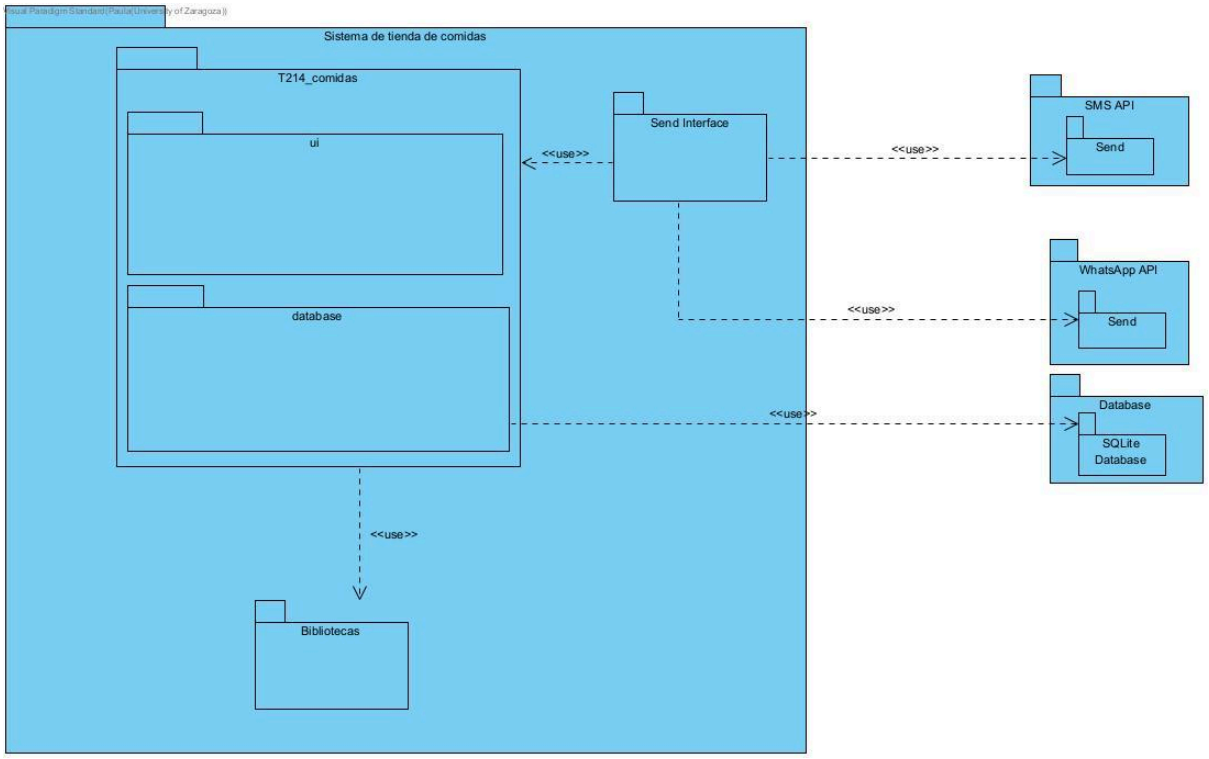
Carlota Moncasi Gosá 839841
Paula Soriano Sánchez 843710

VISTAS DE MÓDULOS

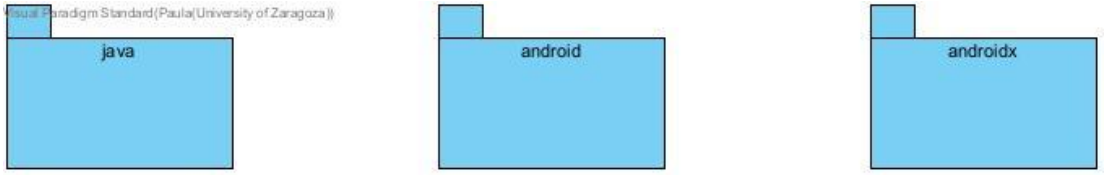
Vista primaria



Carlota Moncasi Gosá 839841
Paula Soriano Sánchez 843710

Actúa como un nivel superior de encapsulación para *T214_comidas*, *Send*, *Bibliotecas* y las Interfaces de *Send* y *Database*.

Módulo *Bibliotecas*

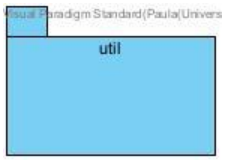


Responsabilidades:

El nivel de detalle debe ser el necesario para que el lector comprenda completamente las funciones del módulo.

Módulo *java*

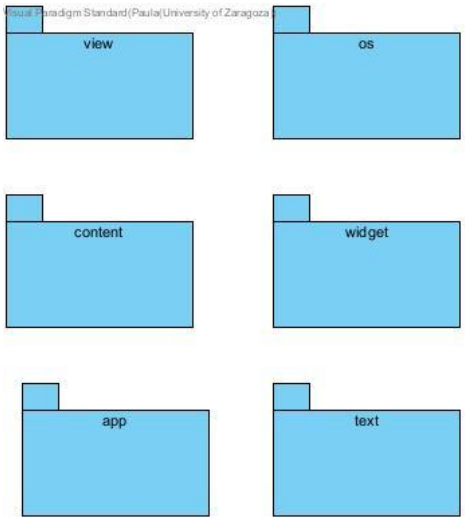
Carlota Moncasi Gosá 839841
Paula Soriano Sánchez 843710



Responsabilidades:

- Util:
  Incluye clases y utilidades comunes que son esenciales para el desarrollo de aplicaciones en Java.
  Proporciona colecciones, manipulación de fechas, estructuras de datos y otras utilidades básicas.

Módulo *android*



Responsabilidades:

- View (Vista):
  Representar la interfaz de usuario (UI) y gestionar la interacción del usuario.
  Proporcionar elementos visuales como botones, campos de texto, y contenedores para la construcción de la interfaz.

- OS (Sistema Operativo):
  Proporcionar una capa de abstracción sobre el hardware del dispositivo.
  Gestionar recursos del sistema como memoria y procesamiento.
  Facilitar la ejecución de aplicaciones y su interacción con el hardware.

- Content (Contenido):
  Gestionar y proporcionar acceso a datos y recursos de la aplicación.
  Permitir el acceso a archivos, bases de datos y otros recursos de contenido.

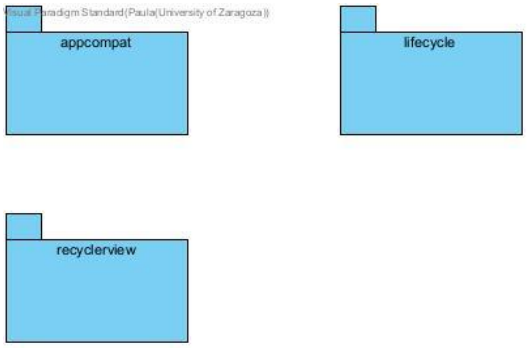
- Widget (Elemento Gráfico):
  Proporcionar componentes reutilizables para la construcción de la interfaz de usuario.

Incluir elementos gráficos interactivos o informativos.

- App (Aplicación):
  Coordinar el ciclo de vida de la aplicación, desde su inicio hasta su finalización.
  Gestionar la navegación entre actividades y fragmentos.

- Text (Texto):
  Gestionar la presentación y manipulación de texto en la interfaz de usuario.
  Proporcionar herramientas para formatear y mostrar texto de manera eficiente.

Módulo *androidx*



Responsabilidades:

- AppCompat:
  Proporcionar compatibilidad con versiones anteriores de Android para funciones modernas de interfaz de usuario.
  Incluir estilos y temas consistentes para una apariencia uniforme en diferentes versiones de Android.

- Lifecycle (Ciclo de Vida):
  Gestionar el ciclo de vida de componentes de Android, como actividades y fragmentos.
  Facilitar la ejecución de código en momentos específicos del ciclo de vida de una aplicación.

- RecyclerView:
  Ofrecer un componente de interfaz de usuario eficiente para mostrar listas y conjuntos de datos.
  Proporcionar mecanismos para la gestión eficiente de grandes conjuntos de datos en una interfaz de desplazamiento.

Paquete *es.unizar.eina*

Carlota Moncasi Gosá 839841
Paula Soriano Sánchez 843710

Responsabilidades:

- Actúa como un contenedor superior que agrupa los módulos *T214_comidas* y *Send*.
- Coordina la interacción entre estos módulos, facilitando una integración eficiente.

Interfaces Internas:

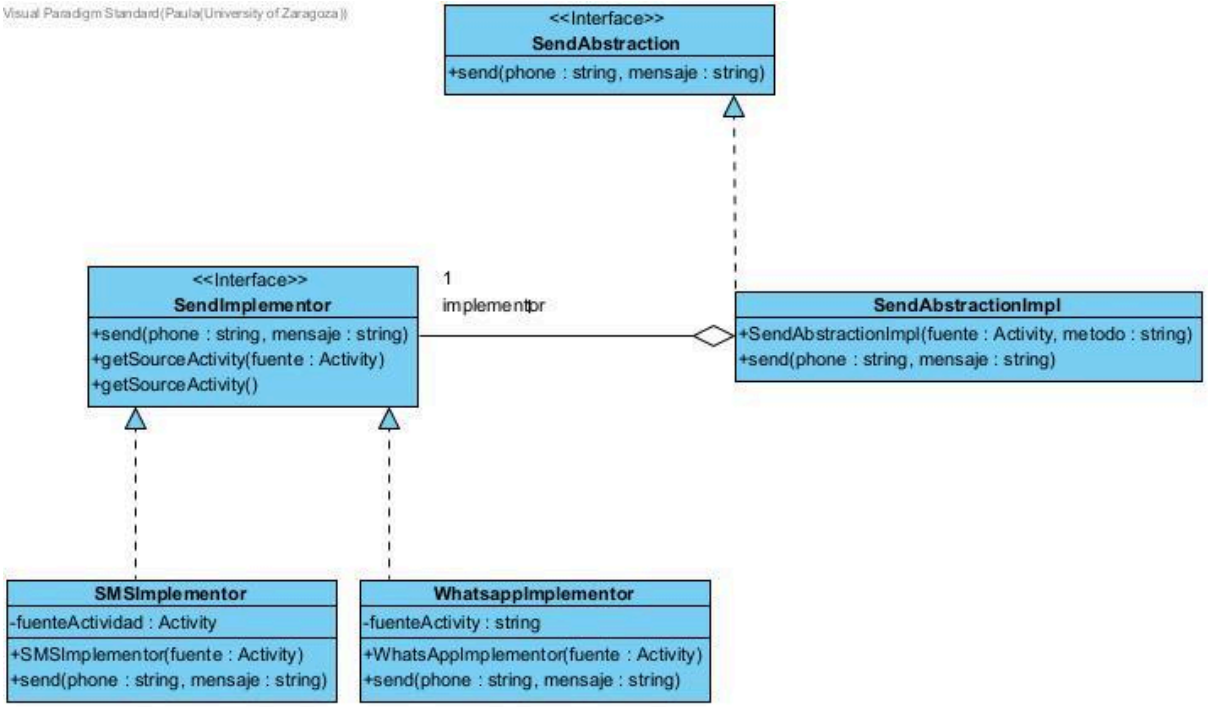
Puede contener interfaces internas utilizadas para la comunicación específica entre *T214_comidas* y *Send*.

Relación con el padre (encapsulación): el módulo es.unizar.eina actúa como un nivel superior de encapsulación para *T214_comidas* y *Send*.

Información de Implementación:

Lista de archivos: incluye los paquetes *ui* y *database*, los cuales cuentan con sus respectivas listas de archivos.

Paquete *es.unizar.eina.send*



Responsabilidades:

- Proporcionar una interfaz unificada para enviar mensajes entre diferentes componentes del sistema.
- Abstraer la implementación específica de envío (SMS, WhatsApp) para permitir flexibilidad.

Interfaces:

Encapsulación: el módulo send encapsula la lógica de envío de mensajes y proporciona una interfaz clara para la comunicación.

Información de Implementación:

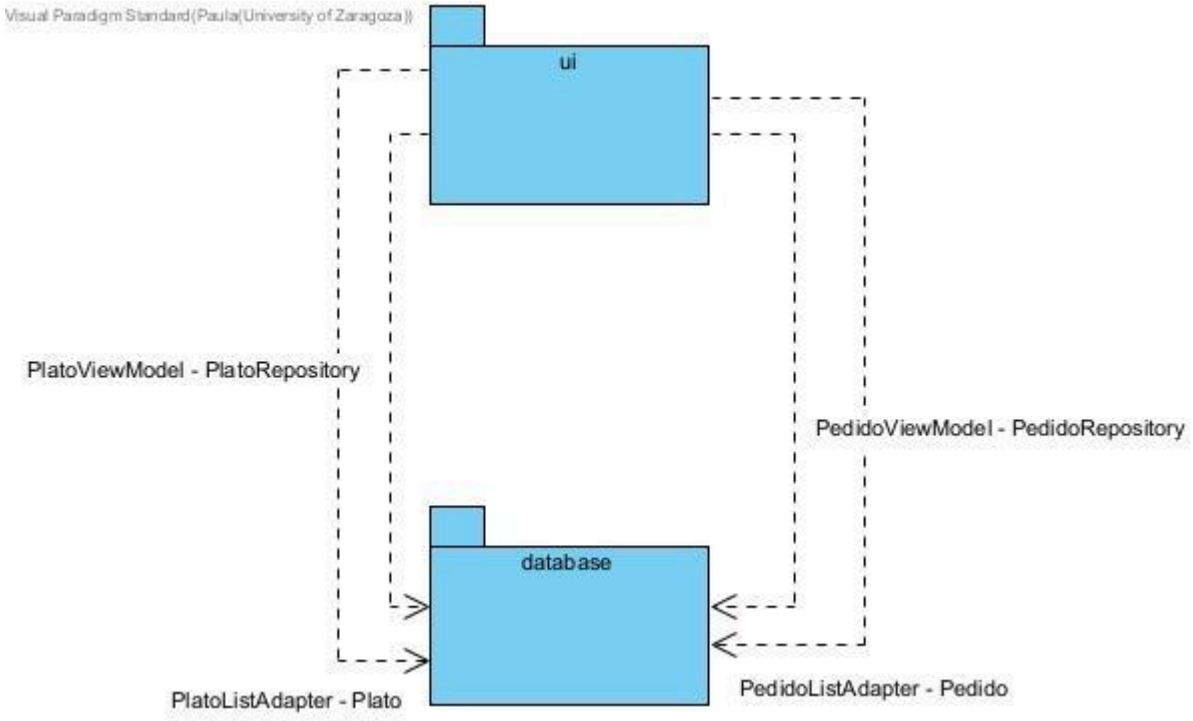
- Lista de archivos: *sendAbstraction.java, sendAbstractionImpl.java SendImplementor.java, WhatsAppImplementor.java, SMSImplementor.java*

Información para probar el módulo:

Datos de gestión:

- Fecha de finalización: 3/12/2023.

Paquete *es.unizar.eina.T214_comidas*



Responsabilidades:

- Orquestar la interacción y coordinación entre los módulos UI y Database.
- Proporcionar interfaces y servicios que permitan a los módulos interactuar de manera eficiente.

Interfaces:

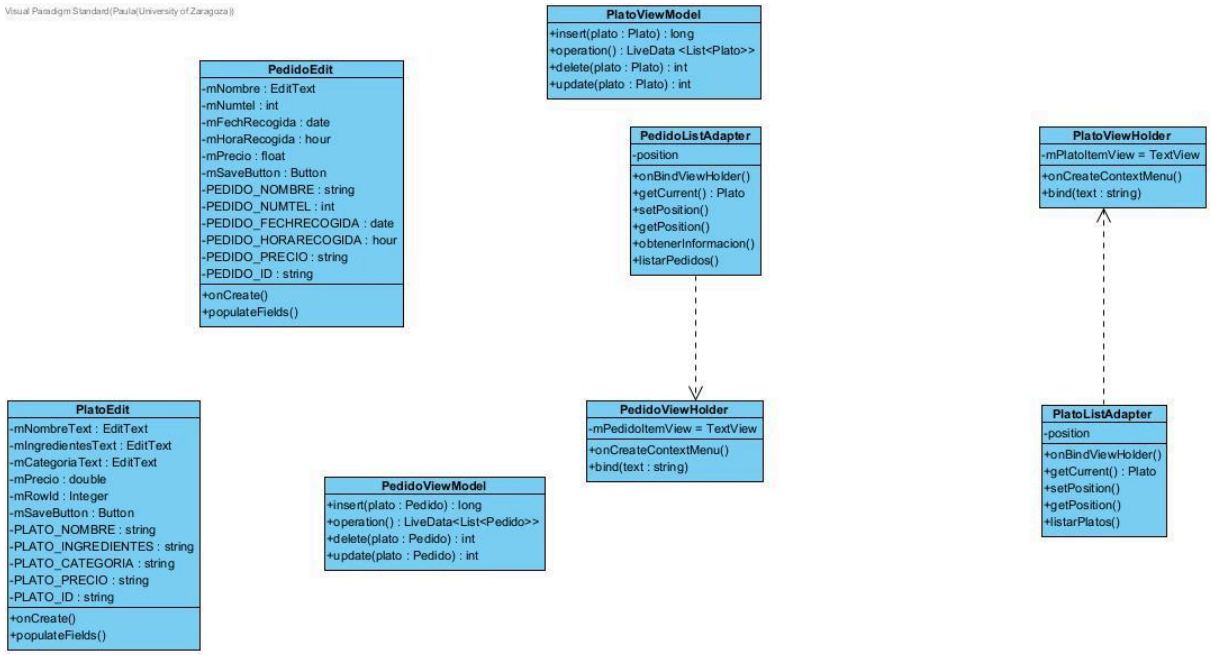
Relación con el padre (encapsulación): el módulo *T214_comidas* actúa como un contenedor que encapsula la funcionalidad de los módulos *ui* y *database*. Proporciona una interfaz de alto nivel para el sistema en su conjunto, ocultando detalles internos.

Información de Implementación:

Lista de archivos: incluye los paquetes *ui* y *database*, los cuales cuentan con sus respectivas listas de archivos.

Paquete *es.unizar.eina.T214_comidas.ui*

Carlota Moncasi Gosá 839841
Paula Soriano Sánchez 843710



Responsabilidades:

- ❖ Proporcionar interfaces y servicios que permitan a los usuarios interactuar de manera eficiente para poder trabajar con las funcionalidades del sistema.

Interfaces:

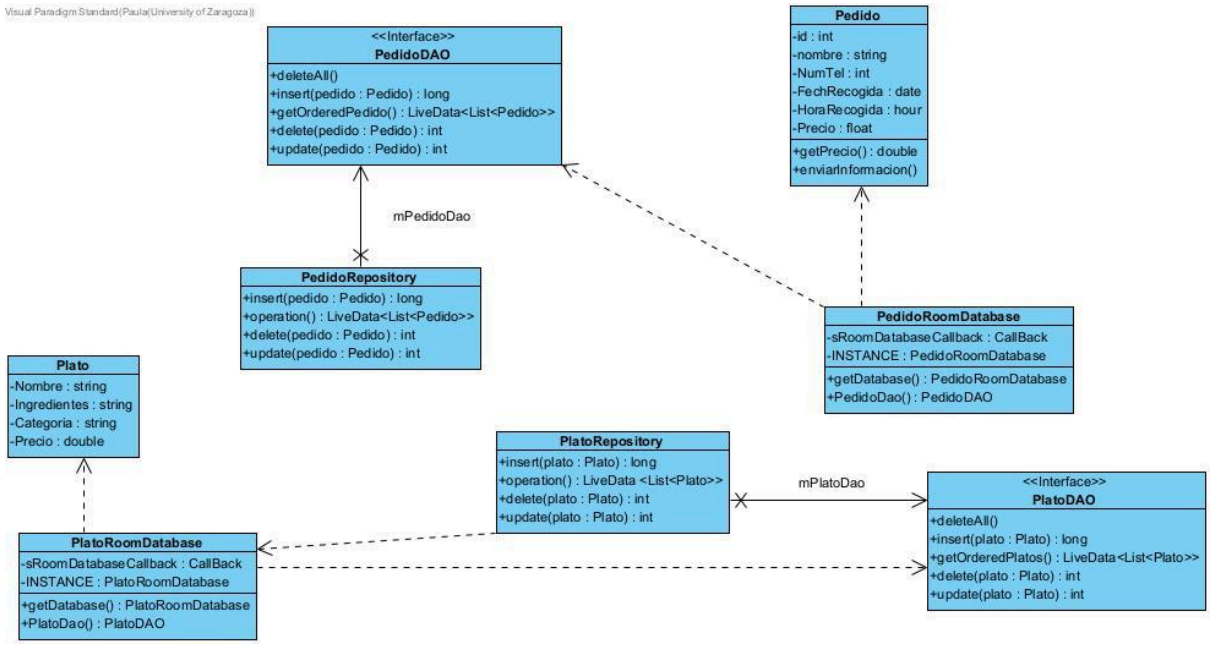
Relación con el padre (encapsulación): el módulo *T214_comidas* actúa como un contenedor que encapsula la funcionalidad del módulo *ui*.

Información de Implementación:

Lista de archivos: *PlatoEdit.java, PedidoEdit.java, PedidoListAsapter, PlatoListAdapter, PlatoViewHolder, PedidoViewHolder, PlatoViewModel, PedidoViewModel*.

Módulo *es.unizar.eina.T214_comidas.database*

Carlota Moncasi Gosá 839841
Paula Soriano Sánchez 843710



Responsabilidades:

- ❖ Almacenar y recuperar información sobre platos y pedidos de manera eficiente.
- ❖ Proporcionar una interfaz de alto nivel (a través de repositorios) para acceder a los datos desde otros módulos.
- ❖ Gestionar la comunicación con la base de datos local (Room Database).

Interfaces:

Encapsulación: El módulo database encapsula la lógica de acceso y modificación de datos, proporcionando una interfaz clara a través de repositorios.

Información de Implementación:

Lista de archivos: *Plato.java, PlatoRoomDatabase.java, PlatoRepository.java, PedidoRepository.java, PedidoDAO.java, Pedido.java, PedidoRoomDatabase.java*

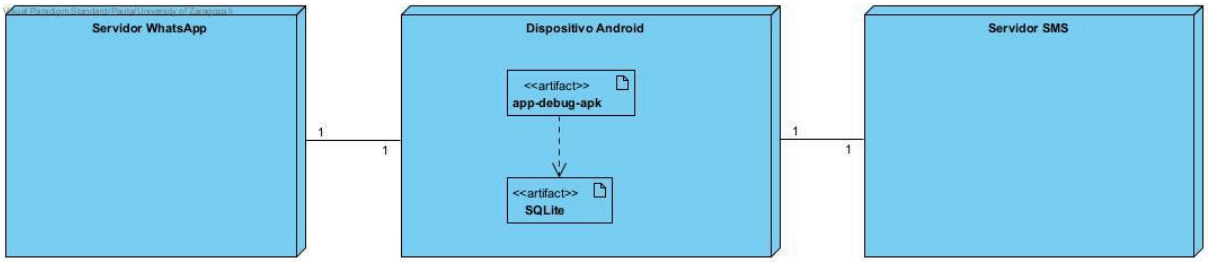
Datos de gestión:

Fecha de finalización: 14/12/2023.

Vista de distribución

- Vista de despliegue

Carlota Moncasi Gosá 839841
Paula Soriano Sánchez 843710



El propósito de la vista de despliegue es proporcionar una representación visual y detallada de cómo los distintos componentes del sistema se despliegan y se relacionan entre sí en el entorno de ejecución.

Los nodos considerados son los siguientes: *Servidor Whatsapp*, *Dispositivo Android* y *Servidor SMS.*

El nodo *Servidor Whatsapp* representa un servidor que se encarga de gestionar y procesar mensajes de Whatsapp en la aplicación.
Componentes ejecutados en el *Servidor WhatsApp*:

- WhatsApp Server Application: la aplicación principal del servidor que ejecuta lógica de negocio, gestión de usuarios y procesamiento de mensajes.

Componentes ejecutados en el *Dispositivo Android*:

- Aplicación Android: representa la aplicación específica que se ejecuta en el dispositivo Android.
- Sistema Operativo Android: el núcleo del sistema operativo Android que gestiona los recursos y proporciona servicios fundamentales.

Artefactos de *Dispositivo Android*:

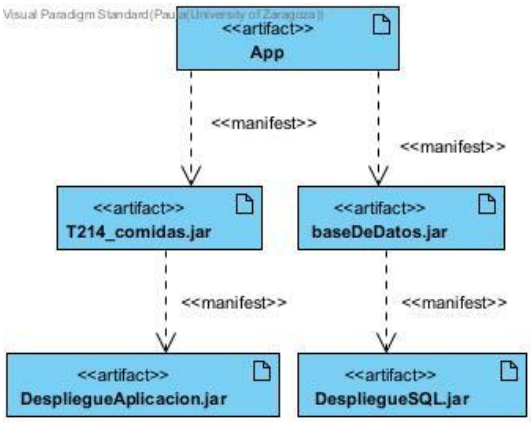
- APK (Android Package): el paquete de la aplicación que contiene el código ejecutable, recursos y *manifest*. Presenta una relación de dependencia hacia la base de datos SQLite.
- SQLite: base de datos que se utiliza como sistema de gestión de bases de datos relacionales incorporado en la aplicación Android.

El nodo *Servidor SMS* representa un servidor que se encarga de gestionar y procesar mensajes de texto (SMS) en la aplicación.
Componentes ejecutados en el *Servidor SMS*:

- Controlador de SMS: un componente que gestiona la recepción y envío de mensajes SMS.

- Vista de instalación

Carlota Moncasi Gosá 839841
Paula Soriano Sánchez 843710



Artefacto: *App*
Artefacto principal de la aplicación.

Artefacto: *T214_comidas.jar*
Contiene la lógica principal de la aplicación relacionada con las comidas.

Artefacto: *baseDeDatos.jar*
Contiene la lógica y recursos relacionados con la base de datos.

Artefacto: *DespliegueAplicacion.jar*
Facilita el despliegue de la aplicación.

Artefacto: *DespliegueSQL.jar*
Facilita el despliegue y configuración de la base de datos.

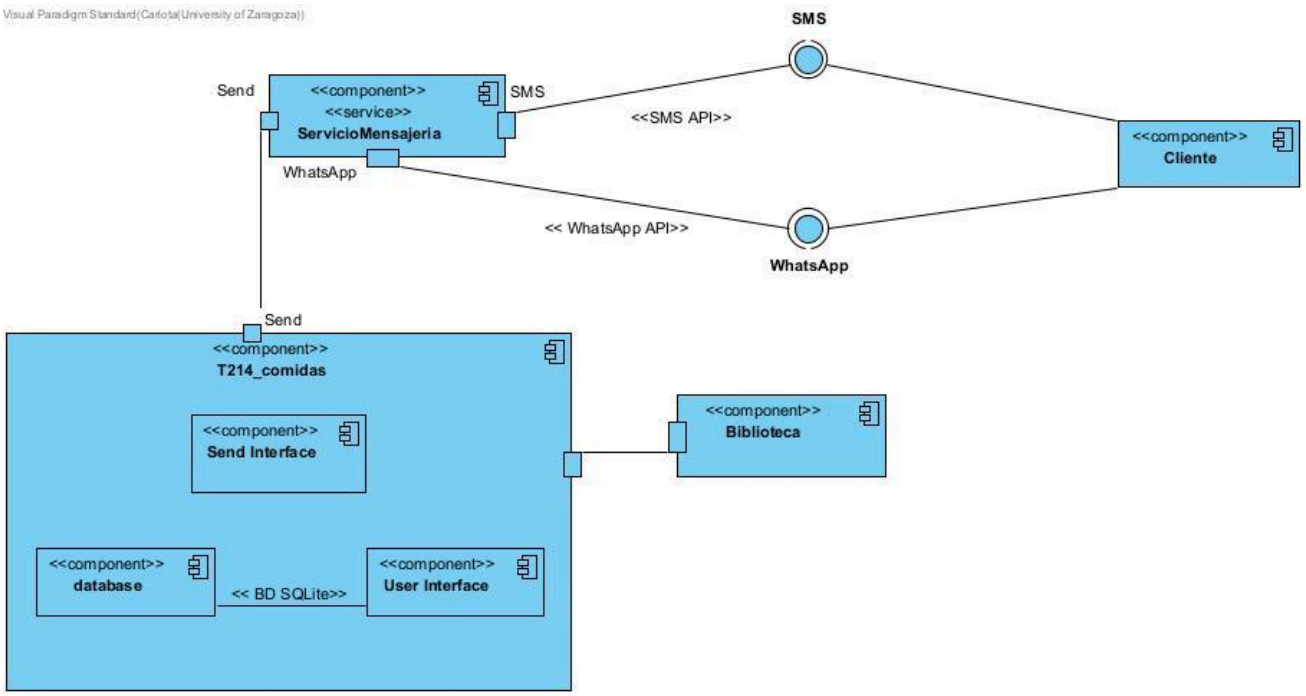
La instalación debe realizarse siguiendo el orden de dependencias.
Se espera que App sea el artefacto principal instalado por el usuario final.
T214_comidas.jar y *baseDeDatos.jar* deben ser instalados para garantizar el funcionamiento adecuado de la aplicación.
DespliegueAplicacion.jar y *DespliegueSQL.jar* son utilidades para facilitar el despliegue y configuración respectivamente.

- Vista de asignación de trabajo

Módulos	**Unidad de organización**
database	Paula Soriano y Aroa Redondo
ui	Paula Soriano y Aroa Redondo
send	Paula Soriano y Aroa Redondo

Carlota Moncasi Gosá 839841
Paula Soriano Sánchez 843710

Vista de Componente y Conector



La vista de Componente y Conector tiene como objetivo representar las conexiones entre elementos hardware y/o elementos software, con presencia en tiempo de ejecución. Resulta una herramienta muy útil para comprender la estructura del sistema y las interacciones entre sus partes.

Está compuesta por Componentes, que son unidades de procesamiento y almacenes de datos, y por Conectores, caminos de interacción entre los componentes. Los componentes presentan puertos para comunicarse entre ellos, mientras que los conectores poseen roles, que indican cómo los componentes pueden usarlos en sus interacciones.

Existen siete componentes: *T214_comidas* que engloba a otros tres componentes (*Send Interface, database y User Interface*) y está asociado por puertos al componente *Biblioteca*, el componente *ServicioMensajería* que tiene un conector al componente de *T214_comidas*, cuyo rol es el de enviar (*Send*) y por último, el componente *Cliente* que está conectado a *ServicioMensajería* mediante el conector *SMS API*, que proporciona la *interfaz SMS* y por otra parte, mediante el conector *WhatsApp API*, que proporciona la *interfaz WhatsApp*. Para su correcto funcionamiento, en el componente *ServicioMensajería* se distinguen dos puertos más: el de *WhatsApp* que se conecta con la funcionalidad de WhatsApp y el de *SMS* análogo a él.

Cada componente tiene su nombre y tipo:

- T214_comidas
- Send Interface
- Database
- User Interface (ui)
- Biblioteca

- ServicioMensajeria, de tipo servicio
- Cliente

Cada conector tiene su nombre y tipo:

- El que conecta database y User Interface, de tipo BD SQLite
- El que conecta ServicioMensajeria con la interfaz de WhatsApp, de tipo WhatsApp API
- El que conecta ServicioMensajeria con la interfaz de SMS, de tipo SMS API
- El que conecta ServicioMensajeria con T214_comidas, en concreto con Send Interface, con rol de Send (enviar)

Cabe destacar que el componente ServicioMensajeria tiene varios puertos (tres) y eso significa que puede atender concurrentemente a 3 funciones diferentes: WhatsApp, SMS y Send general.