

# Arquitecturas de Software para Aplicaciones Empresariales

## Introducción al Modelo C4

### PROGRAMA DE INGENIERIA DE SISTEMAS

Ing. Daniel Eduardo Paz Perafán ([danielp@Unicauca.edu.co](mailto:danielp@Unicauca.edu.co))

Ing. Pablo A. Magé ([pmage@Unicauca.edu.co](mailto:pmage@Unicauca.edu.co))



1. Recordatorio definición de arquitectura y etapa de diseño
2. ¿Cómo comunicar la arquitectura?
3. Modelo C4
4. Niveles del modelo C4



La Arquitectura es un nivel de diseño que hace foco en aspectos "más allá de los algoritmos y estructuras de datos de la computación".

La Arquitectura de Software se refiere a una descripción de cómo está organizado un sistema en subsistemas, módulos y componentes, y como se relacionan entre ellos..

Los elementos pueden ser :

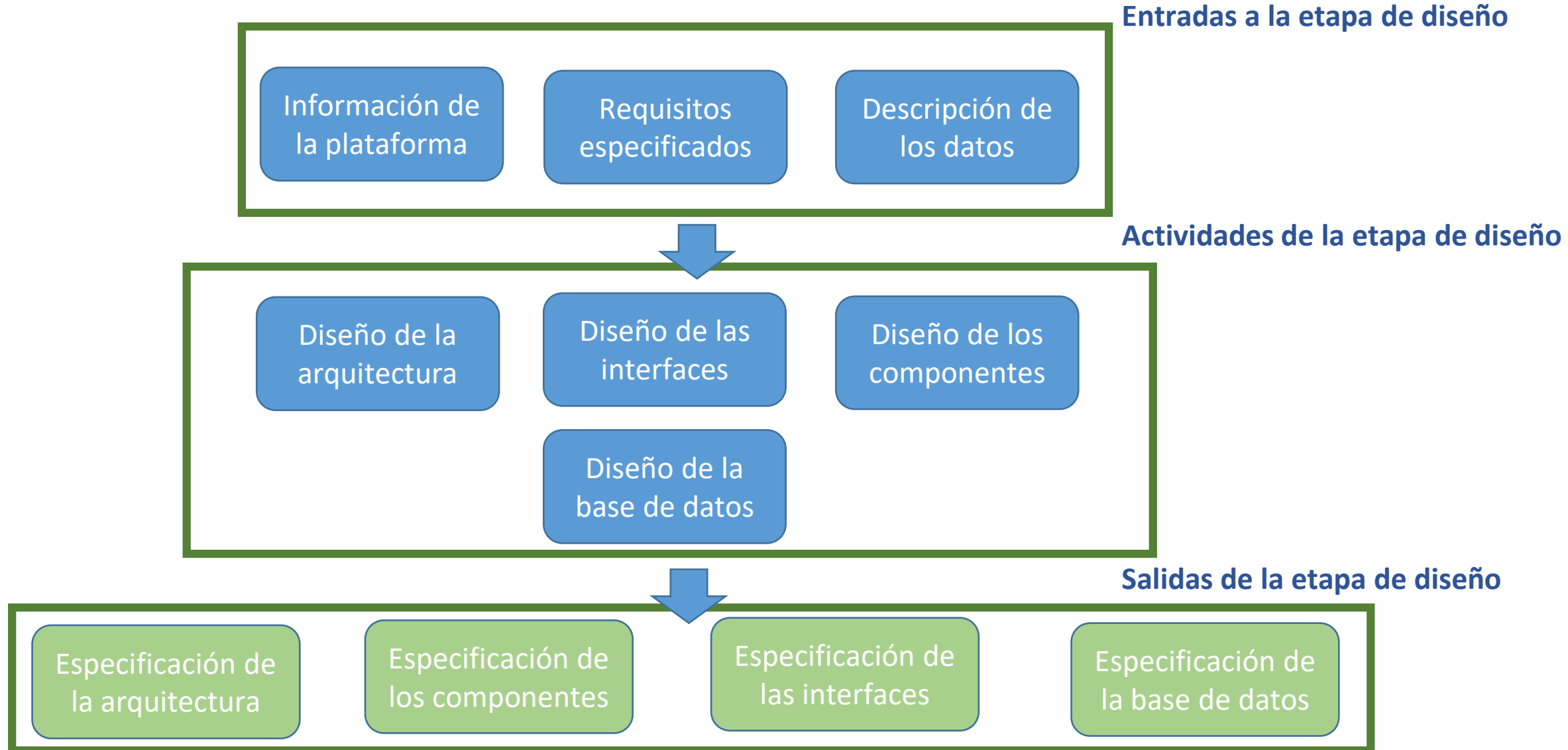
- Entidades lógicas que existen en tiempo de desarrollo (clases, paquetes, módulos)
- Entidades físicas (nodos, directorios).
- Entidades que existen en tiempo de ejecución (objetos, hilos)

La arquitectura debe representarse desde diferentes vistas y cada vista debe estar plasmada mediante un modelo.



# Etapa de diseño del software

UNIVERSIDAD DEL CAUCA – FIET  
DEPARTAMENTO DE SISTEMAS



1. Definición de arquitectura y etapa de diseño
2. ¿Cómo comunicar la arquitectura?
3. Modelo C4
4. Niveles del modelo C4



¿Existe una manera fácil de comunicar la arquitectura de un sistema de software?

- En la industria existen varios lenguajes para modelar la arquitectura (UML), SysML y ArchiMate

¿Podría una propuesta de modelado resaltar los requisitos y aún ser breve?

- Las empresas prefieren utilizar notaciones mucho más simples creadas por ellas.

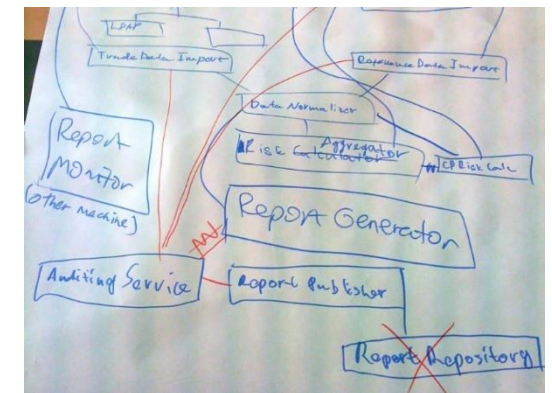
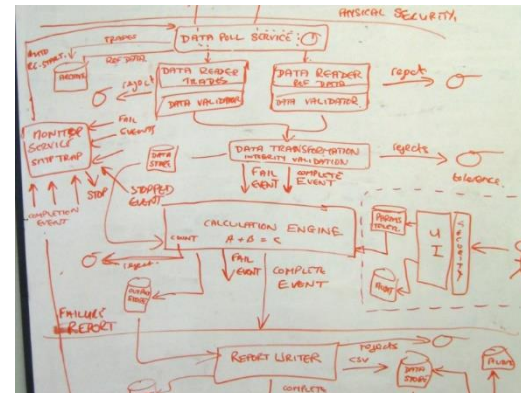
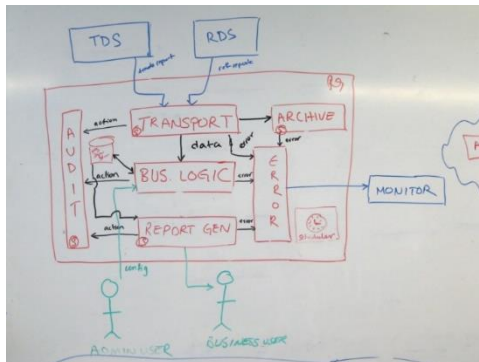
*El Manifiesto Ágil prescribe que los equipos deben **valorar el software funcional sobre la documentación completa**. Se debe crear documentación que aporte valor y que al mismo tiempo no obstaculice el progreso del equipo.*

Podemos lograr esto usando el modelo de arquitectura C4. Es un modelo estático, que proporciona una manera fácil de comunicar el diseño del sistema a todos los involucrados

# ¿Cómo comunicar la arquitectura?

Muchas empresas al comunicar la arquitectura software mediante diagramas probablemente se obtiene un desorden confuso de:

- Cuadros y líneas
- Notación inconsistente (codificación de colores, formas, estilos de línea, etc.).
- Nombres ambiguos .
- Relaciones sin etiquetar
- Terminología genérica



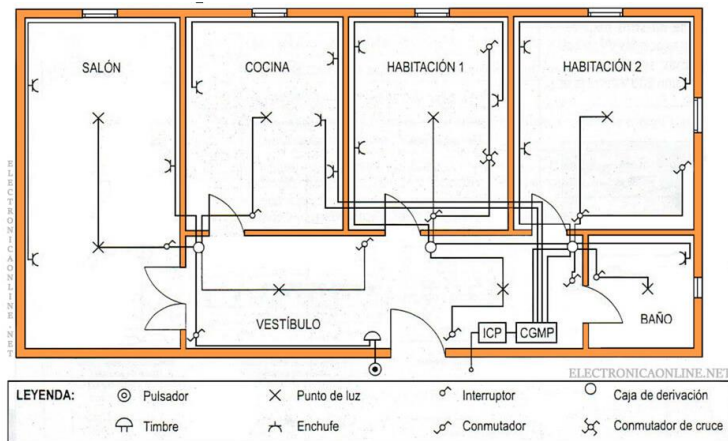
Debemos comunicar la arquitectura de manera consistente a partir de una notación.

# ¿Cómo comunicar la arquitectura?

UNIVERSIDAD DEL CAUCA – FIET  
DEPARTAMENTO DE SISTEMAS

La notación a utilizar debe permitir comunicar la arquitectura desde diferentes vistas.

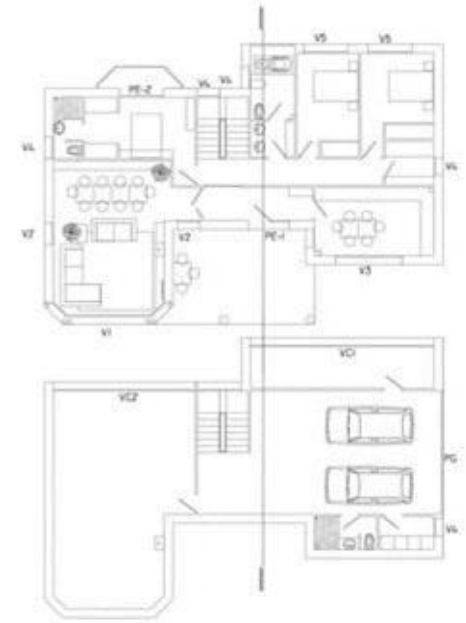
Si solicitamos a un ingeniero civil que nos comunique la arquitectura de un edificio nos mostrará diferentes planos.



Planos del sistema eléctrico



Planos del corte transversal



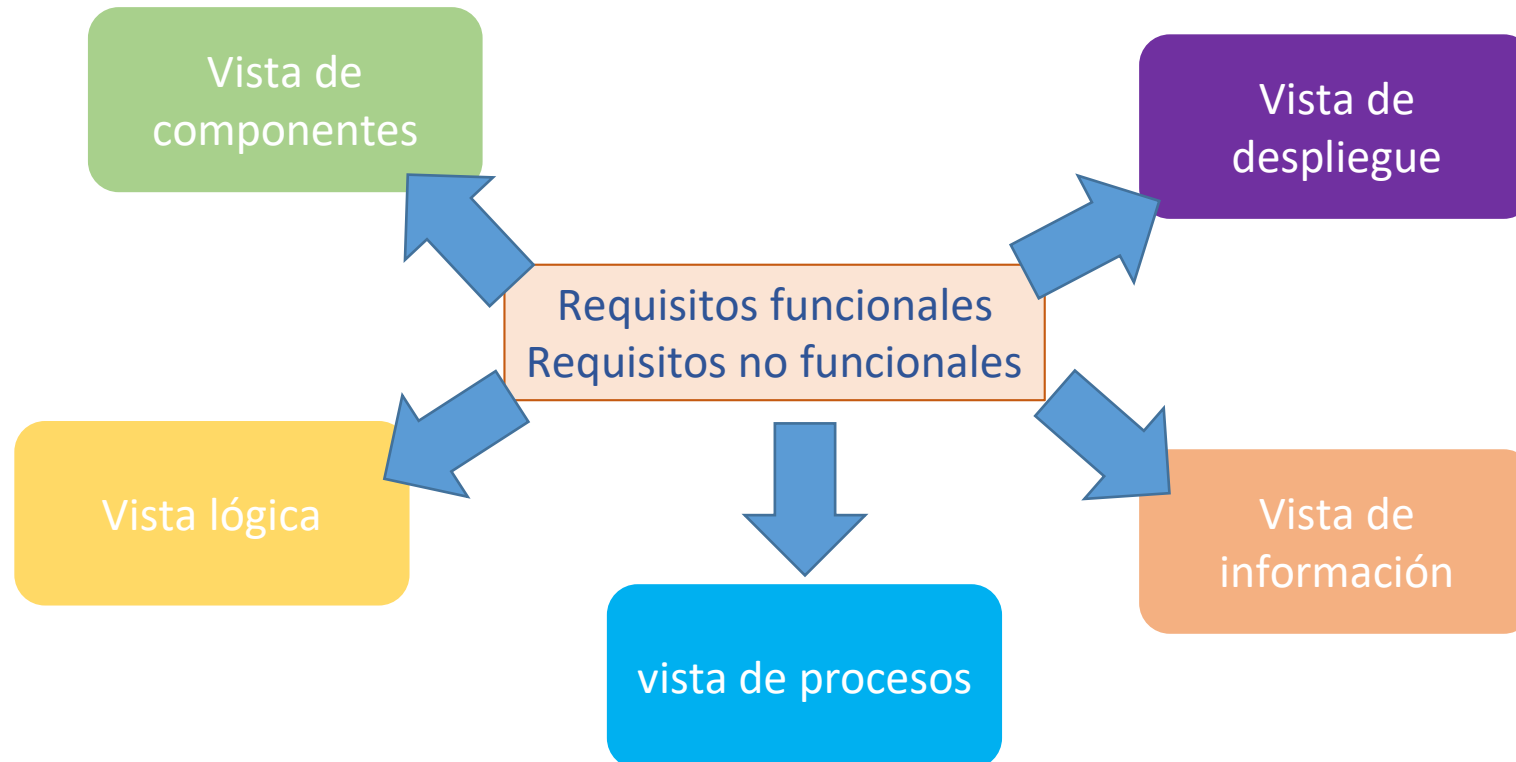
Planos del diseño de la casa



# ¿Cómo comunicar la arquitectura?

Es imposible representar toda la información relevante sobre la arquitectura de un sistema en un solo modelo arquitectónico, ya que cada modelo solo muestra una vista o perspectiva del sistema

Normalmente es necesario presentar varias vistas de la arquitectura del software.



# ¿Cómo comunicar la arquitectura?

UNIVERSIDAD DEL CAUCA – FIET  
DEPARTAMENTO DE SISTEMAS

**Vista de componentes.** Muestra cómo se descompone un sistema en módulos y componentes y cómo interactúan entre sí. La vista es útil para administradores y programadores de software.

**Vista lógica.** Muestra las abstracciones clave en el sistema como clases, procedimientos, funciones, eventos etc.

**Vista de despliegue.** Muestra como se distribuyen los componentes software en diferentes computadores en la red. Esta vista es útil para los ingenieros de sistemas que planean la implementación de un sistema.

**Vista de información.** Muestra la estructura del medio de persistencia. Por ejemplo el modelo relacional en una base de datos.

**vista de procesos.** Muestra en tiempo de ejecución, como los componentes interactúan entre sí. Pueden utilizarse diagramas de secuencia

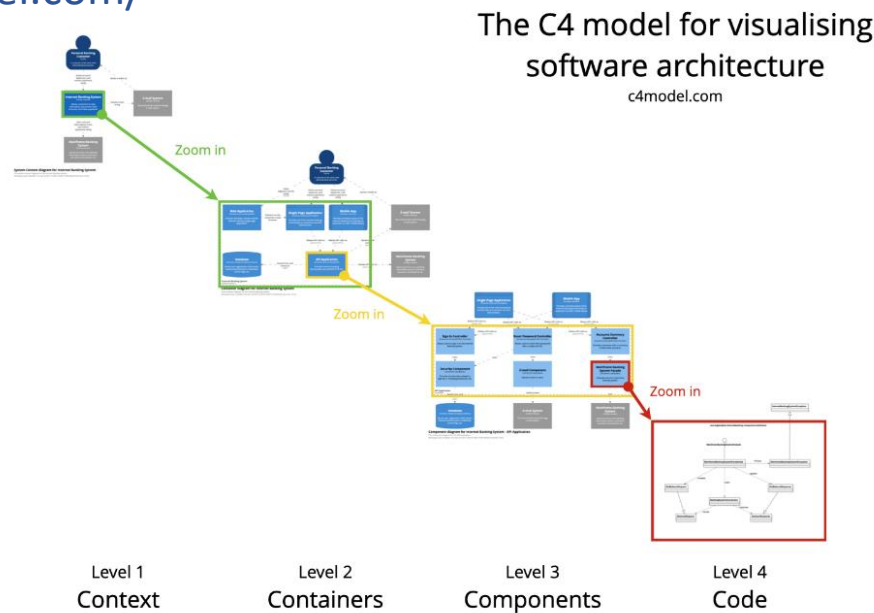


1. Definición de arquitectura y etapa de diseño
2. ¿Cómo comunicar la arquitectura?
3. Modelo C4
4. Niveles del modelo C4



El modelo C4 es solo una forma de describir un sistema de software, desde diferentes niveles de abstracción, y no implica nada sobre el proceso de entrega de software.

<https://c4model.com/>



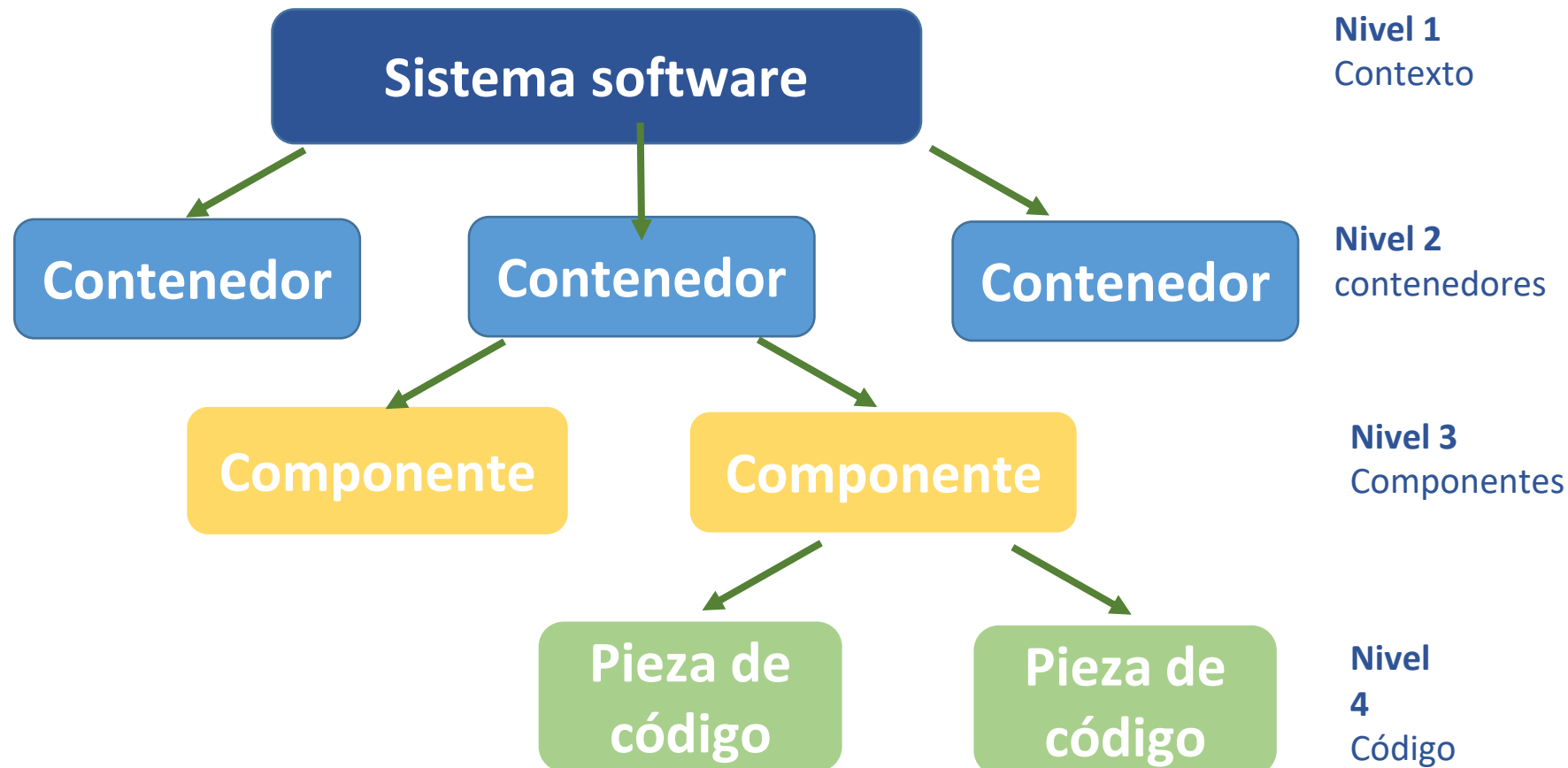
Simon Brown



El modelo C4 se creó como una forma de ayudar a los equipos de desarrollo de software a **describir y comunicar la arquitectura de software**, tanto durante las sesiones de diseño iniciales como al documentar retrospectivamente una base de código existente

# Niveles del modelo C4

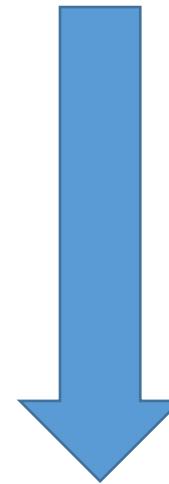
Un **sistema de software** se compone de uno o más **contenedores** (aplicaciones web, móviles, de escritorio, bases de datos), cada uno de los cuales contiene uno o más **componentes** (Controladores, Repositorios), que a su vez son implementados por uno o más **elementos de código** (por ejemplo, clases, interfaces).



El modelo C4 plantea 4 diagramas basicos para describir la arquitectura de un sistema software.

- Diagrama de Contexto.
- Diagrama de contenedores.
- Diagrama de componentes
- Diagrama de Código
- **Diagramas complementarios (despliegue, relacional)**

General



Particular

De aquí es de donde el modelo C4 recibe su nombre. Para construir los diagramas se utiliza una notación.

- ❖ El modelo C4 prioriza la abstracción y es independiente de la notación.
- ❖ Hay que asegurarse de que la notación de los diagramas tenga sentido y que los diagramas sean comprensibles.
- ❖ Los siguientes diagramas de ejemplo se crean utilizando una notación de "recuadros y líneas"

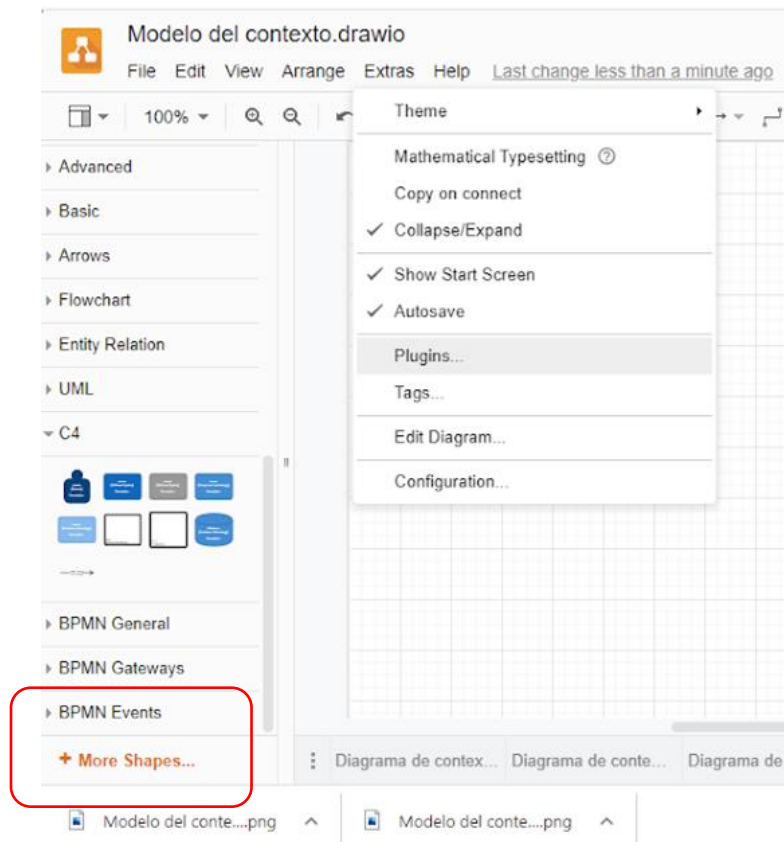


- ❖ Es posible crear los diagramas utilizando UML con el uso apropiado de paquetes, componentes y estereotipos.

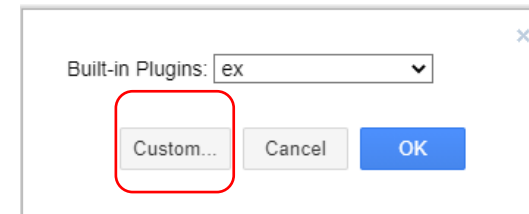
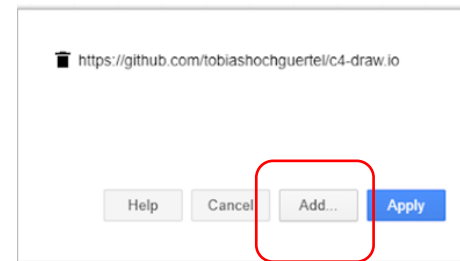
# Aplicación para crear los diagramas

UNIVERSIDAD DEL CAUCA – FIET  
DEPARTAMENTO DE SISTEMAS

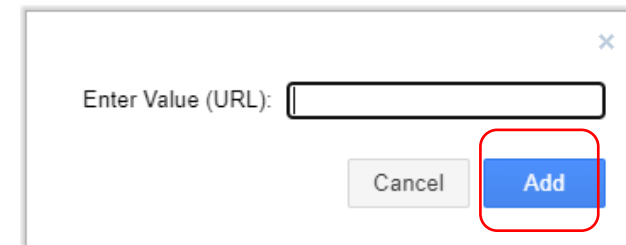
## Aplicación para realizar el modelado Plugin



<https://app.diagrams.net/>



<https://github.com/tobiashochguertel/c4-draw.io>

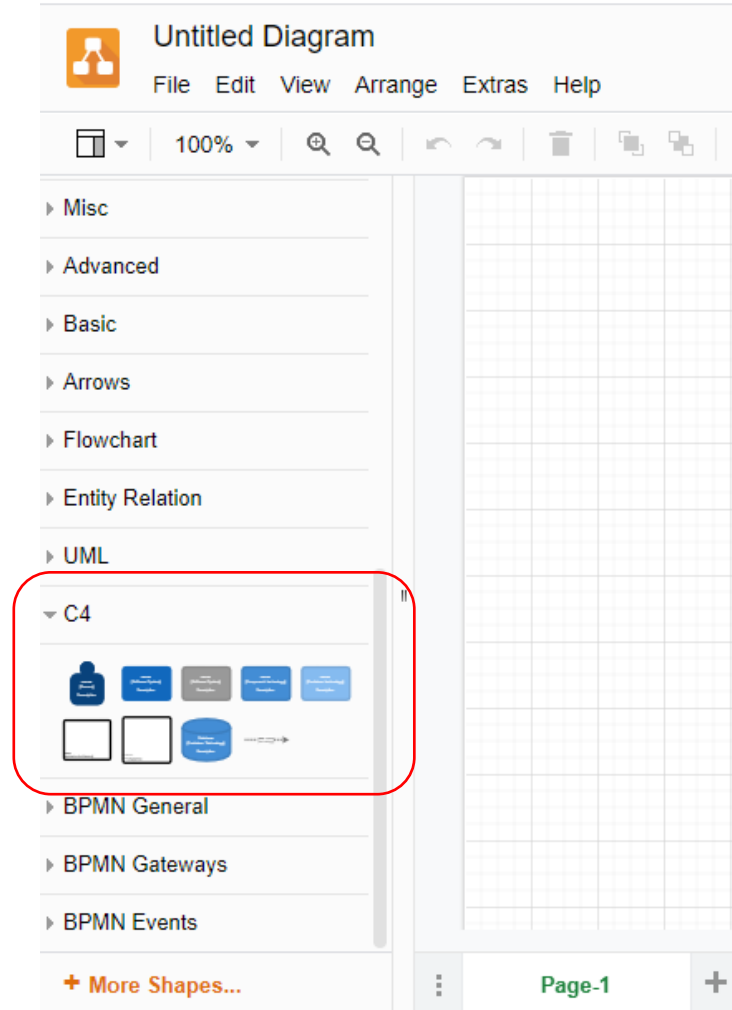
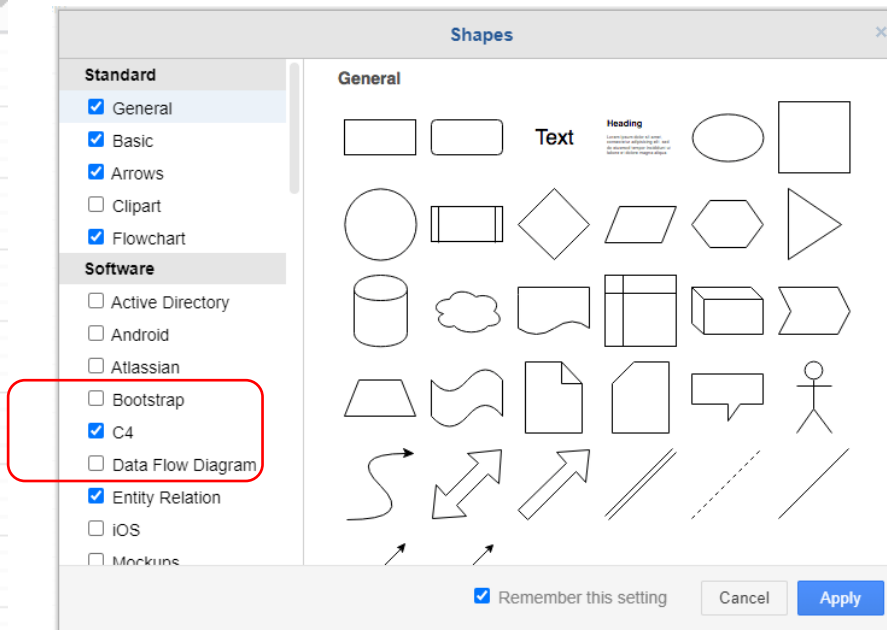
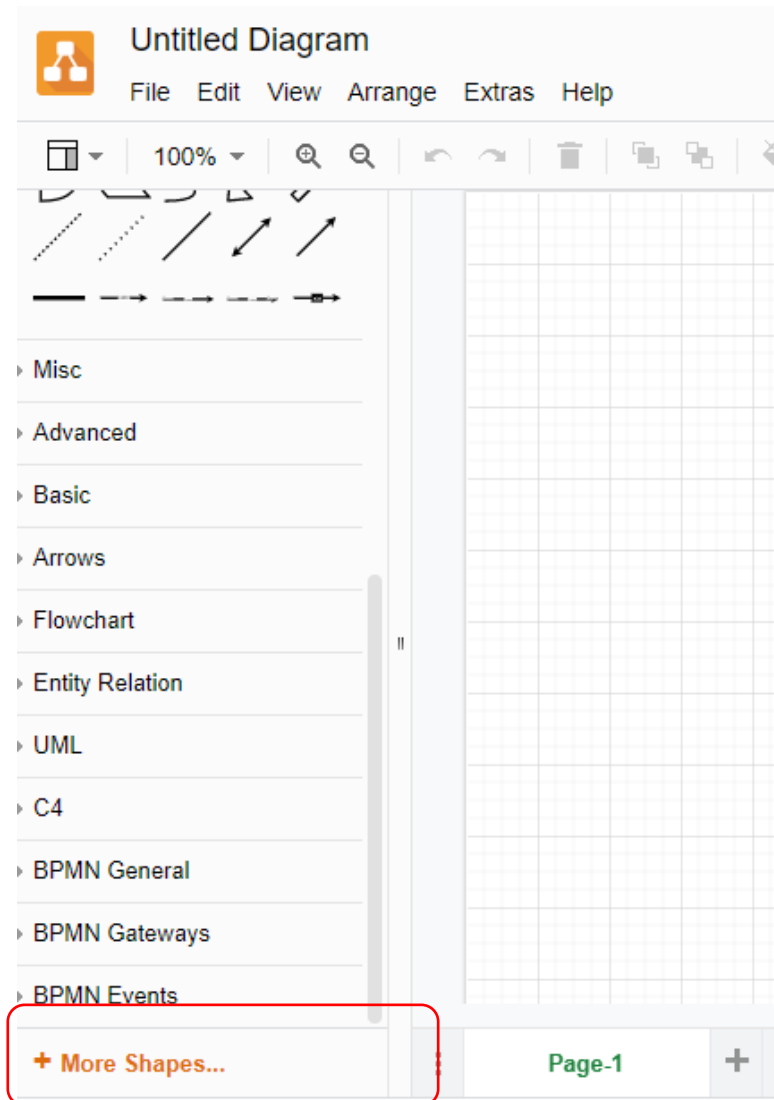


Reiniciar la aplicación



# Aplicación para crear los diagramas

UNIVERSIDAD DEL CAUCA – FIET  
DEPARTAMENTO DE SISTEMAS



1. Definición de arquitectura y etapa de diseño
2. ¿Cómo comunicar la arquitectura?
3. Modelo C4
4. Niveles del modelo C4



- ❖ El diagrama muestra el sistema software a desarrollar como una caja en el centro, rodeado por sus usuarios y los otros sistemas con los que interactúa.
- ❖ El enfoque está en las personas y sistemas de software en lugar de tecnologías, protocolos y otros detalles de bajo nivel.
- ❖ Este diagrama proporciona un panorama general y responde a las siguientes preguntas:

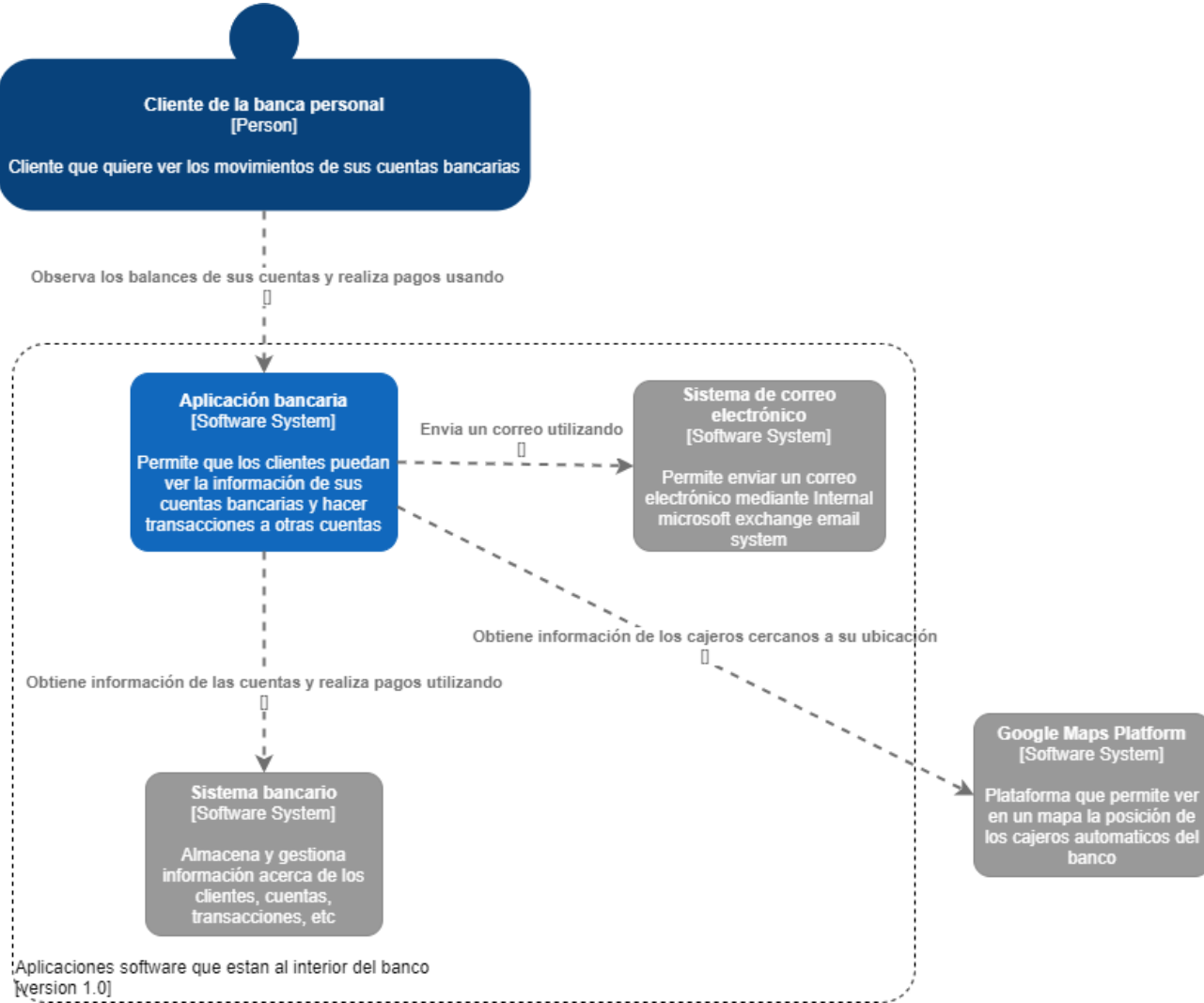
*1. ¿Cuál es el sistema de software que estamos construyendo?*

*2. ¿Qué usuarios lo usan?*

*3. ¿Con que sistemas existentes debemos interoperar (internos a la organización y externos a la organización)?*

- ❖ Se requiere desarrollar una aplicación para un sistema bancario que permita a los usuarios del banco realizar las siguientes funcionalidades:
  - Ver el resumen de sus cuentas.
  - Ver las transacciones realizadas.
  - Hacer transacciones entre cuentas del banco.
  - Observar los cajeros más cercanos en un mapa, pero únicamente desde el computador de escritorio.
  - Modificar sus datos personales como dirección, código postal, teléfono de contacto, correo electrónico.
- ❖ La aplicación debe poder ser accedida por el cliente desde una aplicación web, o desde una app móvil.
- ❖ Al analizar el sistema se han identificado el uso de patrones arquitectónicos tales como: capas, niveles, cliente-servidor, MVC

# Nivel 1. Diagrama de contexto



## Notación

Elemento	Descripción
	<b>Persona:</b> representa a uno de los usuarios humanos del sistema de software.
	<b>Software a desarrollar:</b> Software a implementar, el cual esta constituido de contenedores.
	<b>Software externo:</b> Software con el cual el sistema a desarrollar debe comunicarse.
	<b>Limite de la organización :</b> establece cuales son los sistemas nuevos y existentes en la organización
	<b>Relación entre elementos</b>

- ❖ El diagrama muestra los contenedores que conforman el sistema a desarrollar. También muestra las principales opciones tecnológicas y cómo los contenedores se comunican entre sí.
- ❖ Un contenedor es esencialmente un límite dentro del cual se ejecuta algún código o se almacenan algunos datos. Cada contenedor normalmente (pero no siempre) se ejecuta en su propio espacio de proceso.
  - **Aplicación web del lado del servidor** : una aplicación web Java EE que se ejecuta en Apache Tomcat.
  - **Aplicación web del lado del cliente** : una aplicación fronEnd desarrollada en Angular que se ejecuta en un contenedor docker, usando HTML, CSS, jQuery, etc.
  - **Aplicación de escritorio del lado del cliente**
  - **Aplicación móvil**
  - **Base de datos**

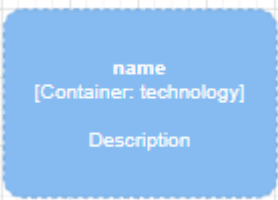
El diagrama responde a las siguientes preguntas:

- 1. ¿Cuáles son los subsistemas que conforman el sistema de software?*
- 2. ¿Cuáles son las decisiones tecnológicas de alto nivel?*
- 3. ¿Cómo se distribuyen las responsabilidades en el sistema?*
- 4. ¿Cómo se comunican los contenedores entre sí?*
- 5. Como desarrollador, ¿dónde necesito escribir código para implementar funcionalidades?*

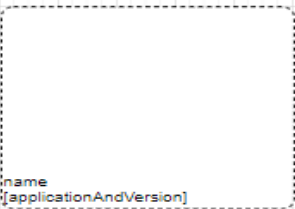
# Nivel 2. Diagrama de contenedores

## Notación

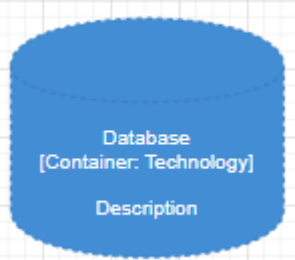
Elemento                      Descripción



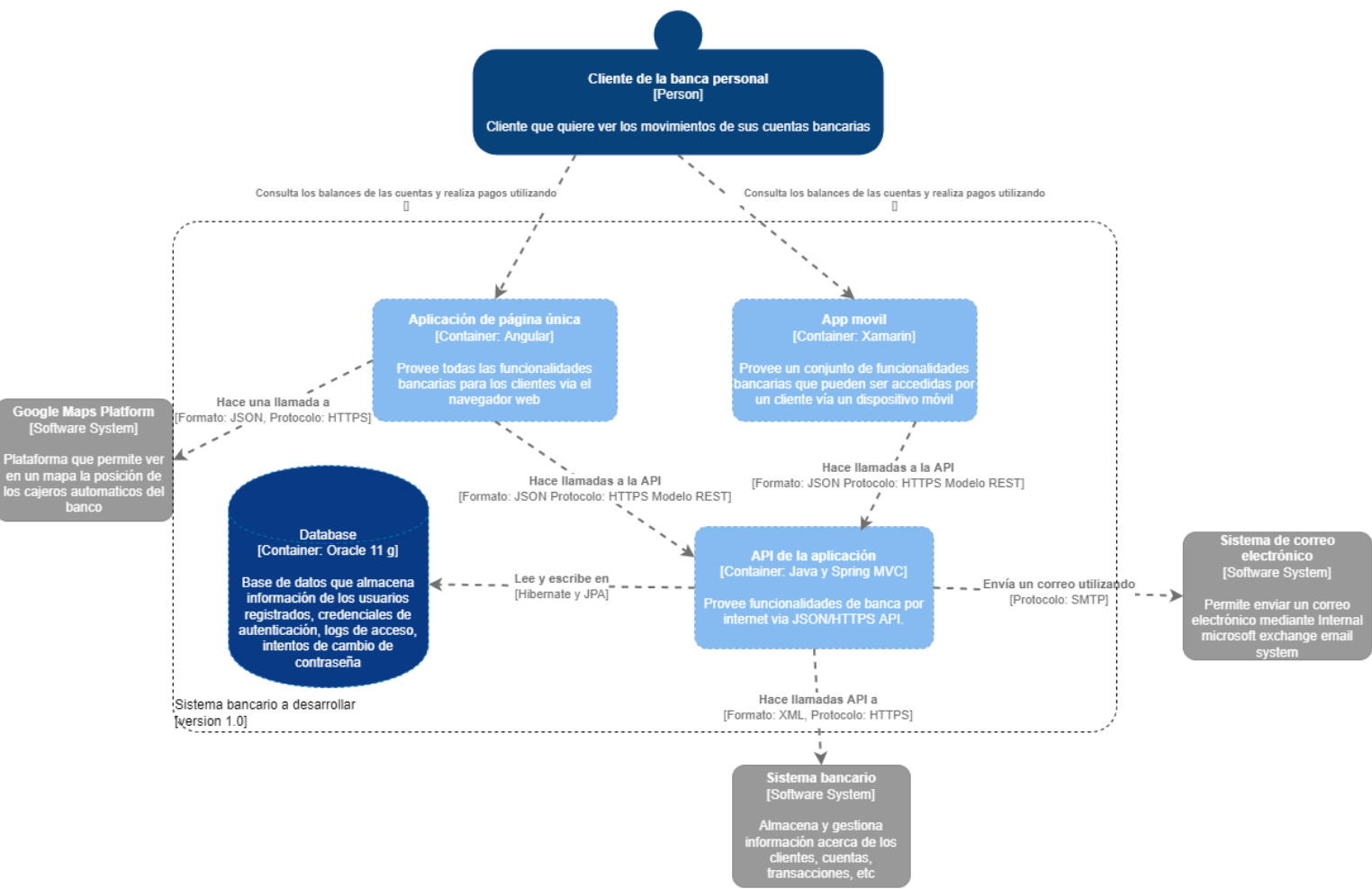
**Contenedor:** representa una aplicación o un almacén de datos que puede ejecutarse como un proceso independiente.



**Limite de una aplicación:** representa los contenedores que constituyen una aplicación a desarrollar.



**Base de datos contenedora:** representa un almacén de datos.





# Nivel 3. Diagrama de componentes

Un componente es una agrupación de funciones relacionadas encapsuladas detrás de una interfaz bien definida.

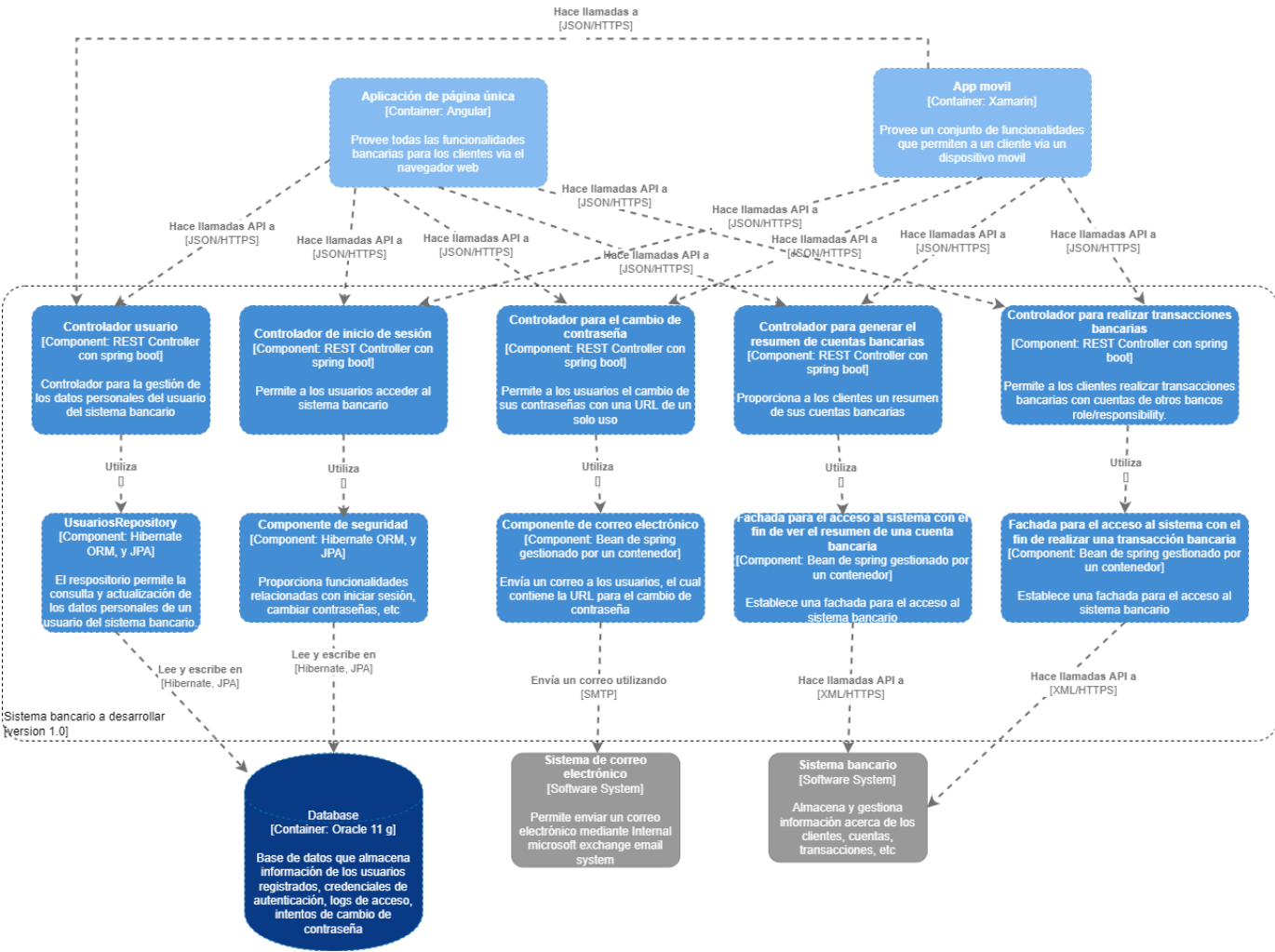
En un lenguaje como java o c# un componente es una colección de clases de implementación detrás de una interfaz

## Ejemplos de componentes

- Controladores
- Repositorios de acceso a base de datos
- Clases DAO que utilizan los repositorios.
- Clases fachada
- Clases de seguridad para inicio de sesión, cambios de contraseña, registro de usuario

El diagrama muestra por cada contenedor los principales componentes que los constituyen, sus responsabilidades y los detalles de tecnología e implementación.

# Nivel 3. Diagrama de componentes



## Notación

### Elemento

### Descripción



**Componente:** representa un conjunto de funcionalidades que son expuestas mediante una interface..



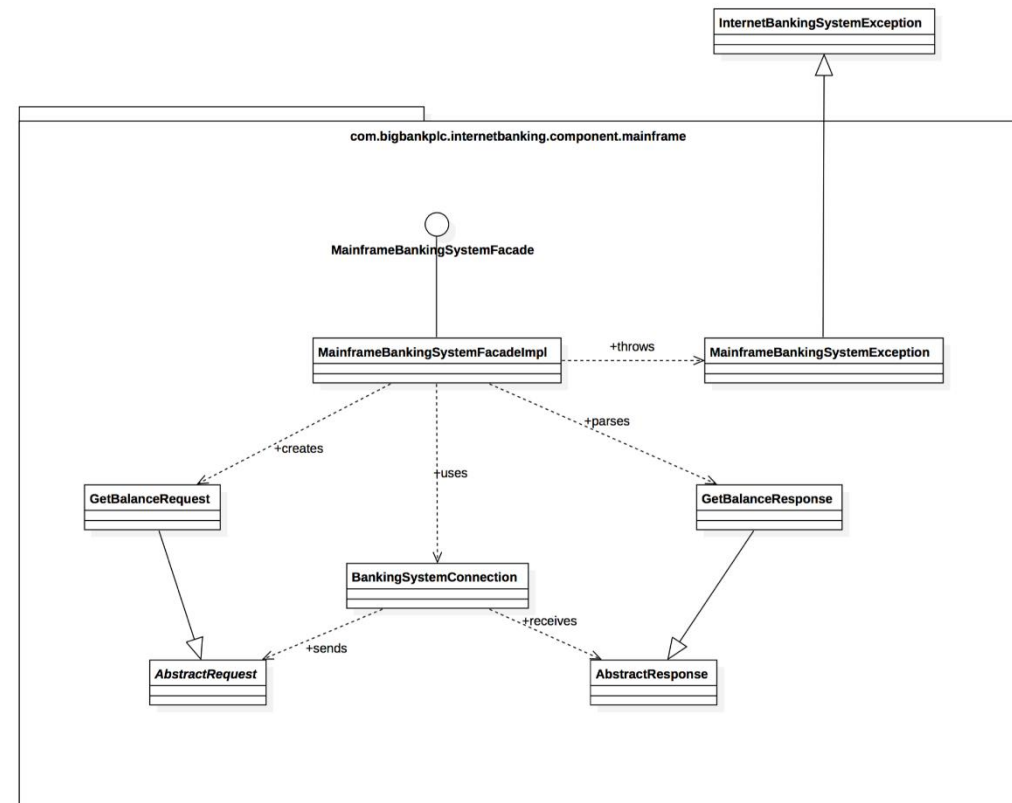
**Limite de un contenedor:** representa los componentes que constituyen un contenedor. Pueden encontrarse controladores, fachadas, repositorios,

En el modelo C4, los componentes no son unidades desplegables por separado.

## Nivel 4. Diagrama de código.

El diagrama muestra por cada componente como se implementa. En este nivel se utilizan diagramas como de clases, de relación, de secuencia. Este es un nivel de detalle opcional .

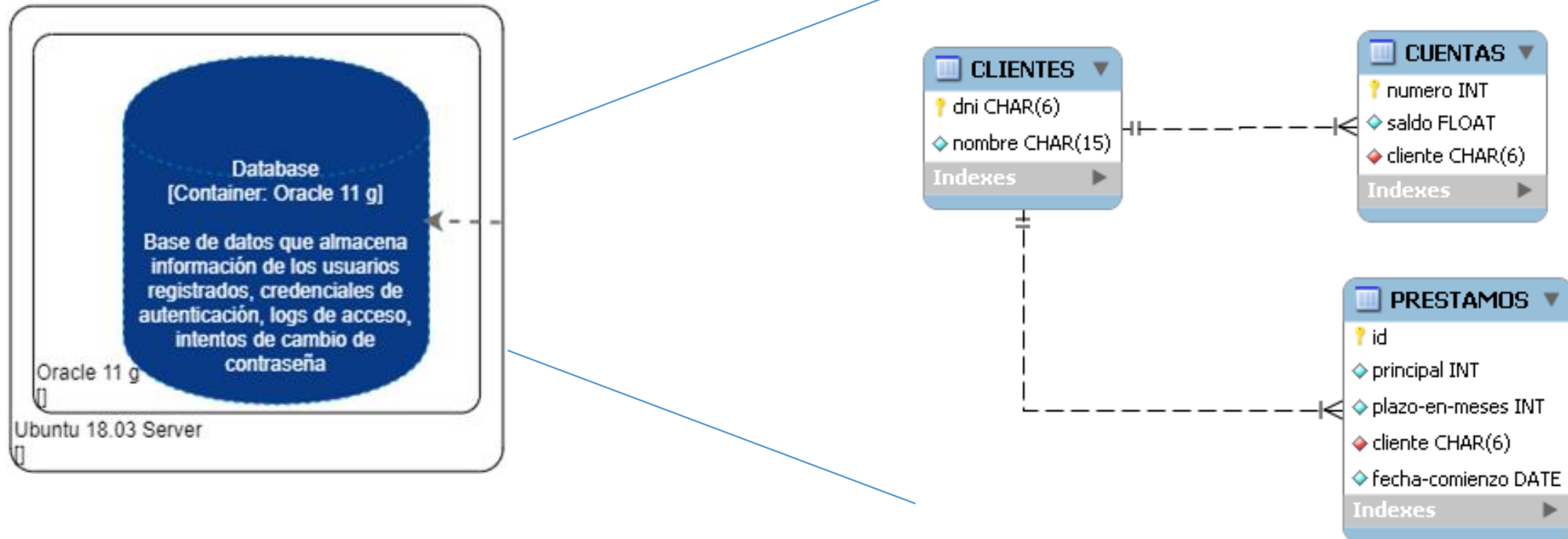
Muestra los elementos de código (interfaces y clases) que constituyen el componente fachada para el sistema bancario



# Diagramas complementarios:

## Diagrama de la base de datos

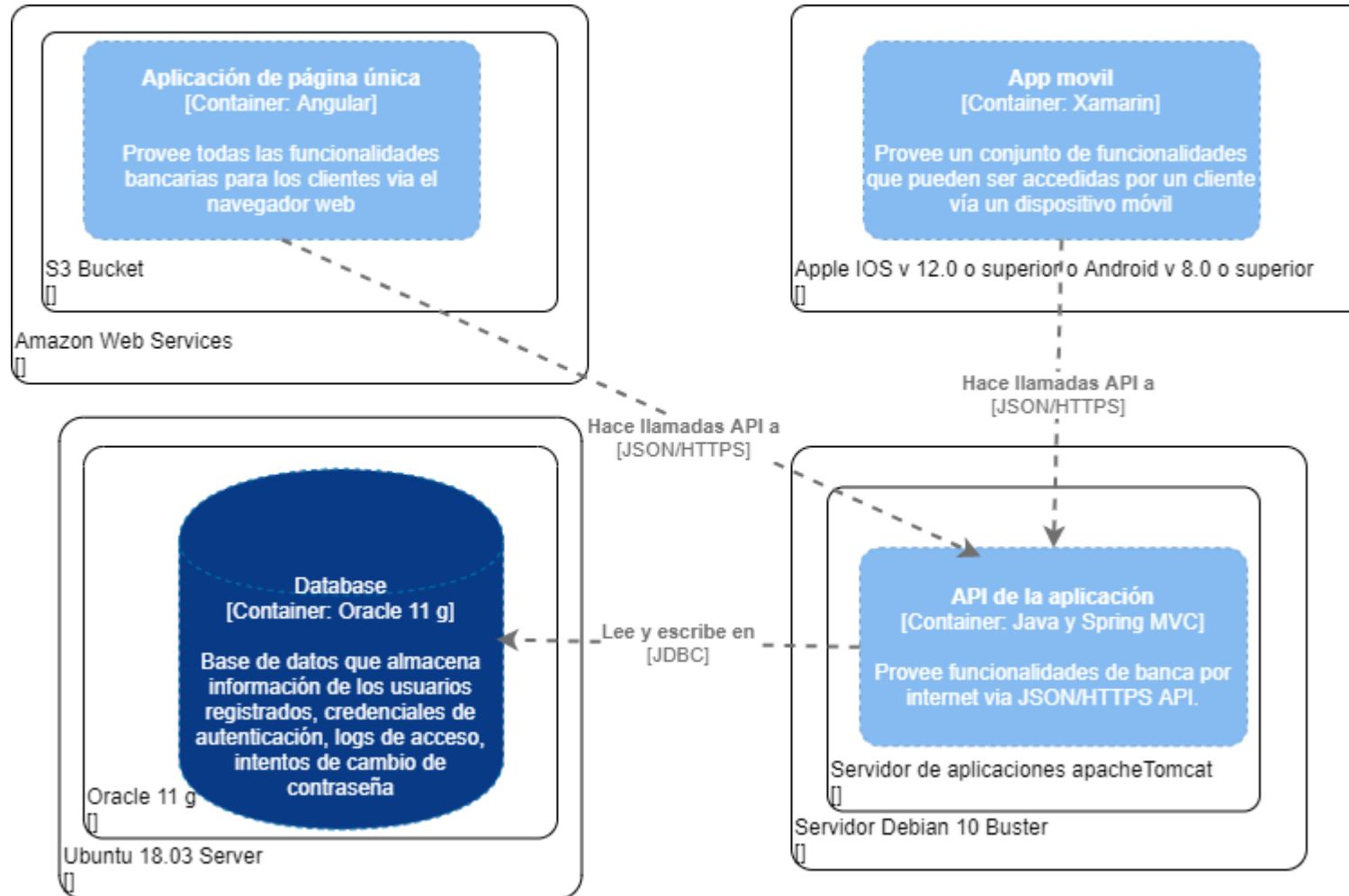
UNIVERSIDAD DEL CAUCA – FIET  
DEPARTAMENTO DE SISTEMAS



# Diagramas complementarios:

## Diagrama de despliegue

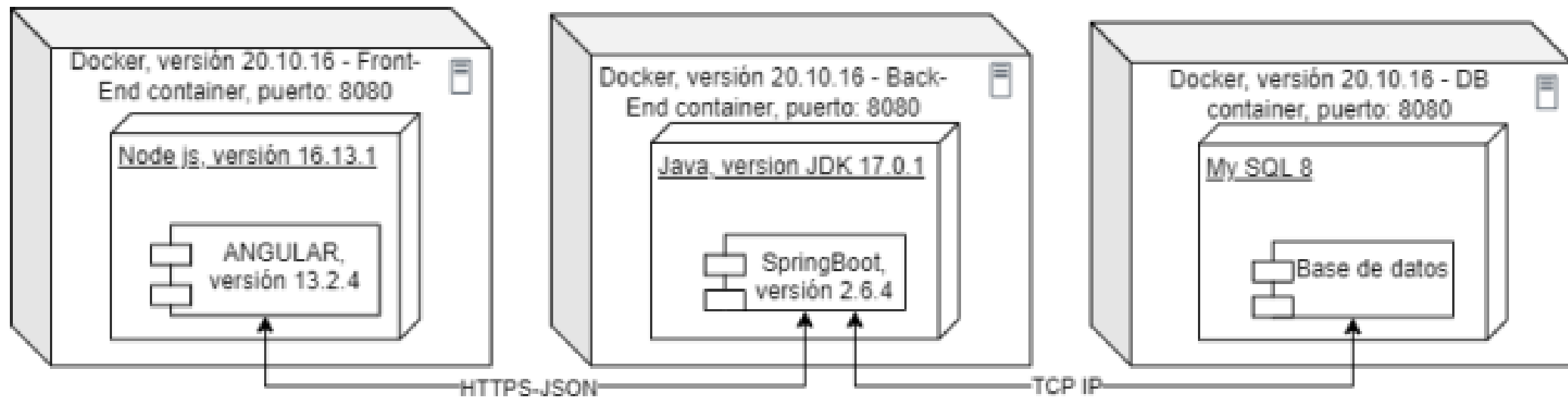
UNIVERSIDAD DEL CAUCA – FIET  
DEPARTAMENTO DE SISTEMAS



# Diagramas complementarios:

## Diagrama de despliegue

UNIVERSIDAD DEL CAUCA – FIET  
DEPARTAMENTO DE SISTEMAS



**Muchas gracias**  
**Preguntas**

