



ARQUITECTURA HEXAGONAL

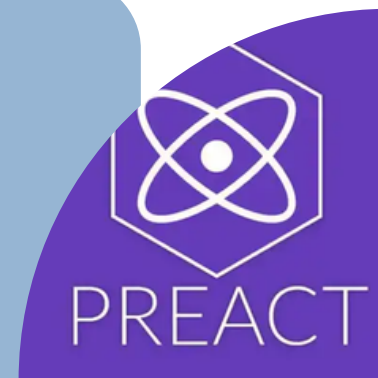
GRUPO - 1



HERRAMIENTAS

PREACT

Biblioteca de JavaScript ligera que ofrece una alternativa más rápida y compacta a React



CASSANDRA

Es una base de datos distribuida NoSQL, diseñada para manejar grandes cantidades de datos estructurados a través de múltiples servidores, sin un solo punto de falla.



DJANGO

Django es un framework de desarrollo web de alto nivel escrito en Python, diseñado para facilitar el desarrollo rápido de aplicaciones web.



GRAPHQL

Es un lenguaje de consulta para APIs, que permite a los clientes solicitar únicamente los datos que necesitan. Con GraphQL es posible obtener todos los datos requeridos en una sola consulta.



HEXAGONAL

DEFINICION:

- Es un estilo de arquitectura de software
- Diseñado para separar el núcleo lógico de una aplicación de los sistemas externos con los que interactúa (bases de datos, interfaces de usuario, APIs, etc.)
- Se estructura en torno a un núcleo central (la aplicación) y diversos "adaptadores" que permiten interactuar con servicios externos

Mantenibilidad

Escalabilidad

Flexibilidad tecnológica

Facilidad para realizar pruebas

Modularidad



Complejidad inicial



Curva de aprendizaje

Mayor número de capas

CARACTERISTICAS:



Desacoplamiento



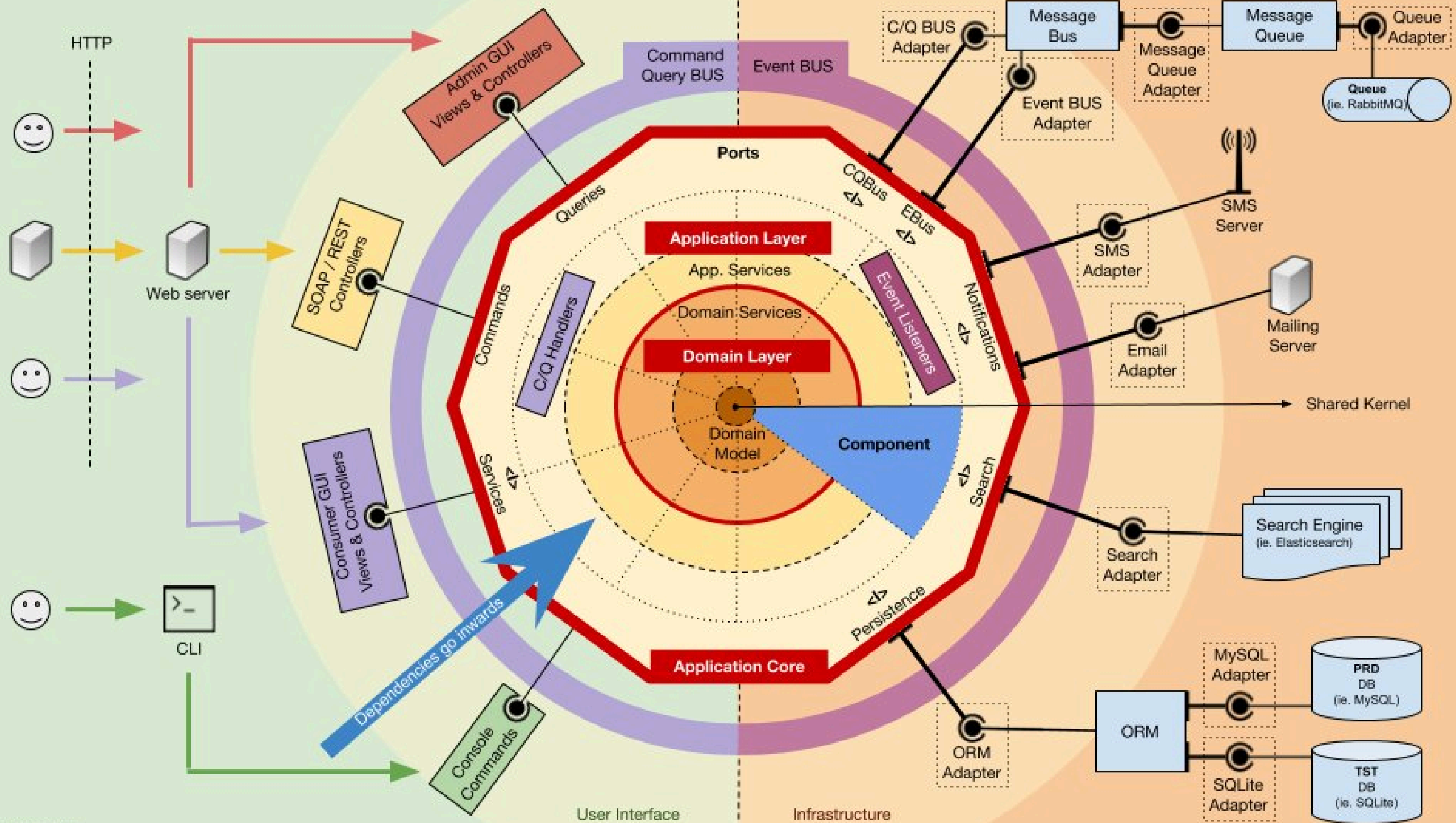
Independencia de
Infraestructura



Facilidad
para Testing

Primary/Driving Adapters

Secondary/Driven Adapters



However:

- The map is not the territory.
- Plans are worthless, but planning is everything.
- Understand all of this, but use only what you need.

GRAPHQL

Autodocumentación y facilidad de uso

Un único endpoint

Menos sobrecarga de datos

Compatibilidad con el tiempo real



Curva de aprendizaje

Complejidad en el servidor

Problemas de optimización



DEFINICION:

- Es un lenguaje de consulta para APIs
- Permite a los clientes solicitar únicamente los datos que necesitan.
- A diferencia de REST, con GraphQL es posible obtener todos los datos requeridos en una sola consulta.

CARACTERISTICAS:



Consultas
específicas



Consultas, mutaciones y
suscripciones



Flexibilidad en los
endpoints



API
autodocumentada

GRAPHQL VS REST



ÚNICO ENDPOINT

- Con una sola llamada **obtiene todos los datos y subdatos.**

{age, height, name}



- Usa **Query, Mutation y Subscription** para crear, eliminar, modificar y consultar datos.



- Pide **solo los datos que necesita** haciendo que la respuesta sea más rápida y eficiente.



MÚLTIPLES ENDPOINT

- Hace **varias llamadas** para obtener la información que se necesita.

/age /height /name



- Utiliza **verbos HTTP** (GET, POST, PUT, DELETE).



- Obtiene todos los datos**, no puede pedir solo lo que se necesita y esto afecta en el rendimiento.

Aplicaciones con altos volúmenes de escritura

Aplicaciones de IoT



REST y GraphQL son dos maneras distintas de enviar datos a través de HTTP.

CASSANDRA

DEFINICION:

- Base de datos **distribuida** NoSQL
- Manejar grandes cantidades de datos estructurados
- Múltiples servidores, sin un solo punto de falla.
- Alta disponibilidad
- Escalabilidad horizontal.

Alta disponibilidad y tolerancia a fallos

Escalabilidad horizontal
(Agregar nodos sin interrupciones)



Rendimiento de escritura

Distribución geográfica (replicación de datos)

Complejidad operativa
(configuración y mantenimiento)



Limitaciones en consultas complejas

CARACTERISTICAS:



Distribución -
Replicación



Escalabilidad



Lenguaje
Similar a SQL



Lecturas y
Escrituras



Nodos iguales -
Evita fallas

CASSANDRA

DAMA DE LAS BASES
DE DATOS NOSQL

Cassandra se ha convertido en uno de los sistemas de almacenamiento más interesantes y reseñables dentro del universo Big Data

CARACTERÍSTICAS

DISTRIBUIDA

Información repartida a lo largo de los nodos del cluster

ESCALA LINEALMENTE

Aporta predictibilidad a los sistemas

ESCALA DE FORMA HORIZONTAL

Escala el sistema añadiendo nuevos nodos

IMPLEMENTA UNA ARQUITECTURA PEER-TO-PEER

Elimina puntos de fallo único y no sigue patrones maestro-esclavo

MODELO DE DATOS

Combina propiedad de una base de datos clave-valor y una orientada a columnas

LENGUAJE
CQL

BIG DATA

¿QUIÉN USA CASSANDRA?

Su origen:
Facebook

2008
Liberada
como proyecto
Open Source

2010
Se convierte en
proyecto top-level
de Apache

paradig.ma/cassandra

Texto: Manuel Zaforas
Ilustración: Marta Ruiz

facebook

Para su sistema de búsqueda en la bandeja de entrada.

NETFLIX

Para su servicio de streaming

CISCO

Almacenamiento de actividad de usuarios en tiempo real.

IBM

Desarrollo de sistema de email.



Usa 75.000 nodos de Cassandra

DJANGO

DEFINICION:

- Framework de desarrollo web de alto nivel escrito en Python
- Facilitar el desarrollo rápido de aplicaciones web
- Sigue el principio de "No repetir código" (DRY)
- Enfoque Model-View-Template (MVT).

Desarrollo rapido
(herramientas pre configuradas)

Comunidad Amplia

Escalabilidad

Seguridad
(caracteristicas integradas para proteccion)



Curva de aprendizaje

Rigidez

Rendimiento



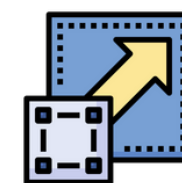
CARACTERISTICAS:



ORM Integrado
(integracion con BD)



Admin - panel
automático



Escalabilidad



Enrutamiento
(Url amigables y flexibles)

¿QUÉ ES **django** ?



Es uno de los **frameworks** más populares de **Python** para el desarrollo de **aplicaciones web** y usa el **patrón MVC**.



CARACTERÍSTICAS

ADMINISTRADOR

Cuenta con un panel de administración que permite **crear, consultar, actualizar y eliminar** contenido.



BASES DE DATOS

Tiene un **ORM** que se encarga de la comunicación con la base de datos. Es compatible con **PostgreSQL, MySQL, SQLite, etc.**



SEGURIDAD

Tiene opciones de protección contra ataques de **SQL injection** o **XSS** (cross site scripting).



The World Company (2003)
para gestionar sitio de noticias

EMPRESAS QUE USAN DJANGO

Pinterest



Reddit



Spotify



Instagram



Su nombre se inspira en el Guitarrista Django Reinhardt, del cual Adrian Holovaty, uno sus cocreadores, es admirador.

- Desarrollo rápido de aplicaciones web (panel admin)
- Sistemas de autenticación
- Aplicaciones escalables

PREACT

Rendimiento

Tamaño pequeño

Compatibilidad



Funcionalidades Avanzadas de react

Comunidad mas pequeña



DEFINICION:

- Preact es una biblioteca de JavaScript ligera
- Ofrece una alternativa más rápida y compacta a React.
- Se enfoca en ser pequeña (alrededor de 3KB gzip)

* con el objetivo de proporcionar la misma funcionalidad esencial que React pero menor tamaño

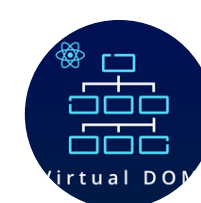
CARACTERISTICAS:



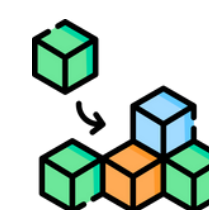
Tamaño
Reducido



Compatibilidad
con React



DOM virtual (Eficiencia
interfaz)



Componentes
modulares



Simple
integración

Casos de Aplicación

Pinterest:
mejorar el rendimiento de su versión móvil,
logrando tiempos de carga más rápidos sin
sacrificar la funcionalidad.

Uber:
sistema de gestión de conductores para
proporcionar una experiencia fluida con
tiempos de respuesta más cortos.

Casos de Uso

Aplicaciones web ligeras:
tiempos de carga rápidos y optimización en
el uso de recursos

Proyectos donde se busca minimizar el
tamaño:
dispositivos móviles o en regiones con
conexiones lentas.

Transición de aplicaciones pequeñas de
React a una versión más ligera.