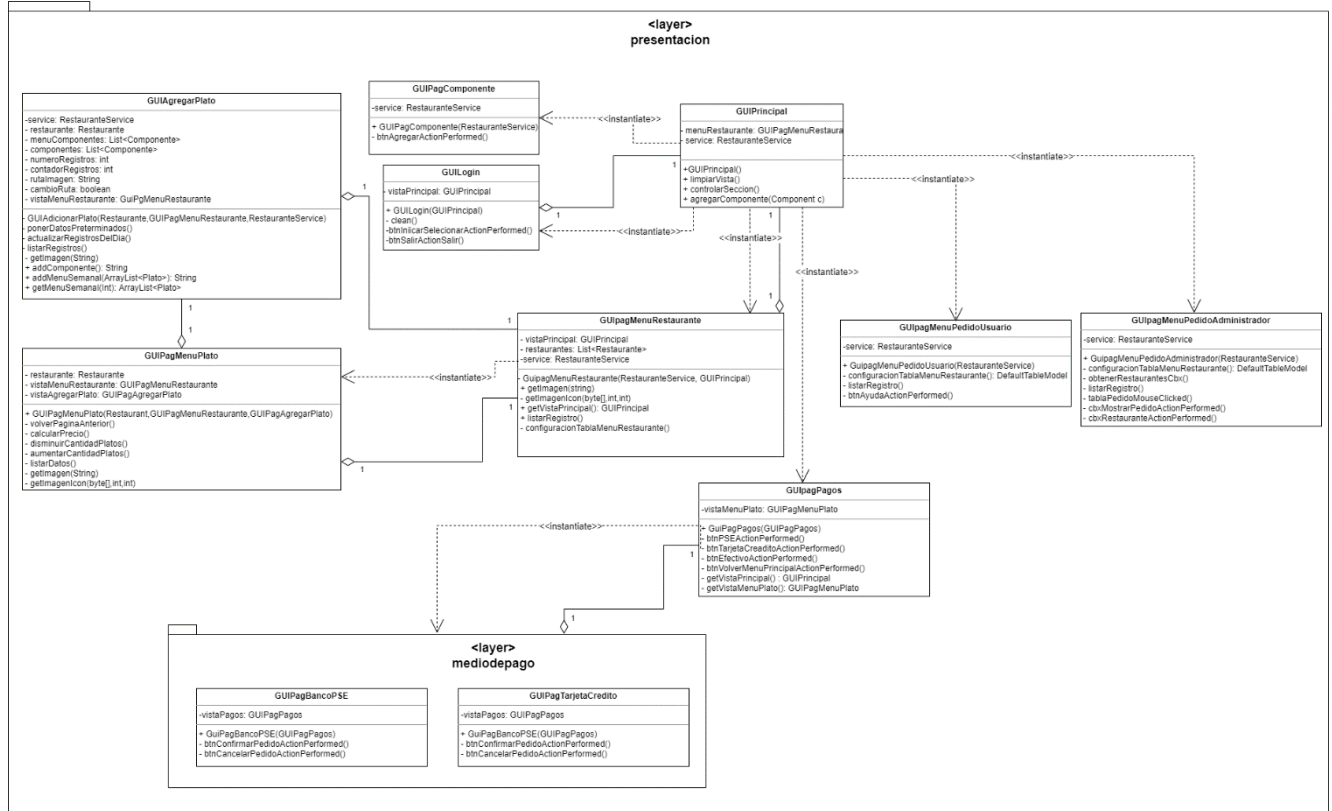
sistema. Este modelo fue modificado y ahora sólo presenta cinco actores principales (Se borró Chef-cocineros)

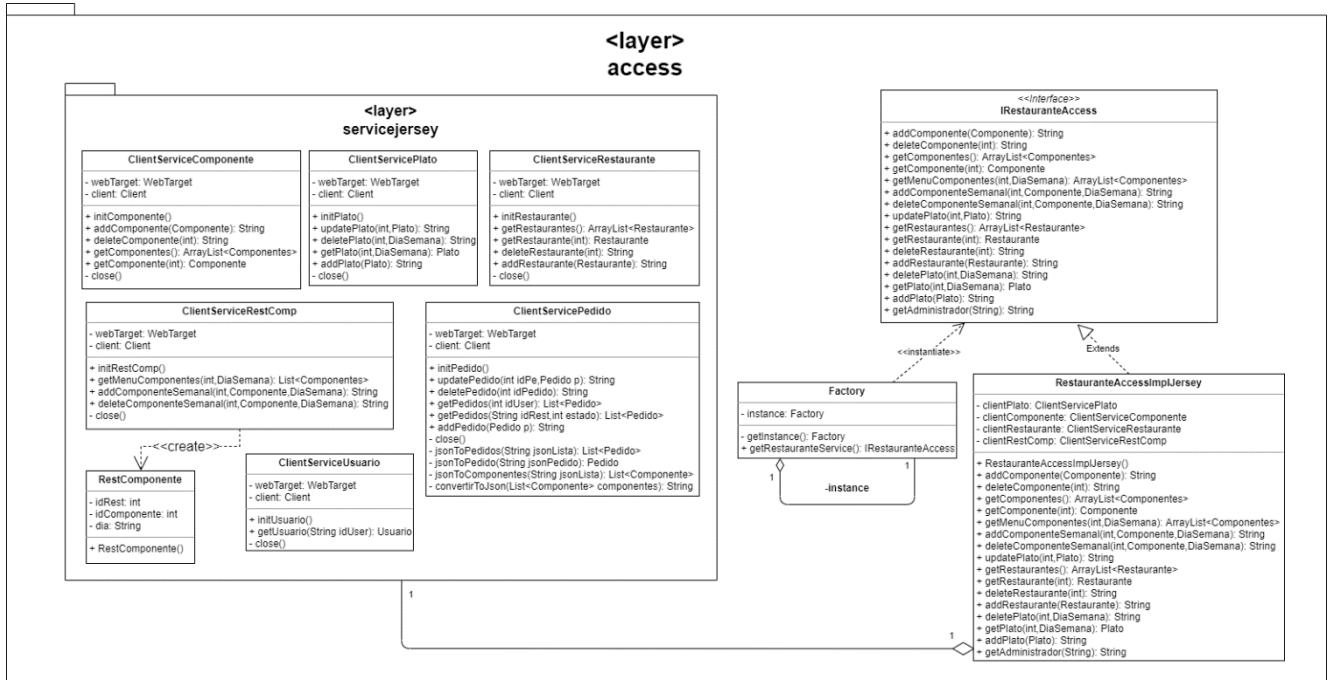
El diagrama presenta cinco actores y tres sistemas de acompañamientos, principales, los cuales se describen a continuación:

- **Repartidor:** Persona encargada de recoger el pedido solicitado y transportarlo hacia la dirección estipulada. Dentro de la plataforma, esta persona también tiene la tarea de cambiar los estados del pedido “Enviado” – “En camino” y “En camino” – “Entregado” y, además de cambiar el estado del pedido como tal, también deberá cambiar su estado “Disponible” – “Ocupado” y “Ocupado” – “Disponible” para poder ser seleccionado para realizar entregas.
- **Visitante:** Un visitante es todo aquel que ingresa a la plataforma, puede mirar los restaurantes listados, los platos tanto ejecutivos como a la carta que ofrecen los diferentes restaurantes sin ningún tipo de inicio de sesión. Si un visitante desea realizar un pedido, debe registrarse en la plataforma y pasará a ser llamado “Cliente”.
- **Cliente:** Un cliente es todo aquel que se ha registrado satisfactoriamente en la plataforma, puede realizar pedidos, observar el estado de su pedido y el tiempo estimado que tardará en ser entregado, además de las diferentes opciones de pago como lo son: tarjeta de crédito, PSE y efectivo.
- **Administrador del restaurante:** Un administrador del restaurante es quien se encarga de la gestión de los diferentes platos. La gestión puede incluir registro, actualización, eliminado de un componente de un plato ejecutivo, registro y actualización del plato ejecutivo como tal, registro de platos a la carta, entre otros.
- **Súper administrador:** Es quien se encarga de brindar soporte técnico a la aplicación (mantenimiento y solución de bugs) además, este súper administrador tiene como tarea la gestión de restaurantes, lo que incluye el registro, la actualización, eliminado, entre otros.
- **GPS System:** Sistema que brindará apoyo a la aplicación en el seguimiento del pedido en tiempo real. En adición, también proporcionará una ruta al repartidor desde el restaurante hasta la dirección estipulada para entregar los diferentes pedidos.
- **E – mail System:** El sistema de E – mail brindará apoyo a la aplicación con el envío de mensajes, notificaciones a los diferentes actores y dependiendo de las diferentes situaciones.
- **Online payment System:** El sistema de pago en línea será utilizado como medio para realizar los pagos de los diferentes pedidos que no se den en modalidad de “pago con dinero en efectivo









7.3. Vista de componentes y conectores

Se muestran, a continuación, los componentes que harán parte de una primera versión de entrega, estos componentes son los que se considera entregarán valor al cliente desde el inicio.

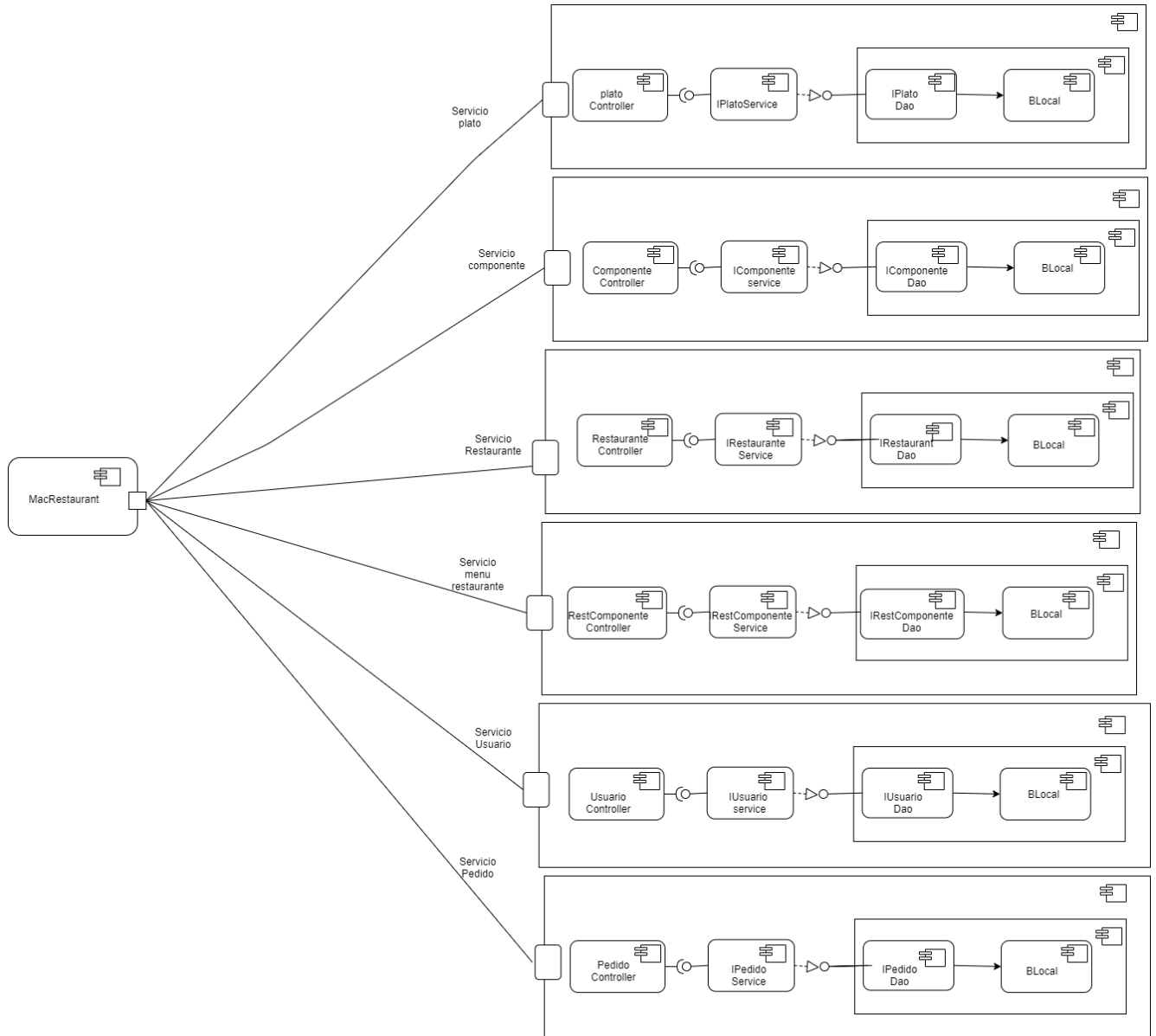
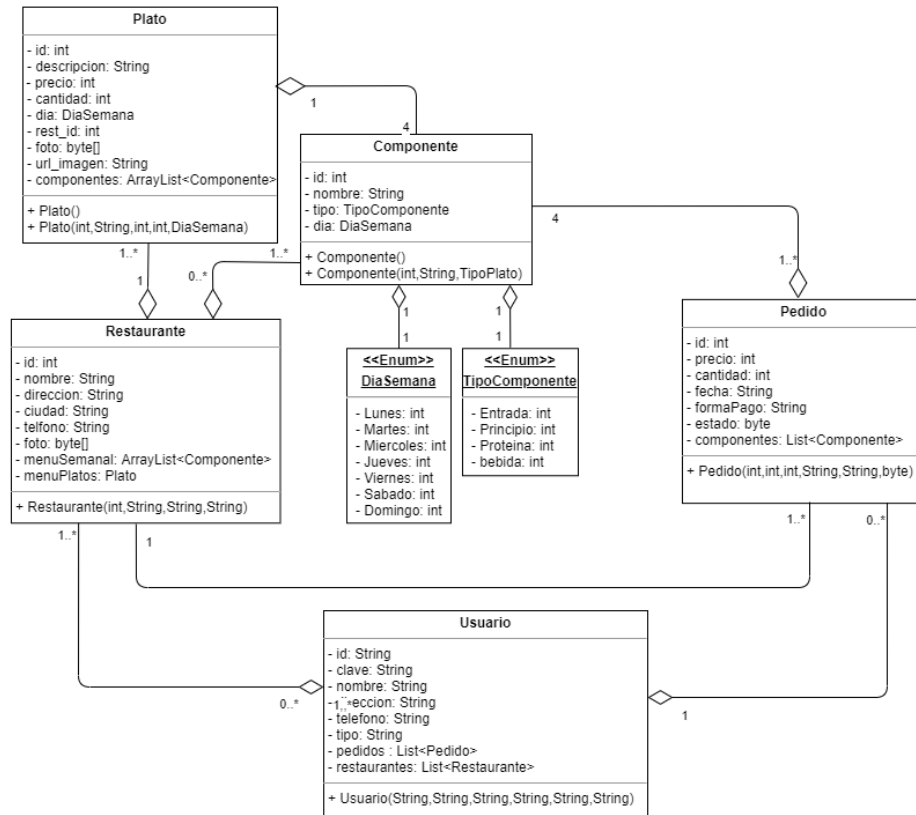


Figura 7 Diagrama de contenedores - 1era versión

8. Vista de implementación

En la figura 8 se muestra el diagrama de clases y la implementación del subsistema components – handler. Los demás subsistemas (al menos los que se tiene hasta el momento) siguen el mismo esquema, exceptuando el subsistema restaurant-handler-service, el cual utiliza un paquete adicional denominado dto.



9. Vista de despliegue

La vista de despliegue es de suma importancia, permite observar de una manera clara, la instalación que debe hacerse de cada uno de los componentes del sistema, en las diferentes máquinas necesarias.

