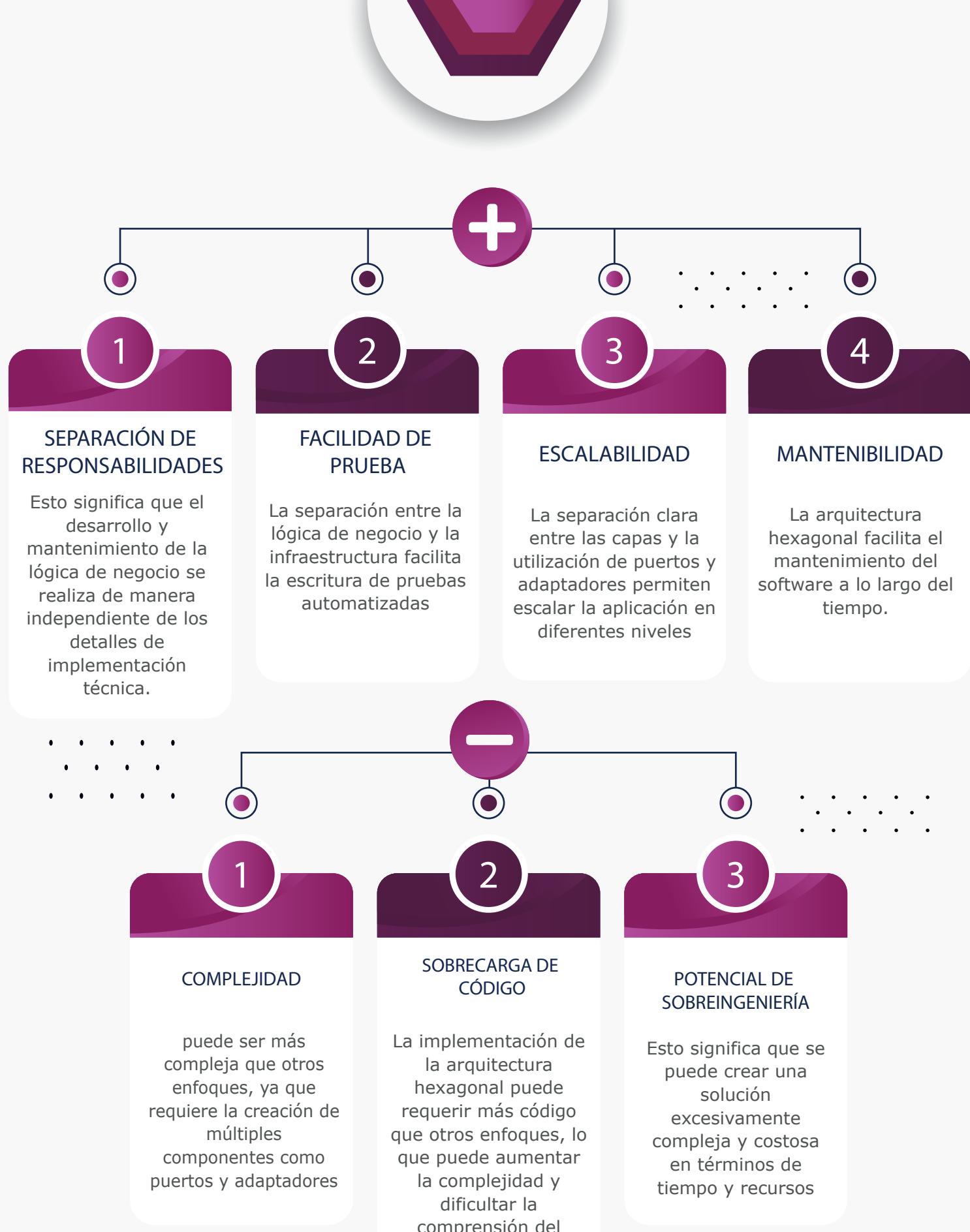
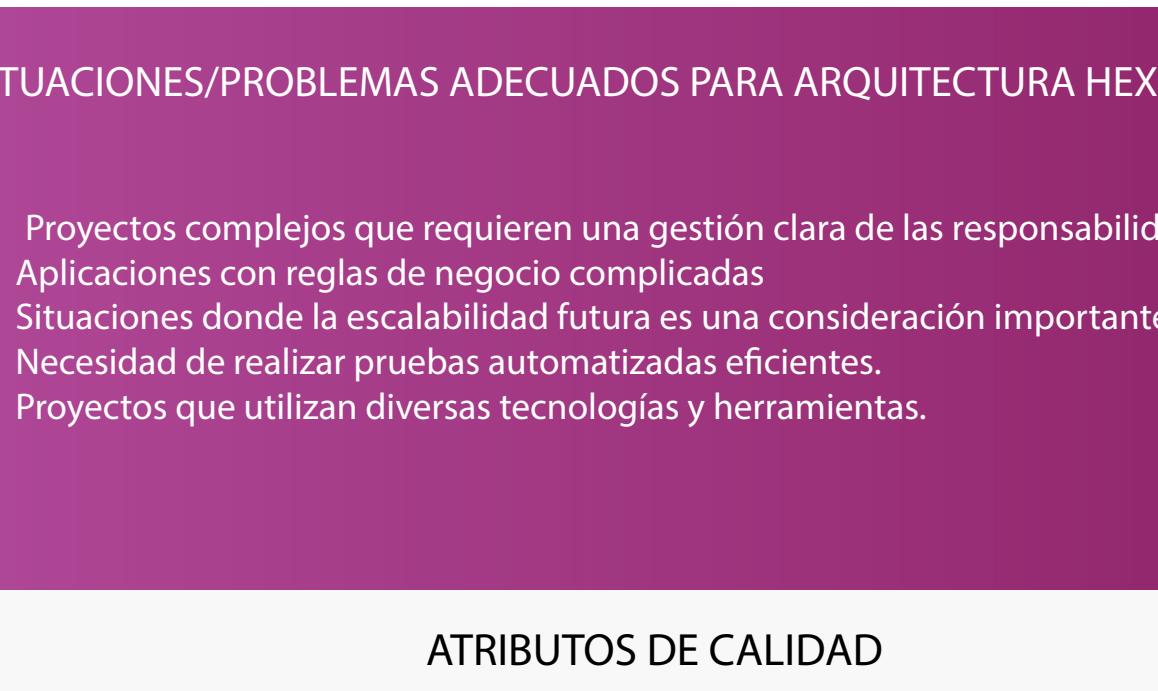
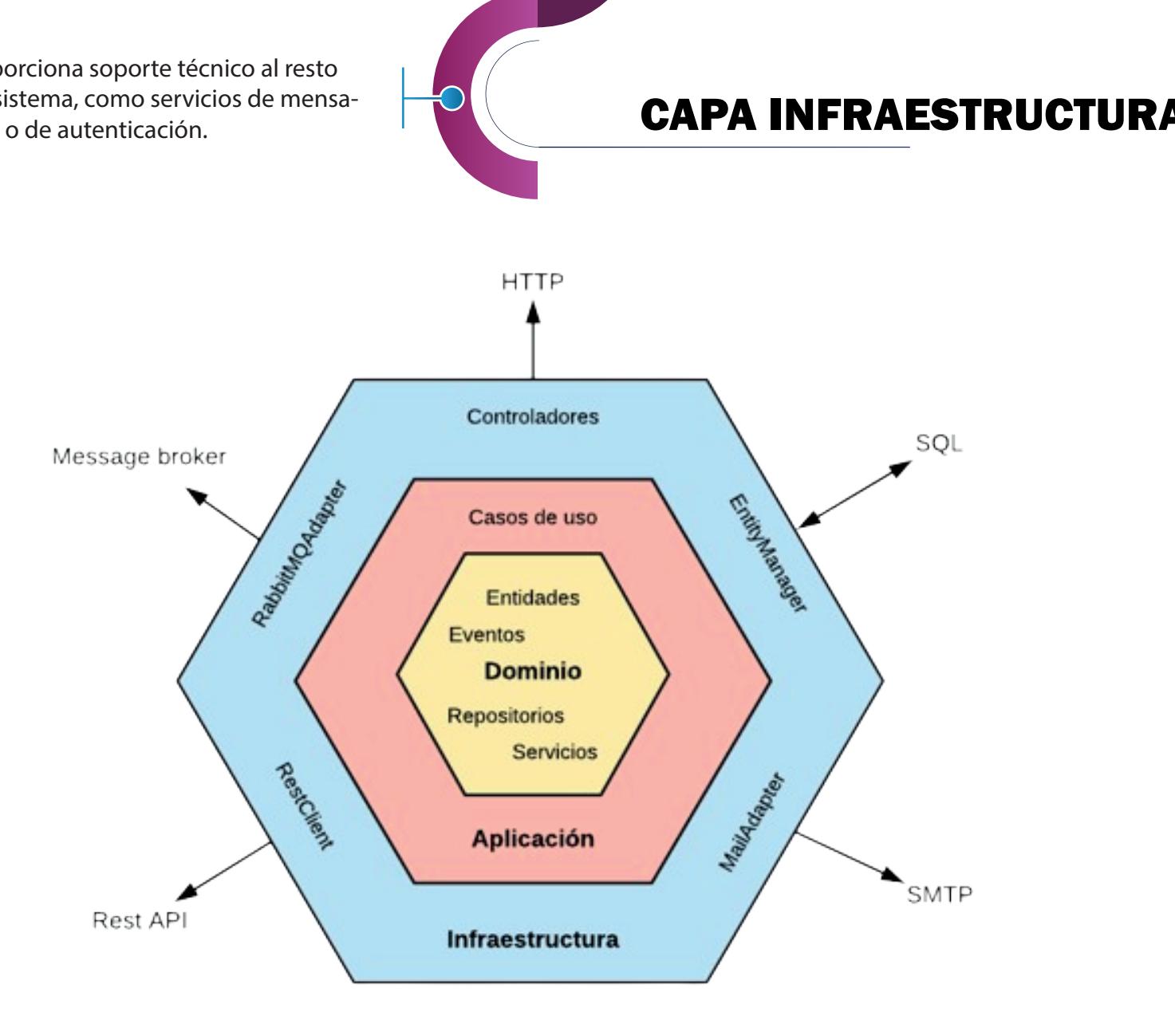


La Arquitectura Hexagonal es un patrón de diseño de software que busca la separación clara de las responsabilidades de cada componente de un sistema. Se visualiza como un hexágono, donde el núcleo del sistema se encuentra en el centro, rodeado de puertos y adaptadores



## COMPONENTES



## SITUACIONES/PROBLEMAS ADECUADOS PARA ARQUITECTURA HEXAGONAL

- Proyectos complejos que requieren una gestión clara de las responsabilidades
- Aplicaciones con reglas de negocio complicadas
- Situaciones donde la escalabilidad futura es una consideración importante.
- Necesidad de realizar pruebas automatizadas eficientes.
- Proyectos que utilizan diversas tecnologías y herramientas.

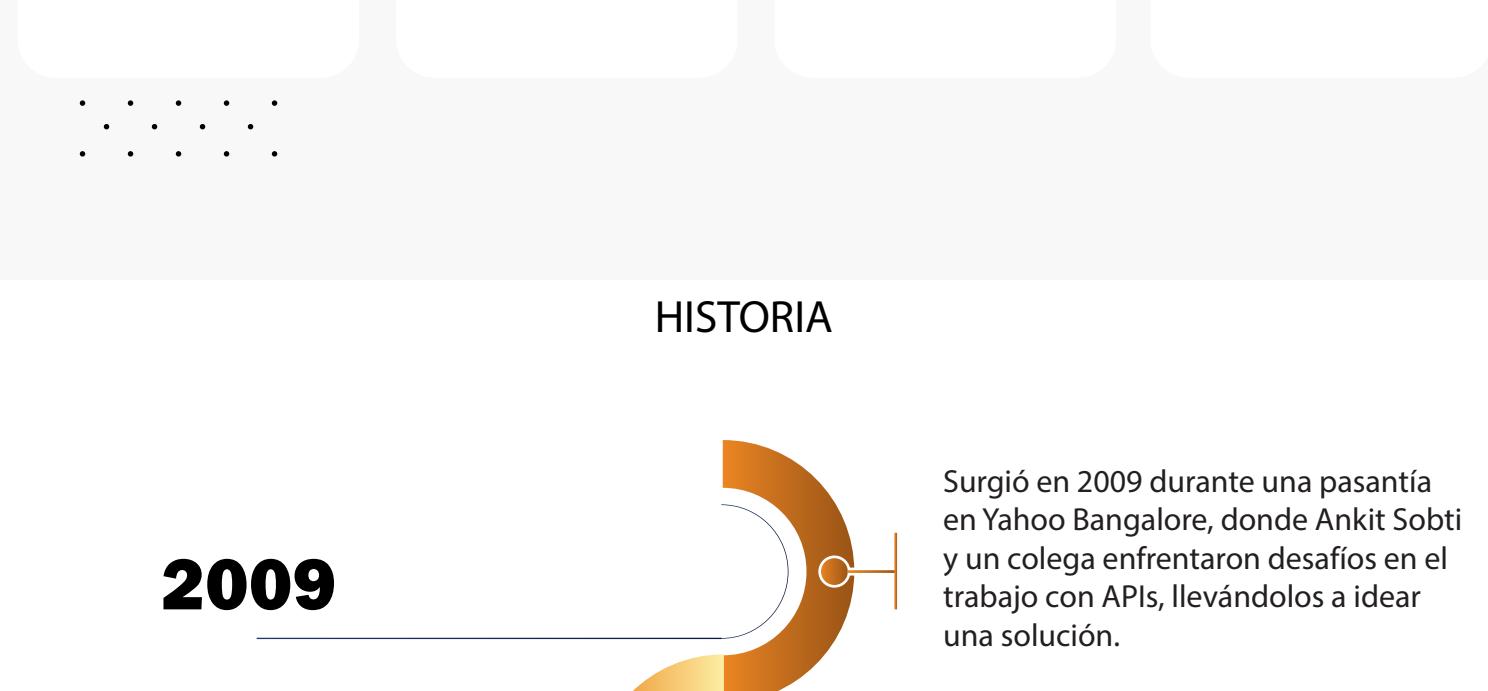
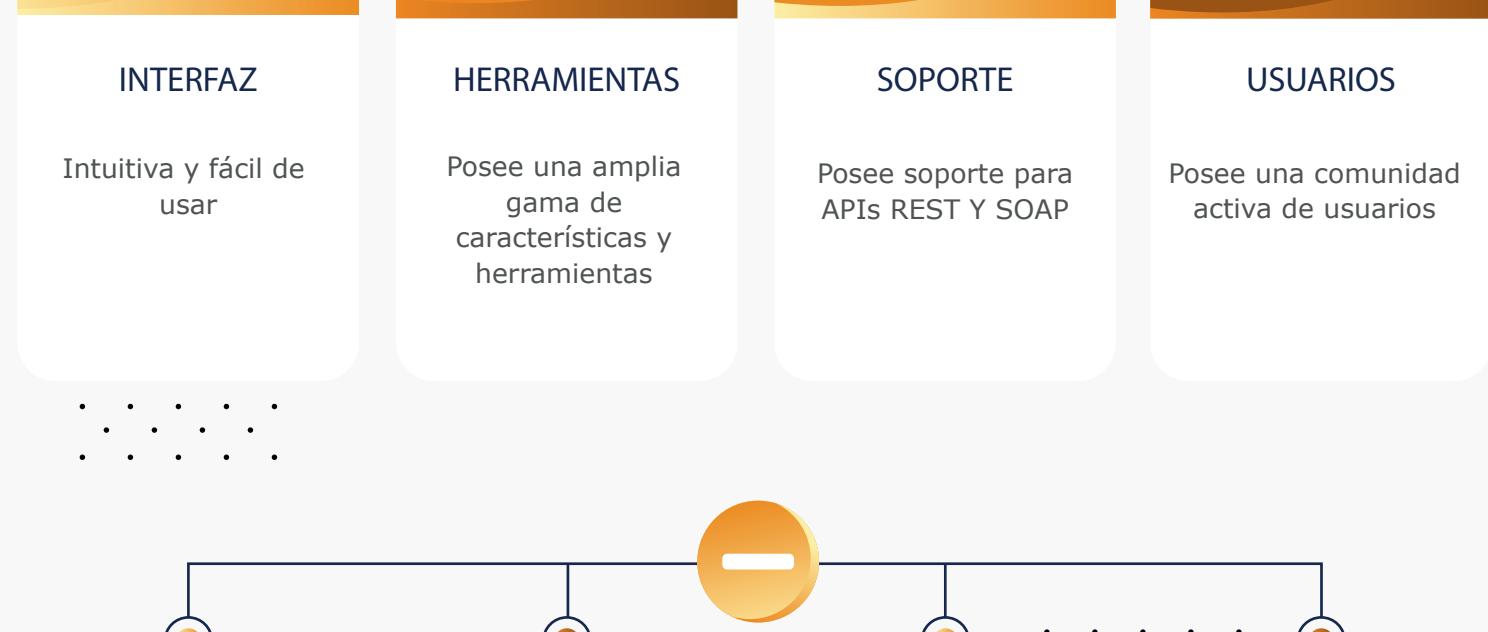
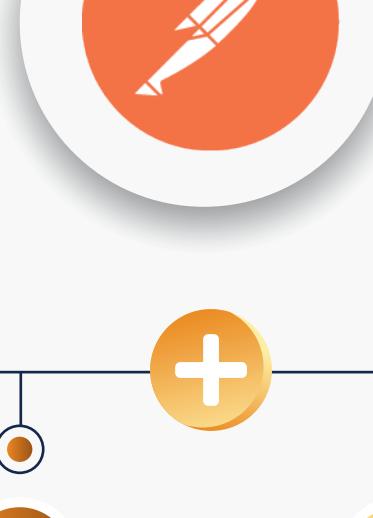
## ATRIBUTOS DE CALIDAD



## CASOS DE ESTUDIO

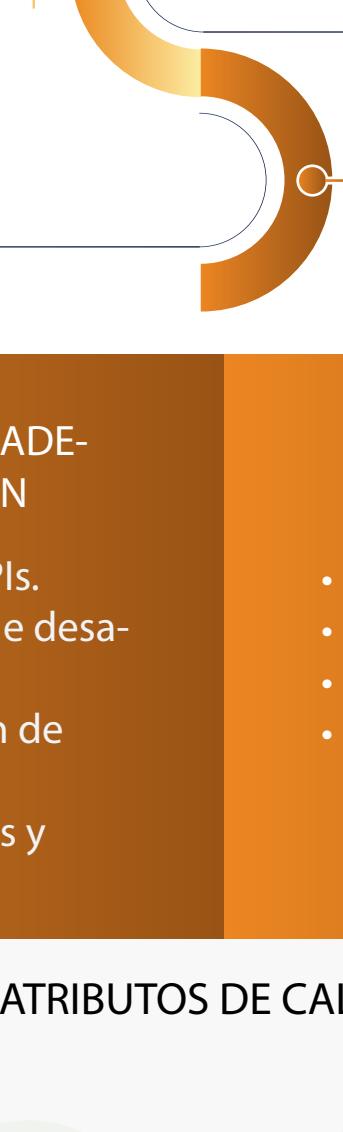


Postman es una plataforma integral para la construcción y utilización de APIs. Simplifica cada etapa del ciclo de vida de las APIs y fomenta la colaboración para que puedas crear APIs mejores y más rápidamente.



## HISTORIA

**2009**



Surgió en 2009 durante una pasantía en Yahoo Bangalore, donde Ankit Sobti y un colega enfrentaron desafíos en el trabajo con APIs, llevándolos a idear una solución.

## ENFOQUE USUARIO

Decisiones basadas en datos y una infraestructura tecnológica simple permitieron a Postman escalar sin comprometer la calidad del producto.

**HOY**

## PRIMEROS PASOS

La retroalimentación de los usuarios fue clave, guiando la mejora continua del producto y estableciendo una mentalidad centrada en el usuario desde el principio.

## ENFOQUE Y EVOLUCIÓN

Con una comunidad de más de 17 millones de desarrolladores, Postman sigue comprometido con su misión de simplificar el desarrollo y la gestión de APIs para todos.

## SITUACIONES/PROBLEMAS ADECUADOS PARA POSTMAN

- Desarrollo y pruebas de APIs.
- Colaboración en equipos de desarrollo de APIs.
- Gestión de documentación de APIs.
- Automatización de pruebas y monitoreo de APIs

## ASPECTOS RELEVANTES

- Repositorios de API
- Gobernanza
- Espacios de trabajo
- Integraciones

## ATRIBUTOS DE CALIDAD

01.



USABILIDAD

02.



MANTENIBILIDAD

03.



FLEXIBILIDAD

04.



SEGURIDAD

## OPERABILIDAD

Postman proporciona una interfaz intuitiva y fácil de usar que permite a los usuarios operar y controlar la plataforma con facilidad.

## C. PARA SER PROBADO

Postman ofrece una facilidad significativa para establecer criterios de prueba para sistemas o componentes y llevar a cabo las pruebas necesarias para determinar si se cumplen esos criterios.

## ADAPTABILIDAD

Postman es altamente adaptable y puede integrarse con una variedad de herramientas y servicios en el pipeline de desarrollo de software.

## CONFIDENCIALIDAD

Postman asegura que los datos de los usuarios y los detalles de las APIs sean confidenciales y estén protegidos contra accesos no autorizados.

## CASOS DE ESTUDIO

**PayPal**



Flutter es un framework de código fuente abierto de desarrollo de aplicaciones creado por Google que sirve para crear aplicaciones nativas visualmente atractivas en varias plataformas.



## HISTORIA



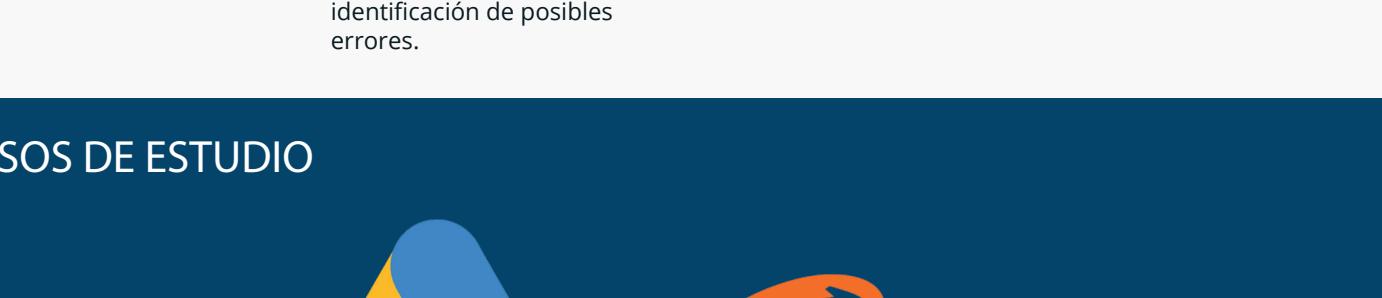
## SITUACIONES/PROBLEMAS ADECUADOS PARA FLUTTER

- Desarrollo Multiplataforma:
- UI/UX de Alta Fidelidad
- Prototipado Rápido

## ASPECTOS TECNOLÓGICOS CLAVE DE FLUTTER

- Motor de Renderizado
- Marco de Trabajo
- Biblioteca de Widgets
- Pruebas y Depuración
- Lenguaje y Compatibilidad

## ATRIBUTOS DE CALIDAD



## CASOS DE ESTUDIO



REST/JSON es un estilo arquitectónico para diseñar APIs web que utiliza JSON como formato de intercambio de datos. Este enfoque se basa en los principios de REST (Transferencia de Estado Representacional), que promueve una arquitectura de software orientada a recursos y utiliza métodos estándar de HTTP para manipular estos recursos.



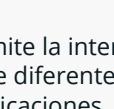
## SITUACIONES/PROBLEMAS ADECUADOS PARA REST/JSON

- Desarrollo y pruebas de APIs.
- Colaboración en equipos de desarrollo de APIs.
- Gestión de documentación de APIs.
- Automatización de pruebas y monitoreo de APIs.

REST/JSON se compara favorablemente con otros estándares utilizados en el diseño de APIs web. A diferencia de algunos enfoques más complejos, como SOAP (Simple Object Access Protocol), REST/JSON ofrece una arquitectura simple y liviana que se adapta bien a las necesidades de las aplicaciones web modernas.

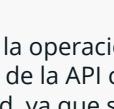
## ATRIBUTOS DE CALIDAD

01.



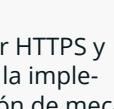
COMPATIBILIDAD

02.



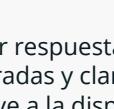
USABILIDAD

03.



SEGURIDAD

04.



FIABILIDAD

### INTEROPERABILIDAD

permite la interacción entre diferentes sistemas y aplicaciones, lo que favorece la coexistencia y la interoperabilidad.

### OPERABILIDAD

Facilita la operación y control de la API con facilidad, ya que sigue principios claros y predecibles.

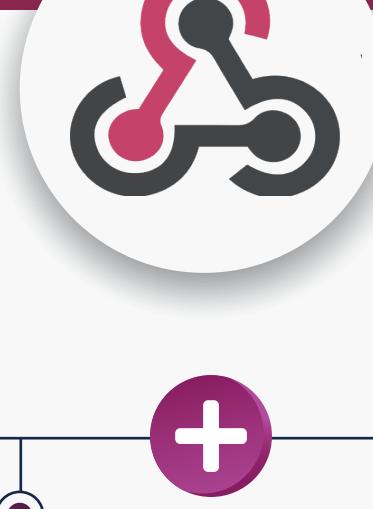
### CONFIDENCIALIDAD

Al utilizar HTTPS y permitir la implementación de mecanismos de autenticación y autorización.

### DISPONIBILIDAD

Al ofrecer respuestas estructuradas y claras, contribuye a la disponibilidad de la API al garantizar que los recursos estén accesibles para su uso cuando sea necesario.

Un webhook es una función de devolución de llamada que utiliza el protocolo HTTP para permitir la comunicación ligera entre dos interfaces de programación de aplicaciones (API) a través de eventos.



## HISTORIA

### CONCEPTO



Se popularizó en 2007 con una publicación en el Blog de Jeff Lindsay

los webhooks han sido adoptados por numerosas plataformas y servicios en línea debido a su capacidad para simplificar y automatizar la comunicación entre aplicaciones.

### Uso inicial



2007+

En el ámbito del desarrollo web y la integración de sistemas.

Su versatilidad y eficacia los han llevado a ser ampliamente utilizados en diversas áreas, incluyendo el marketing digital, las ventas en línea, la gestión de infraestructura, entre otros.

### Uso avanzado



### SITUACIONES/PROBLEMAS ADECUADOS PARA webhooks

- Marketing digital:
- Gestión de infraestructura
- Automatización de procesos empresariales

### POSIBILIDADES ME OFRECE WEBHOOK

- Sincronizar sistemas en tiempo real
- Enviar cualquier tipo de notificación
- Procesar los datos como quieras
- Crear informes

### 01. MANTENIBILIDAD



### 02. USABILIDAD



### 03. EFICIENCIA



### 04. FIABILIDAD



#### FACILIDAD DE C.

Requieren una configuración mínima y utilizan HTTP para la transferencia de datos, lo que facilita su implementación y mantenimiento

#### AUTOMATIZACIÓN

Permiten la automatización de tareas y flujos de trabajo al activar eventos específicos

#### AUMENTO E.

Al eliminar la necesidad de intervención manual y reducir los tiempos de espera para la transferencia de datos, los Webhooks aumentan la eficiencia

#### TIEMPO REAL

Los Webhooks envían datos de manera instantánea cuando se produce un evento, lo que garantiza la entrega oportuna de la información

## ATRIBUTOS DE CALIDAD

01.



MANTENIBILIDAD

02.



USABILIDAD

03.



EFICIENCIA

04.

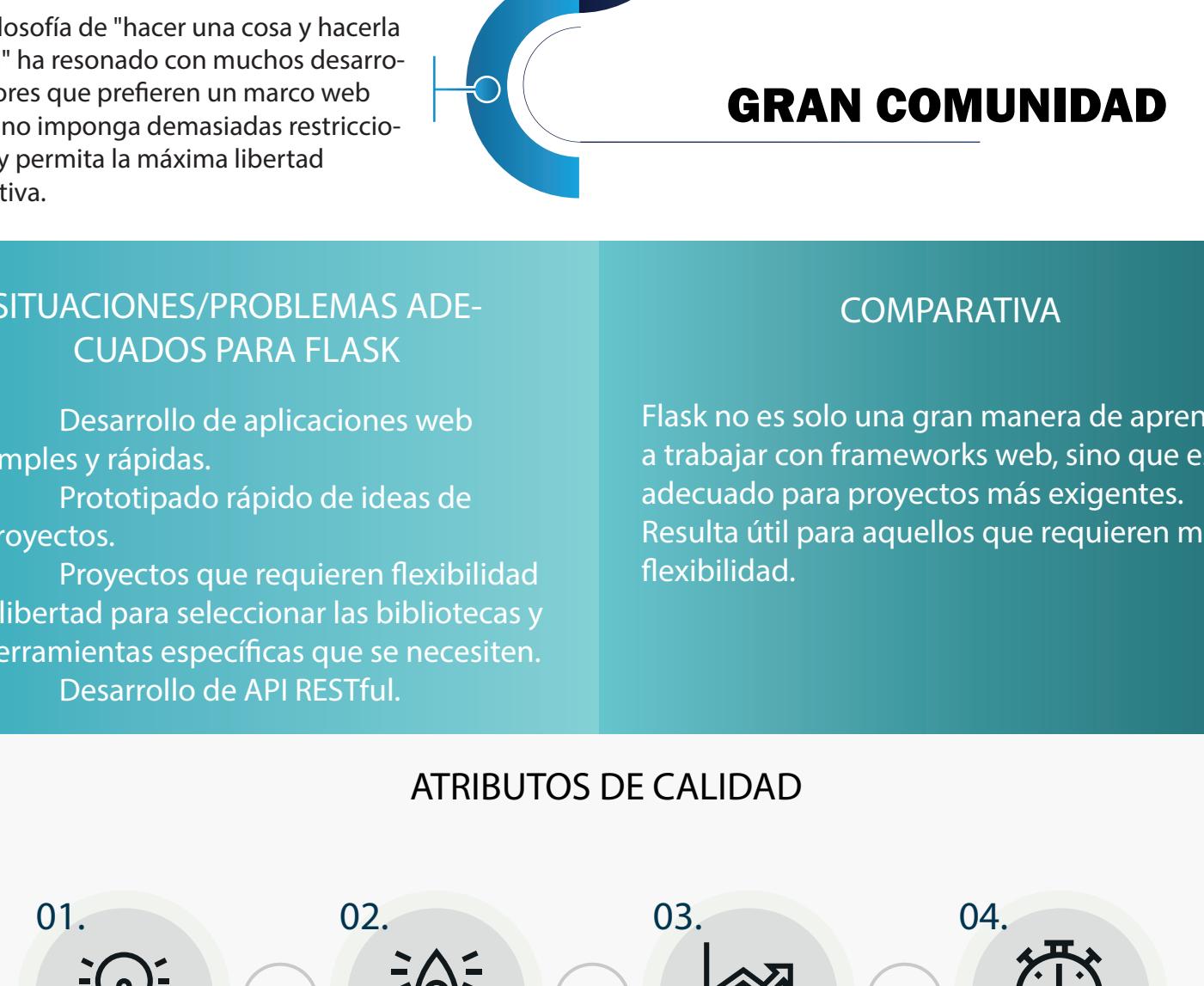


FIABILIDAD

Flask es un marco web ligero desarrollado en Python, adecuado tanto para principiantes como para profesionales. Flask se limita a incluir solo lo esencial, pero los usuarios pueden implementar bibliotecas externas para expandir su funcionalidad.



## HISTORIA



## SITUACIONES/PROBLEMAS ADECUADOS PARA FLASK

- Desarrollo de aplicaciones web simples y rápidas.
- Prototipado rápido de ideas de proyectos.
- Proyectos que requieren flexibilidad y libertad para seleccionar las bibliotecas y herramientas específicas que se necesiten.
- Desarrollo de API RESTful.

## COMPARATIVA

Flask no es solo una gran manera de aprender a trabajar con frameworks web, sino que es adecuado para proyectos más exigentes. Resulta útil para aquellos que requieren más flexibilidad.

## ATRIBUTOS DE CALIDAD

01. COMPATIBILIDAD	02. SEGURIDAD	03. FIABILIDAD	04. FIABILIDAD
INTEROPERABILIDAD	CONFIDENCIALIDAD	AUSENCIA FALLOS	DISPONIBILIDAD
Flask puede interoperar con otros sistemas o componentes al permitir el intercambio de información a través de APIs y servicios web	Flask protege la información confidencial asegurando que solo los usuarios autorizados puedan acceder a ella, garantizando la privacidad de los datos.	Flask es conocido por su estabilidad y confiabilidad, permitiendo que los sistemas desarrollados con él funcionen sin fallos bajo condiciones normales de operación.	Flask tiene la capacidad de estar operativo y accesible para su uso cuando se requiere, asegurando que las aplicaciones web desarrolladas con él estén disponibles para los usuarios.

## CASOS DE ESTUDIO



Oracle Database es un sistema de gestión de bases de datos relacional (RDBMS) desarrollado y comercializado por Oracle Corporation. Es uno de los sistemas de gestión de bases de datos más populares y ampliamente utilizados en el mundo empresarial. Oracle Database permite a las empresas almacenar, gestionar y recuperar grandes volúmenes de datos de manera eficiente y segura. Utiliza el lenguaje de consulta estructurado (SQL)



## HISTORIA



## SITUACIONES/PROBLEMAS ADECUADOS PARA ORACLE

- Empresas de todos los tamaños
- Aplicaciones empresariales críticas
- Entornos de desarrollo
- Proyectos de almacenamiento de datos grandes

## ATRIBUTOS DE CALIDAD

01.



COMPATIBILIDAD

02.



MANTENIBILIDAD

03.



SEGURIDAD

04.



FIABILIDAD

Ofrece una alta compatibilidad con una amplia gama de plataformas y aplicaciones, lo que facilita su integración en entornos heterogéneos

Está diseñado para ser fácilmente mantenible, lo que permite a los administradores de bases de datos realizar modificaciones efectivas y eficientes en la estructura y configuración

La seguridad es una prioridad en Oracle Database, que ofrece funciones avanzadas para proteger la confidencialidad, integridad y disponibilidad de los datos almacenados en la base de datos.

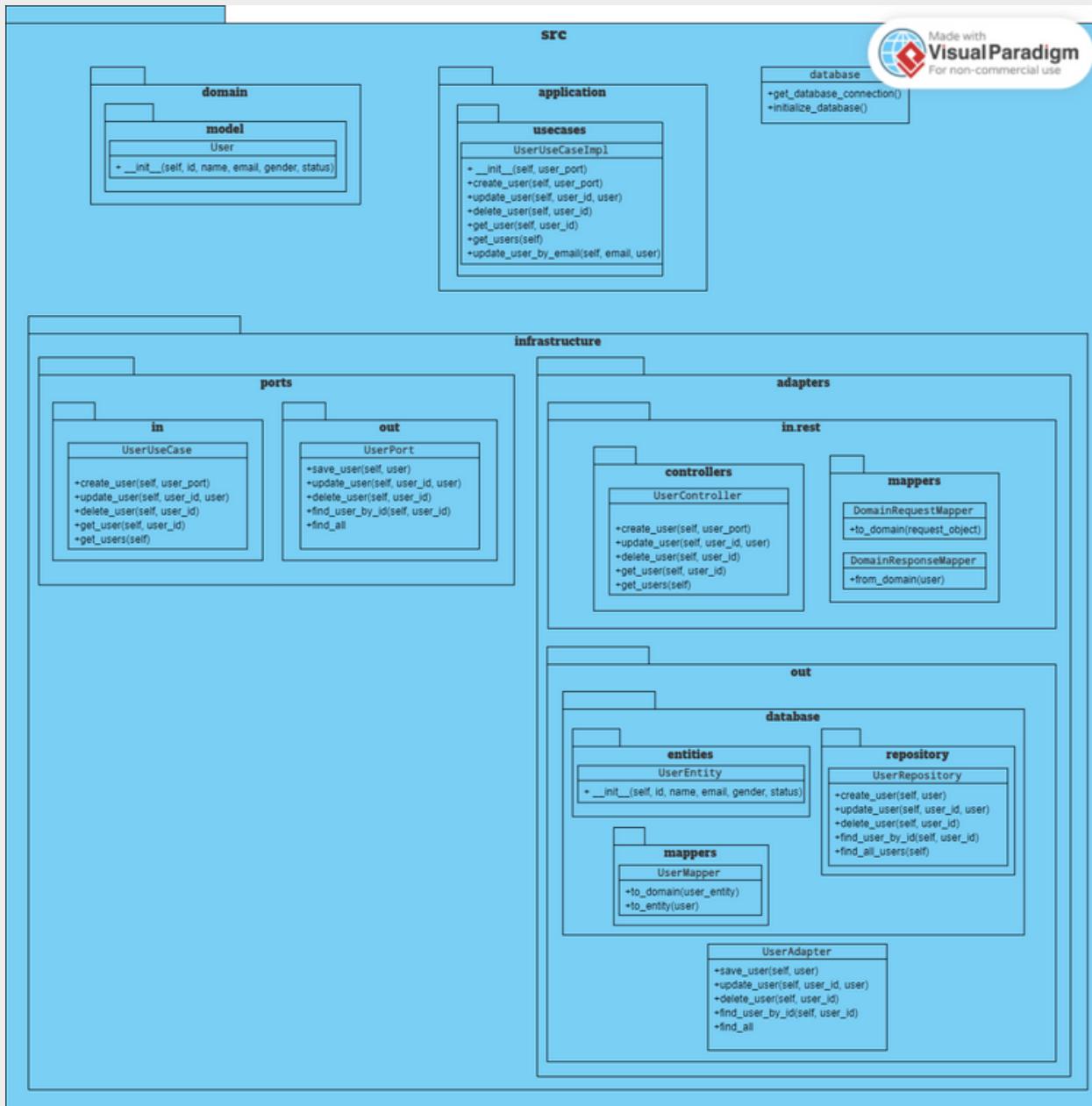
se destaca por su fiabilidad y estabilidad, lo que asegura que el sistema pueda desempeñar sus funciones especificadas durante largos períodos de tiempo sin interrupciones ni fallos.

## CASOS DE ESTUDIO



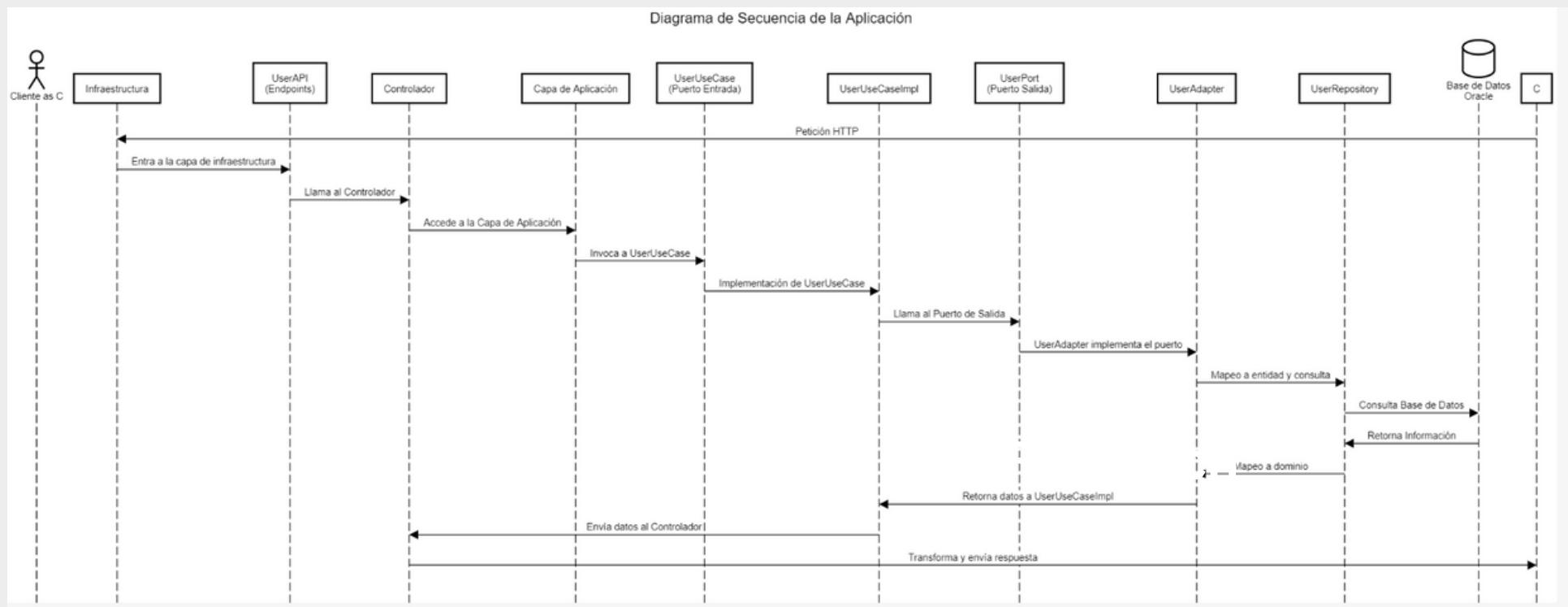
Royal Flying Doctor Service

# DIAGRAMA DE CLASES

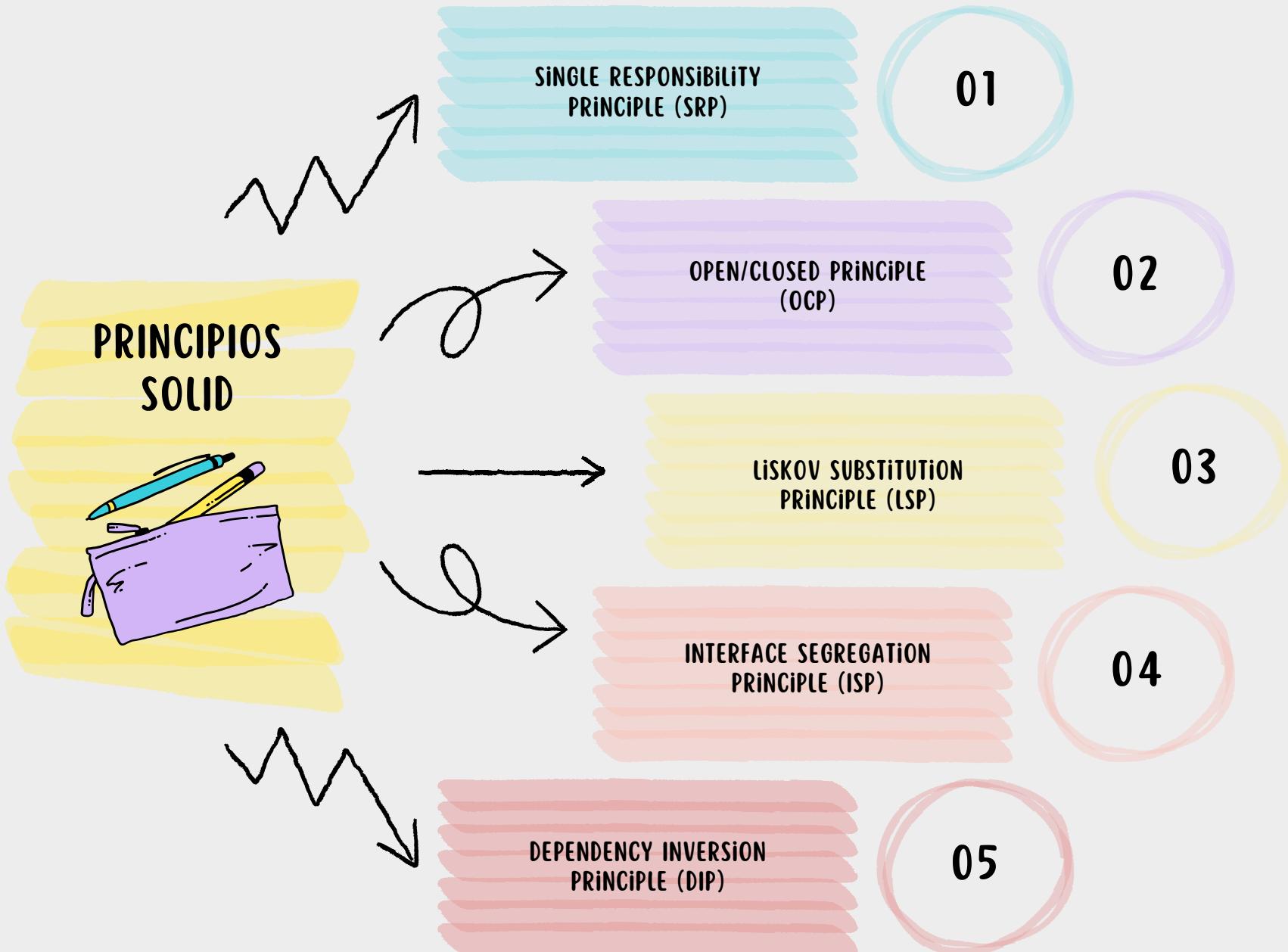


[Link del diagrama de clases](#)

# DIAGRAMA DE SECUENCIA



[Link del diagrama de secuencia](#)



## **SINGLE RESPONSIBILITY PRINCIPLE (SRP):**

En una arquitectura hexagonal, la separación entre la lógica de negocio y la infraestructura ayuda a asegurar que cada componente tenga una única responsabilidad. Por ejemplo, los adaptadores de la interfaz de usuario se centran exclusivamente en la interacción con el usuario, mientras que la lógica de negocio se concentra en las reglas y operaciones del dominio. Esto cumple con el SRP al mantener las responsabilidades bien definidas y separadas.

## **OPEN/CLOSED PRINCIPLE (OCP):**

La arquitectura hexagonal permite que la aplicación sea extensible sin necesidad de modificar el código existente. Por ejemplo, si se desea añadir una nueva forma de interactuar con la aplicación (como un nuevo tipo de interfaz de usuario o un nuevo sistema de persistencia), se puede hacer implementando nuevos adaptadores para los puertos correspondientes sin modificar la lógica de negocio central ni otros adaptadores. Esto se alinea con el OCP, facilitando la extensión de la funcionalidad de la aplicación sin alterar su núcleo.

## **LISKOV SUBSTITUTION PRINCIPLE (LSP):**

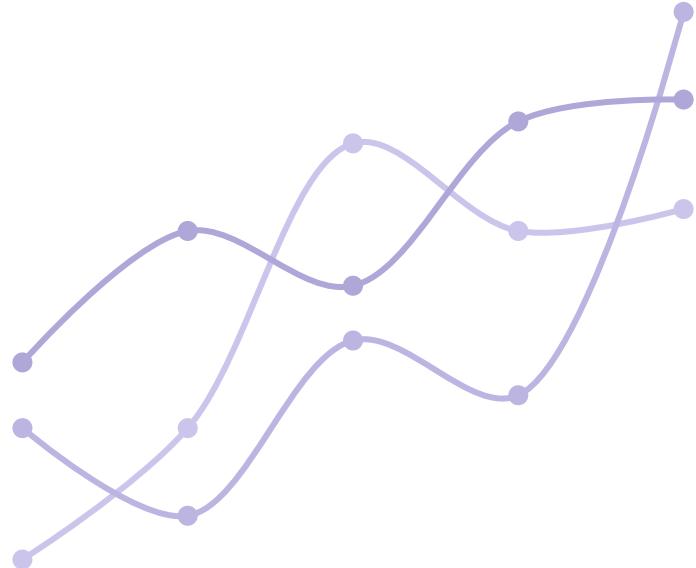
Los puertos definidos en una arquitectura hexagonal actúan como contratos entre la lógica de negocio y los adaptadores. Cualquier adaptador que cumpla con el contrato de un puerto puede ser intercambiado con otro sin afectar el funcionamiento de la aplicación. Esto significa que los adaptadores pueden ser sustituidos por sus subtipos de manera transparente, lo cual es una manifestación directa del LSP.

## **INTERFACE SEGREGATION PRINCIPLE (ISP):**

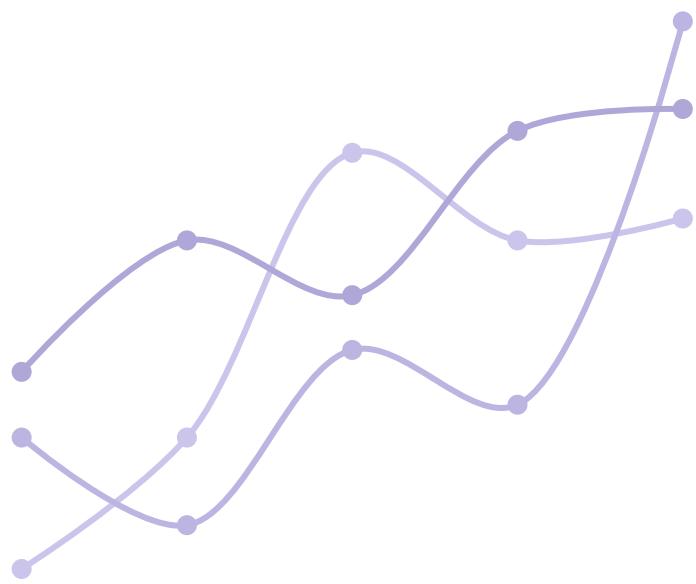
La arquitectura hexagonal fomenta la creación de interfaces específicas (puertos) que los adaptadores deben implementar. Esto significa que los componentes de la aplicación no se ven obligados a depender de interfaces que contengan métodos que no utilizan. Cada puerto define exactamente las operaciones necesarias que un adaptador debe realizar, siguiendo así el ISP.

## **DEPENDENCY INVERSION PRINCIPLE (DIP):**

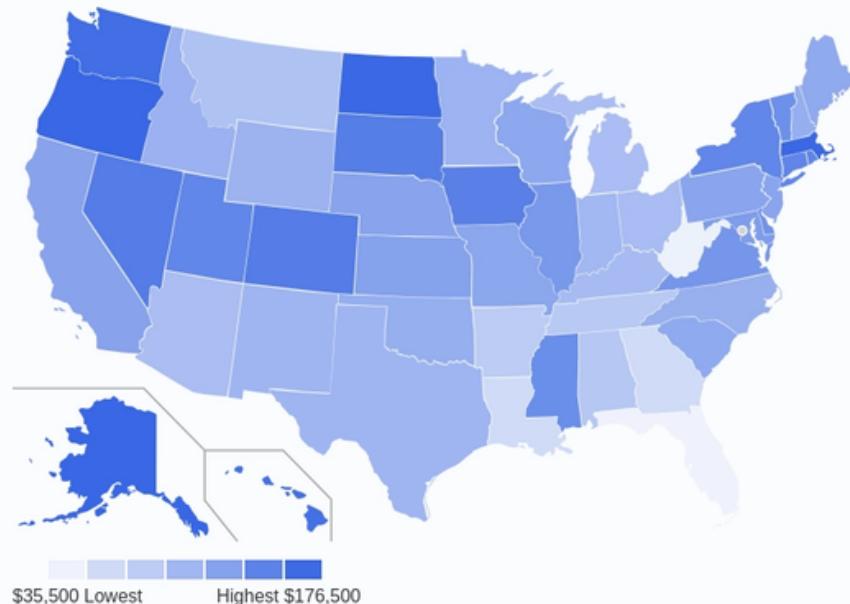
En la arquitectura hexagonal, tanto la lógica de negocio (alto nivel) como los adaptadores (bajo nivel) dependen de abstracciones (los puertos). Los adaptadores implementan estas abstracciones para interactuar con la lógica de negocio, invirtiendo la dependencia tradicional de que la lógica de negocio conozca los detalles de los mecanismos de entrada/salida o persistencia. Esto cumple con el DIP al enfocar las dependencias en las abstracciones en lugar de en los detalles concretos.



# ANÁLISIS DE MERCADO



# Flutter App Developer Salary in the United States



ZipRecruiter salary estimates, histograms, trends and comparisons are derived from both employer job postings and third party data sources.



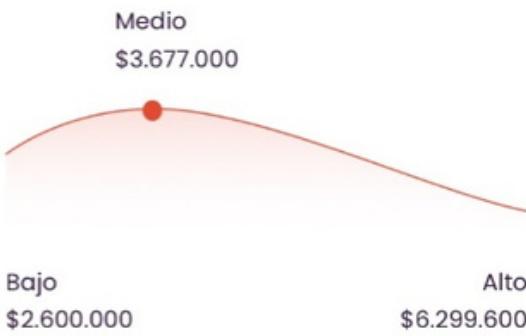
Last Updated : March 11, 2024

## ¿Cuánto gana un Desarrollador flutter en Colombia?

# \$3.677.000 / Mes

Basado en 22 salarios

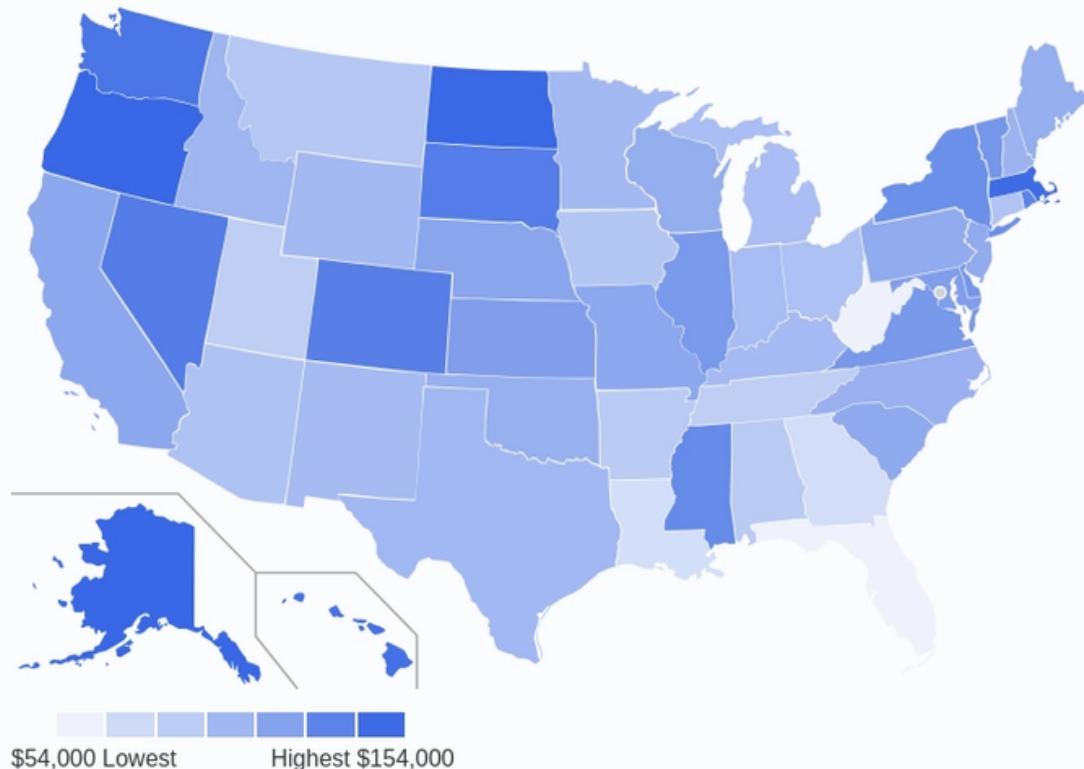
El salario **desarrollador flutter** promedio en **Colombia** es de **\$44.124.000** al año o **\$20.203** por hora. Los cargos de nivel inicial comienzan con un ingreso de **\$31.200.000** al año, mientras que profesionales más experimentados perciben hasta **\$75.595.200** al año.



# Postman Salary in the United States

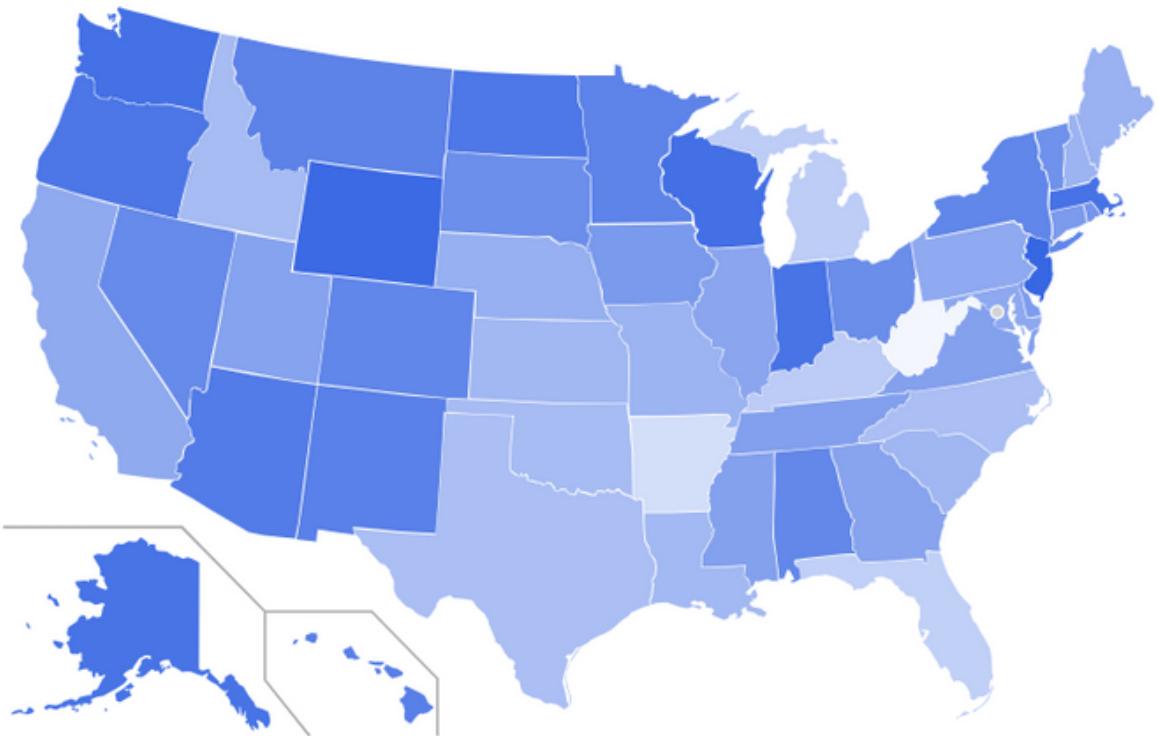


	Annual Salary	Monthly Pay	Weekly Pay	Hourly Wage
Top Earners	\$139,000	\$11,583	\$2,673	\$67
75th Percentile	\$122,500	\$10,208	\$2,355	\$59
Average	\$98,613	\$8,217	\$1,896	\$47
25th Percentile	\$75,500	\$6,291	\$1,451	\$36



ZipRecruiter salary estimates, histograms, trends and comparisons are derived from both employer job postings and third party data sources.

## What Is the Average Rest Api Developer Salary by State



	Annual Salary	Monthly Pay	Weekly Pay	Hourly Wage
Top Earners	\$153,000	\$12,750	\$2,942	\$74
75th Percentile	\$140,000	\$11,666	\$2,692	\$67
Average	\$124,567	\$10,380	\$2,395	\$60
25th Percentile	\$113,500	\$9,458	\$2,182	\$55

**\$4.000.000** / Mes

Basado en 25 salarios

Medio  
\$4.000.000



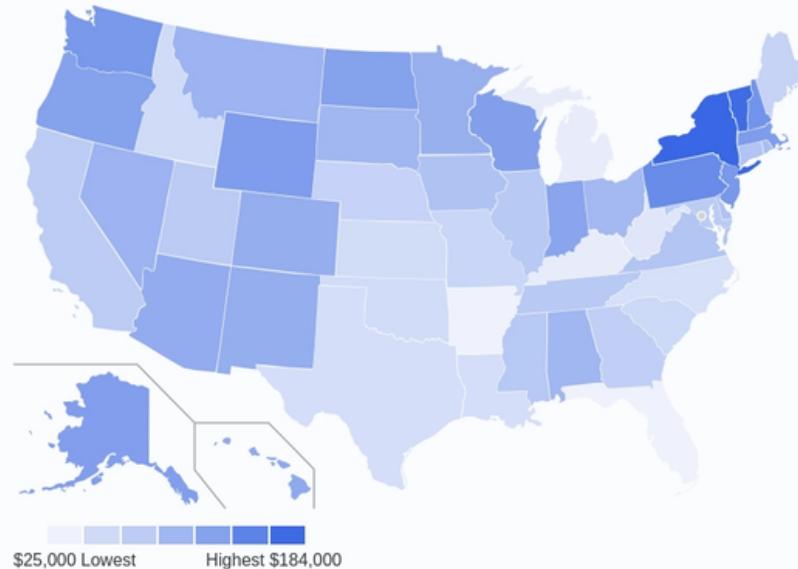
Bajo  
\$3.059.000

Alto

\$5.421.125

El salario **rest** promedio en **Colombia** es de **\$48.000.000** al año o **\$21.978** por hora. Los cargos de nivel inicial comienzan con un ingreso de **\$36.708.000** al año, mientras que profesionales más experimentados perciben hasta **\$65.053.500** al año.

# Oracle Salary in the United States



	Annual Salary	Monthly Pay	Weekly Pay	Hourly Wage
Top Earners	\$178,500	\$14,875	\$3,432	\$86
75th Percentile	\$160,000	\$13,333	\$3,076	\$77
Average	\$142,947	\$11,912	\$2,748	\$69
25th Percentile	\$127,000	\$10,583	\$2,442	\$61

ZipRecruiter salary estimates, histograms, trends and comparisons are derived from both employer job postings and third party data sources.



Last Updated : March 11, 2024

## ¿Cuánto gana un Desarrollador Oracle en Colombia?

**\$4.600.000** / Mes

Basado en 65 salarios

El salario **desarrollador oracle** promedio en Colombia es de **\$55.200.000** al año o **\$25.275** por hora. Los cargos de nivel inicial comienzan con un ingreso de **\$45.000.000** al año, mientras que profesionales más experimentados perciben hasta **\$74.292.000** al año.

Medio

\$4.600.000



Bajo

\$3.750.000

Alto

\$6.191.000