



**TI2041**

**PROGRAMACIÓN BACK END**

**Unidad 1: Tecnología del Lado del Servidor**  
**Clase 1**

# APRENDIZAJE ESPERADO

**Realiza un programa del lado del servidor, de acuerdo a la sintaxis del lenguaje.**



# MOTIVACIÓN



## Front End

- Markup and web languages such as HTML, CSS and Javascript
- Asynchronous requests and Ajax
- Specialized web editing software
- Image editing
- Accessibility
- Cross-browser issues
- Search engine optimisation



## Back End

- Programming and scripting such as Python, Ruby and/or Perl
- Server architecture
- Database administration
- Scalability
- Security
- Data transformation
- Backup

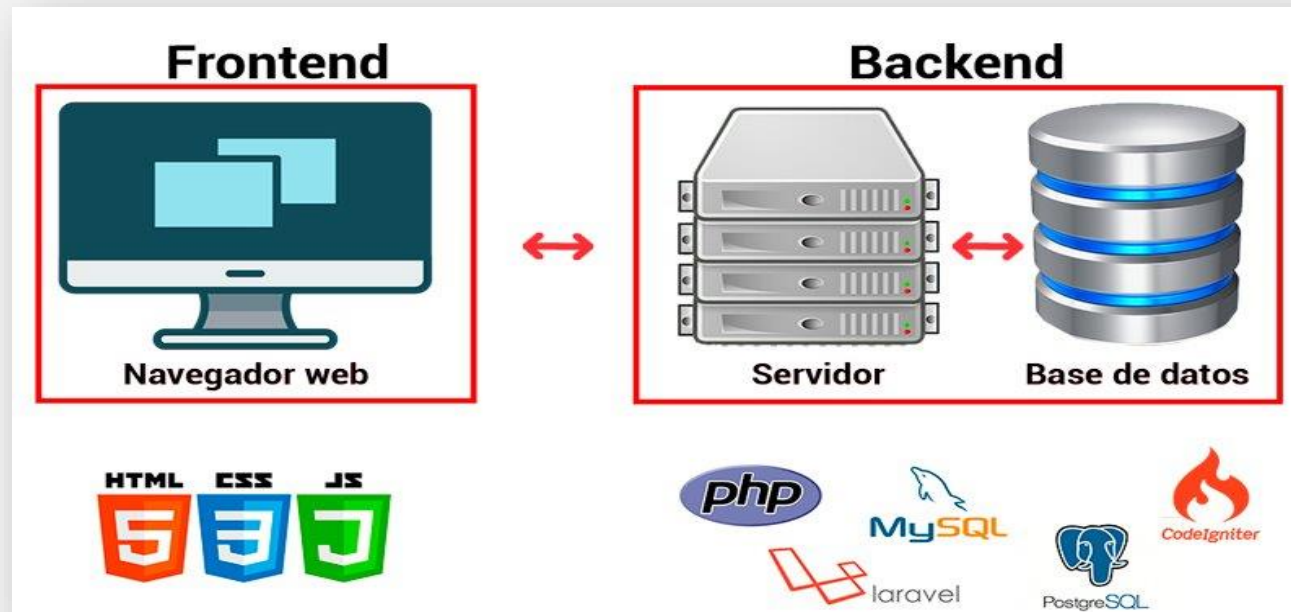
# EN ESTA CLASE

- Tecnología del Lado del Servidor:
  - Conceptos básicos
  - Arquitecturas de programación
  - Lenguajes de programación



# CONCEPTOS BÁSICOS

- Cuando hablamos de Aplicaciones Web (o Web Apps), el **Back End** se refiere a la parte lógica.
- Generalmente está relacionada con el **acceso a los datos**.





# CONCEPTOS BÁSICOS

- En el back-end, se llevan a cabo:
  - Las funciones que hagan **más simple** el proceso de desarrollo.
  - Las **acciones de lógica** de una aplicación web.
  - Las conexiones con las **bases de datos**.
  - Se usan las **bibliotecas del servidor web**, ya sea para comprimir las imágenes de la web, para implementar temas de caché u otras.
  - También se mantiene la **seguridad** de los sitios web.
  - Se pueden **optimizar los recursos** a fin de que las páginas resulten más ligeras.
  - Entre otras funciones.

# CONCEPTOS BÁSICOS

- Un **Desarrollador Back-End** debe:
  - Manejar lo básico, que es la configuración de los servidores; o por lo menos, debe conocer lo suficiente sobre este punto para poder hacer reportes al responsable de sistemas.
  - La lógica y el orden deben ser los dos pilares en los que se fundamente su trabajo.
  - Conocer sobre metodologías de la programación, sus fundamentos y manejar con soltura estos lenguajes, o por lo menos uno de ellos.
  - Práctica constante, seriedad, una alta capacidad de atención y muchas dosis de concentración.



# CONCEPTOS BÁSICOS



Front End

Back End





# ARQUITECTURA



- La **arquitectura del software** es la definición de qué **componentes** constituyen ese sistema, sus **responsabilidades**, las **relaciones de uso** y **dependencia** entre ellos.
- Es completamente **independiente de la tecnología** que se utilice y no debería representar en ningún momento el **framework**, la **base de datos** o la **forma de interactuar** con el usuario.

# ARQUITECTURA

- Existen diferentes arquitecturas de software:

Mutitier

Onion

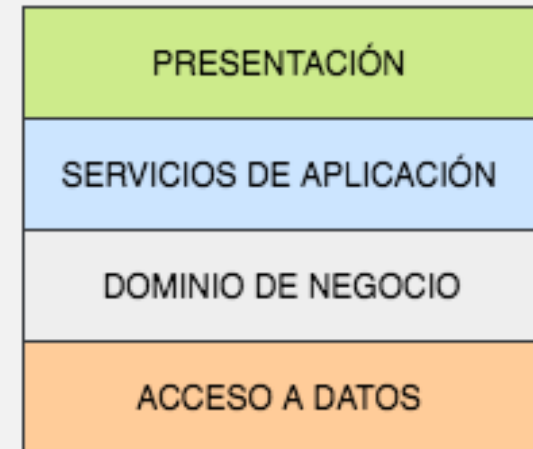
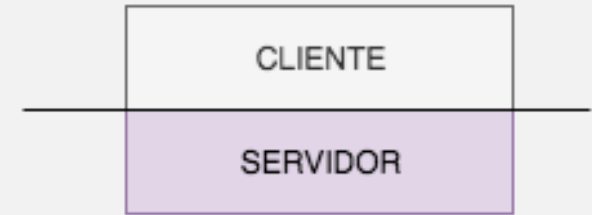
Ports and  
Adapters

Clean  
Architecture

Entre  
otras...

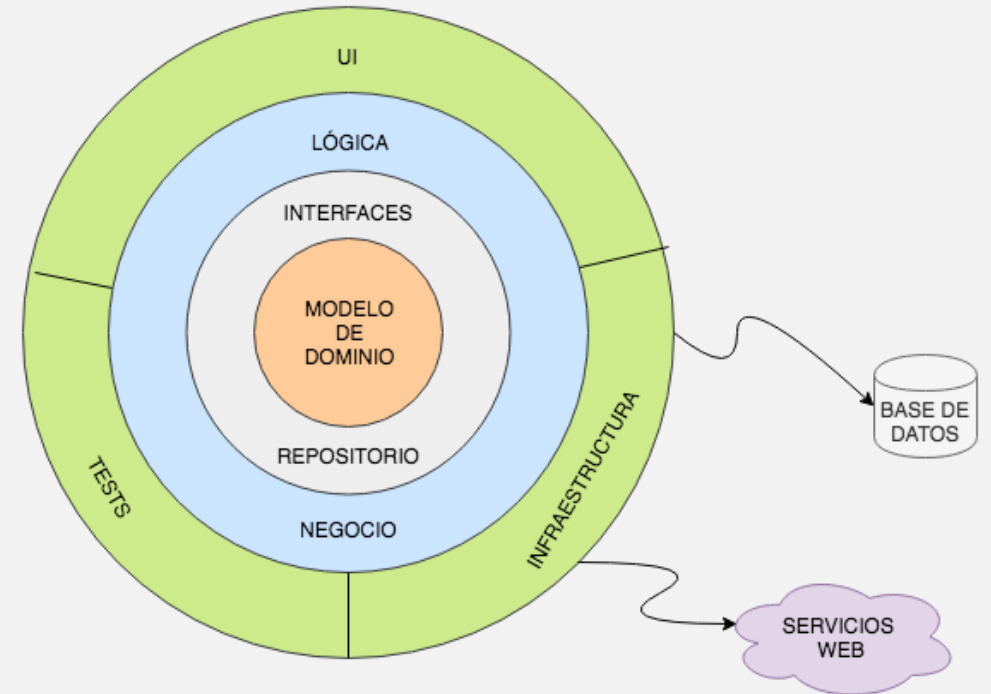
# ARQUITECTURA

- **Multicapas** (*Multitier*) es una arquitectura en la que se expone la separación de sistema en varias capas físicas.
- Es decir, los distintos componentes están en máquinas separadas.



# ARQUITECTURA

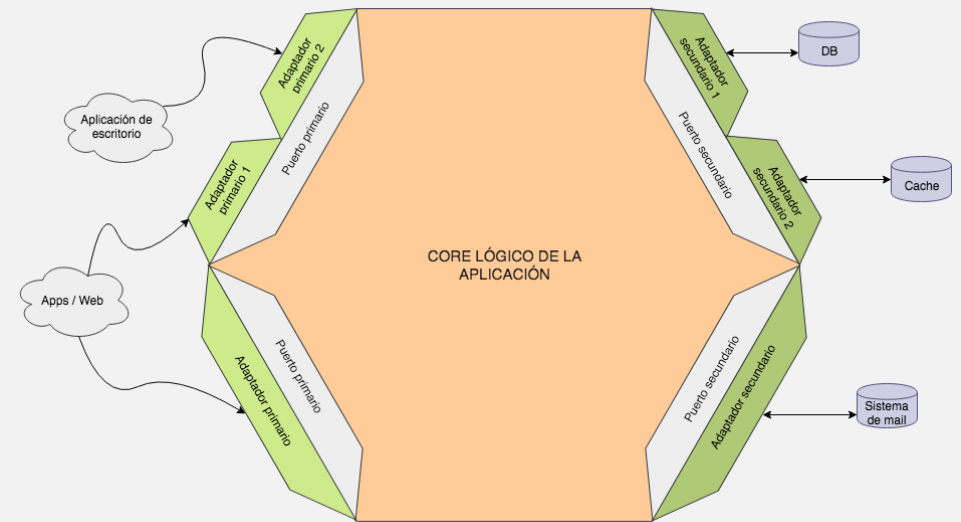
- **Cebolla** (*Onion*) es una arquitectura multicapa construida en torno a un modelo de dominio independiente de todo lo demás.
- Las dependencias van hacia el centro, por lo que todo depende de ese modelo de dominio.





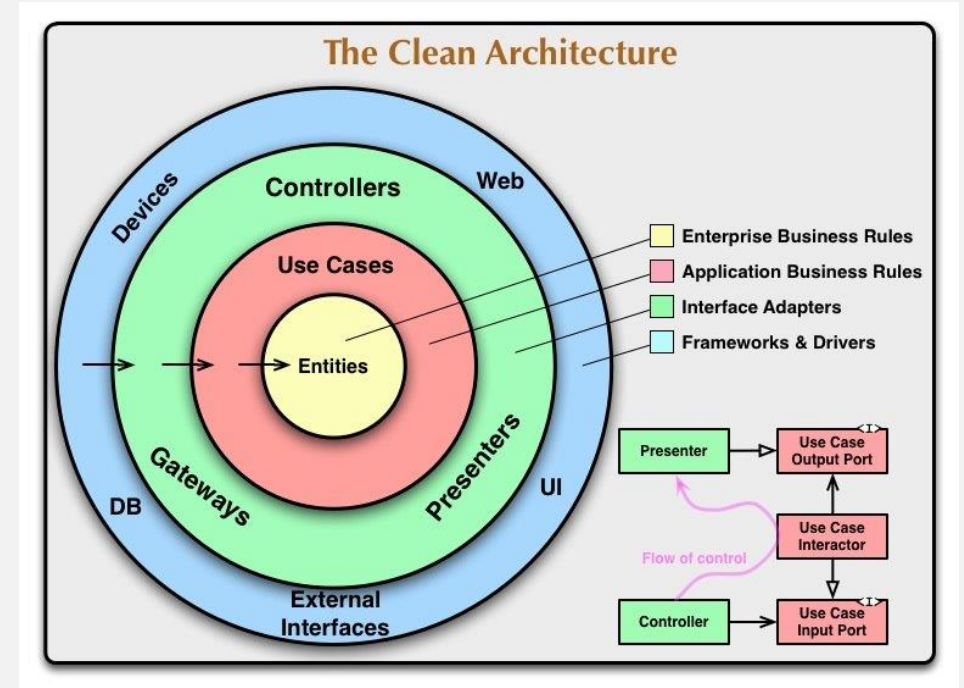
# ARQUITECTURA

- **Puertos y Adaptadores** (*Ports and Adapters*) es una arquitectura hexagonal es poner, una vez más, en el centro del sistema toda la lógica propia del dominio y definir unas fronteras muy claras y unos mecanismos de transformación con el exterior.



# ARQUITECTURA

- **Arquitectura Limpia** (*Clean Architecture*) es una arquitectura que intenta reunir lo mejor de las anteriores, creando una arquitectura de capas, dependientes de un dominio y siempre hacia el centro que es el core de la aplicación.



# ARQUITECTURA

- La mayoría de las arquitectura consideran la **capa de presentación** de alguna manera.
  - En el caso de multitier, es la capa superior (o el cliente).
  - En el caso de onion, está en la capa exterior de la cebolla.
  - En el caso de la hexagonal, está asociada a algunos puertos.
  - En el caso de clean, también está en la capa exterior, relacionada con los presenters.
- Considerando que las **aplicaciones web** tienen su capa de presentación en el front end ¿cómo esto afecta a la **arquitectura del back-end**?

# ARQUITECTURA

- Para el caso exclusivo de las arquitecturas para Back-end se pueden separar en 3 principales arquitecturas:

Capas

Event-Driven

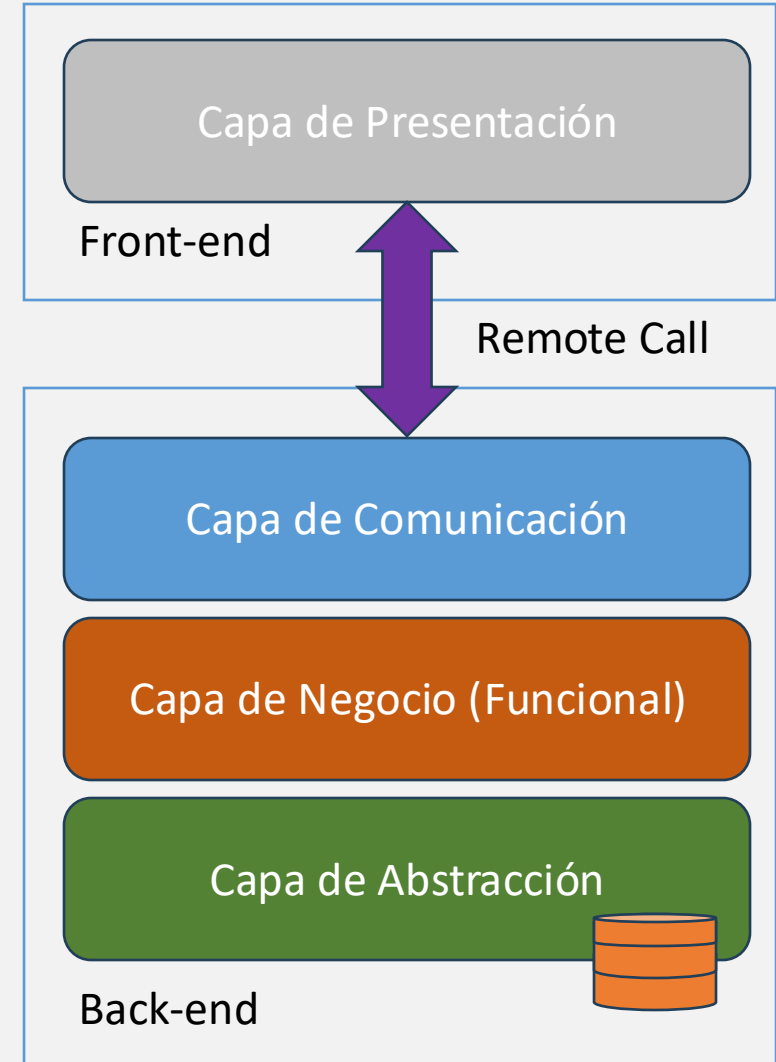
Microservices



# ARQUITECTURA

## • Arquitectura de Capas

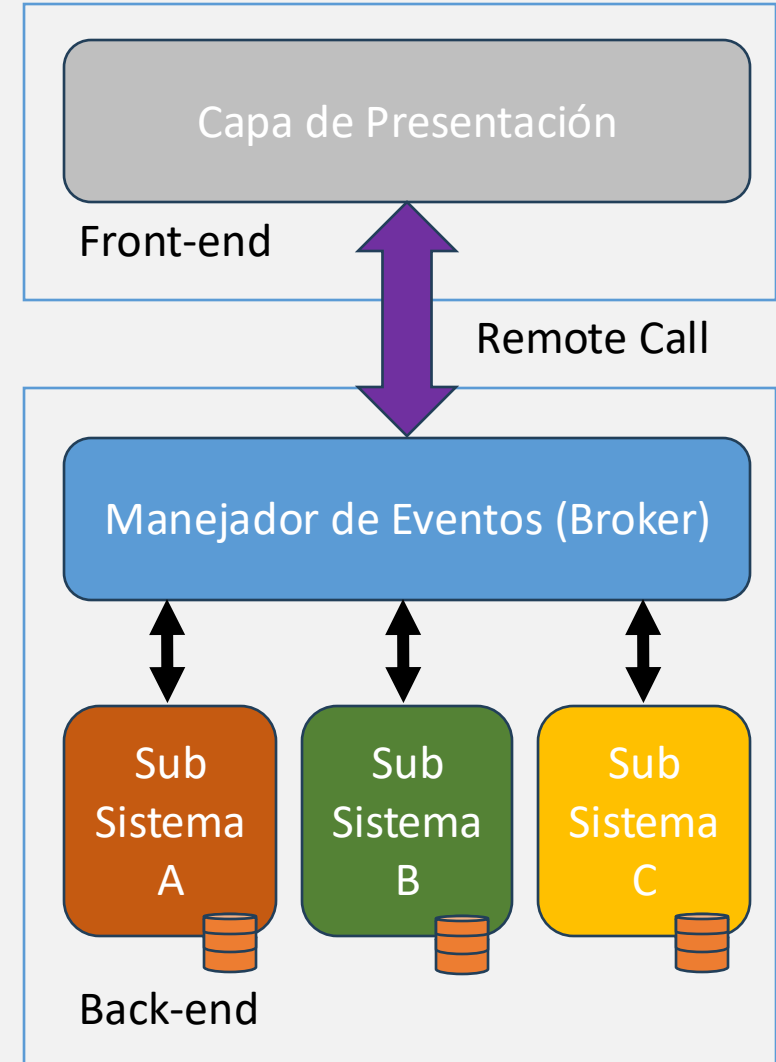
- Similar al multitier, pero dejando la capa de interfaz de usuario fuera.
- Se considera en cambio una capa de comunicación entre el back-end y front-end.
- Ventajas:
  - Cada capa es independiente entre sí.
  - El back-end se convierte en un servicio escalable.
  - Permite hacer cambios a diferentes niveles sin afectar a otros.



# ARQUITECTURA

## • Arquitectura Dirigida por Eventos

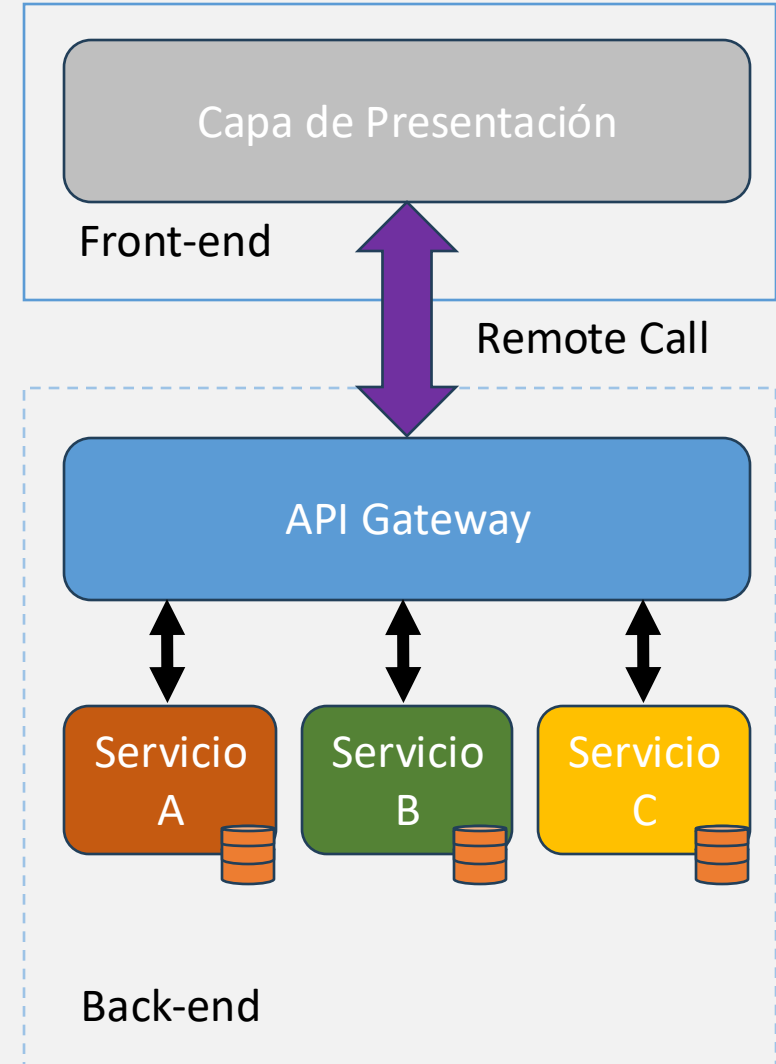
- Más parecido a la arquitectura hexagonal, ya que responde a las peticiones de los usuarios por diferentes eventos.
- Luego, el control del evento debe ser delegado a un área de procesamiento, la cual realiza la función del negocio.
- Ventajas:
  - Los eventos son independientes entre sí, lo cual los convierte en flujos aislados dentro del back-end.
  - Por lo tanto, también es un servicio escalable.
  - Cada unidad permite cambios sin afectar a otras.



# ARQUITECTURA

## • Arquitectura de Microservicios

- Separación máxima de un flujo de procesamiento en pequeñas unidades que son capaces de operar en forma aislada.
- No requieren el uso de otros programas, porque son autocontenidos.
- Ventajas:
  - Los microservicios son altamente cohesionados y son capaces de operar en forma independiente.
  - Por lo tanto, la escalabilidad se habilita a través de mayor cantidad de microservicios.
  - Además, los microservicios pueden estar distribuidos, porque incluso la arquitectura base no tiene relación.



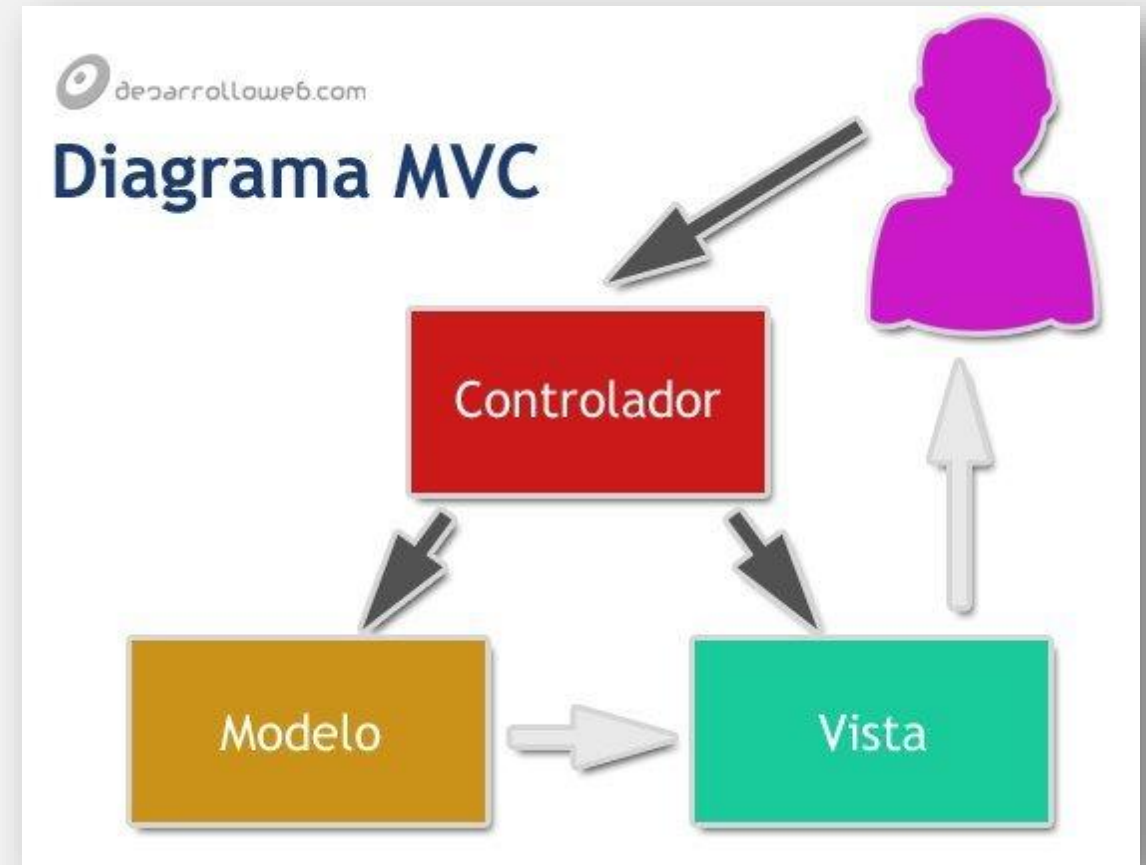
# ARQUITECTURA

- Una de las propuestas más comunes de arquitectura de capas para back-end es el patrón **Modelo-Vista-Controlador** (o MVC).
- Es utilizada para **separar el código** por sus distintas responsabilidades, manteniendo distintas capas que se encargan de hacer una tarea muy concreta, lo que ofrece beneficios diversos.
- MVC ya tiene varias décadas, sin embargo, ha ganado mucha fuerza y seguidores gracias a la aparición de **frameworks de desarrollo web** que lo utilizan como modelo para la arquitectura de las aplicaciones web.



# ARQUITECTURA

- Si bien es cierto, en la capa de Vista se puede suponer que contiene la **interfaz de usuario**, en el caso de Back-ends solo corresponde a la **transferencia de la información** que el navegador debe interpretar.
- De esta manera, **el navegador es el intérprete** de lo que el modelo envía como Vista para la presentación final ante el usuario.
- Más adelante veremos algo más de detalle cuando comencemos a programar el back-end.



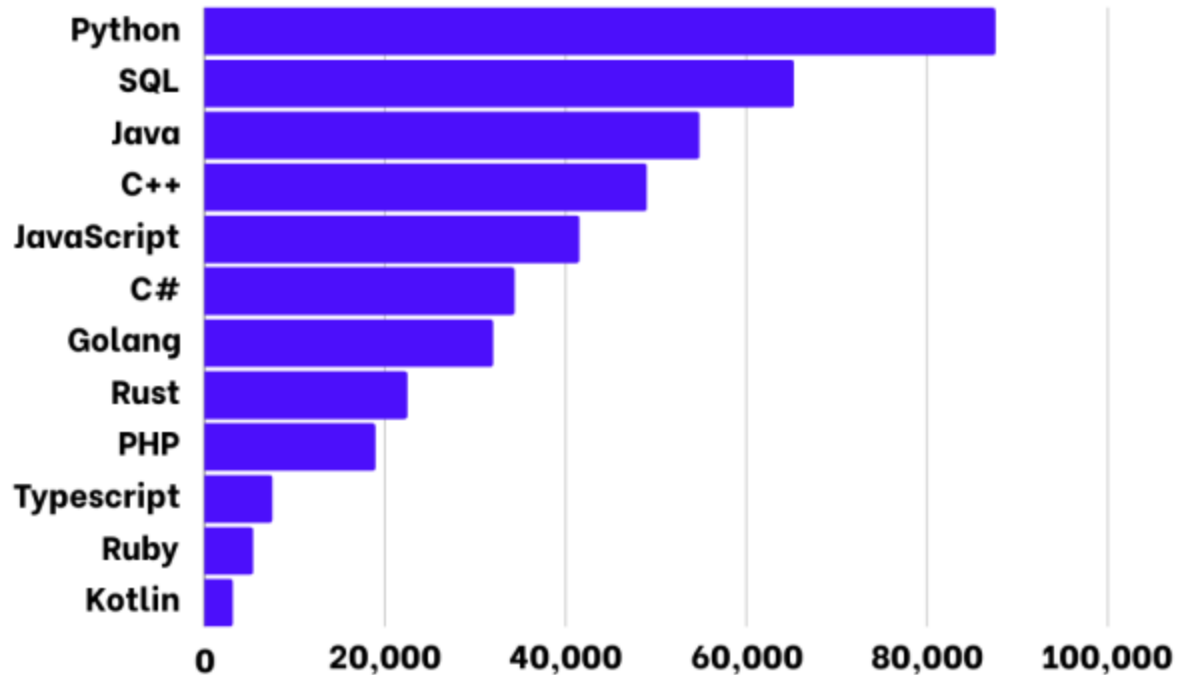
# LENGUAJES DE PROGRAMACIÓN

- El back-end de una aplicación web debe ser construido en un **lenguaje apropiado para la interacción web**.
- A diferencia del front-end, en donde utilizamos HTML, CSS y JS, el back-end requiere un lenguaje que tenga un potencial de **uso de los recursos que el servidor posee**, incluyendo el acceso a bases de datos, principal recurso utilizado por las aplicaciones web.
- A continuación vemos un gráfico con los lenguajes **más populares** para el desarrollo de back-ends.

# LENGUAJES DE PROGRAMACIÓN

## 2024 Programming Language Breakdown

# of open job postings for each language, in the US only



*Lenguajes de programación con cargos solicitados en estados unidos durante el año 2024.*

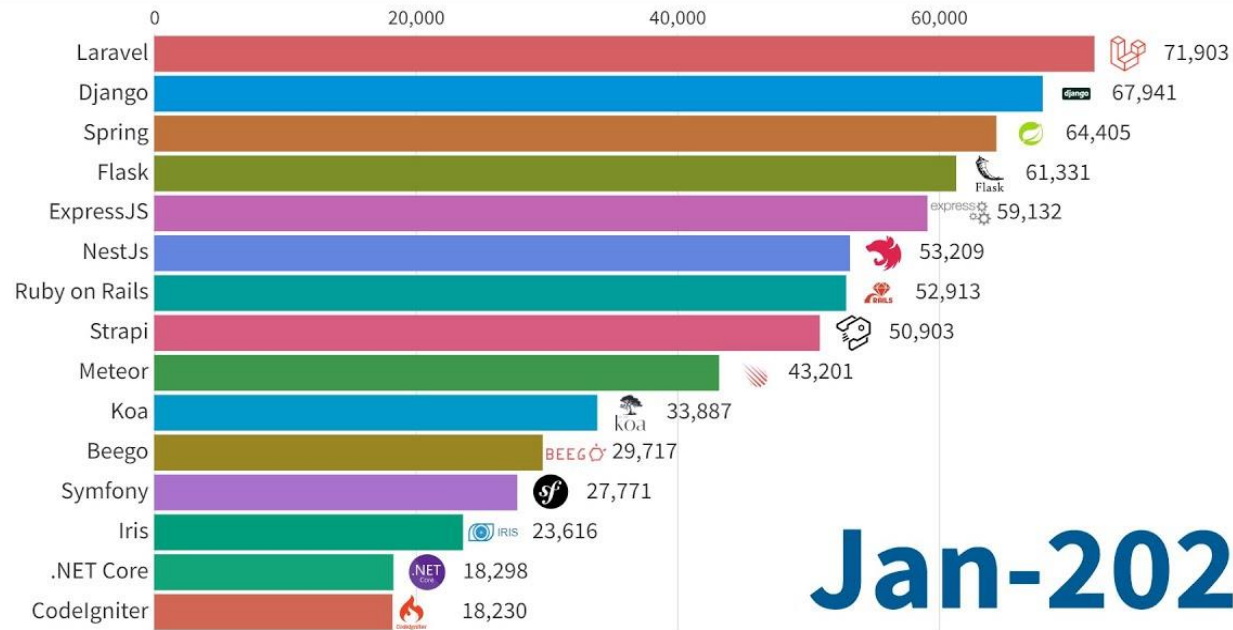
# LENGUAJES DE PROGRAMACIÓN

- Un **Framework de Desarrollo** es, tal como se define, un **conjunto de bibliotecas que facilitan el desarrollo de una unidad de programa**.
- Existen frameworks para la mayoría de los lenguajes conocidos, es porque cada vez **se requiere obtener resultados más rápido**.
- De esta manera, existen muchos frameworks útiles en el mercado que podemos utilizar.



# LENGUAJES DE PROGRAMACIÓN

## Most Popular Backend Frameworks



Jan-2023

*Frameworks de desarrollo para back-end más populares a enero del 2023.*

# EN RESUMEN

- El **Desarrollo Back-End** es la disciplina que permite el desarrollo de programas en el lado del servidor en una arquitectura web.
- Permite el **procesamiento de negocio** y hoy en día, la capa back-end funciona independiente al front-end con **servicios web**.
- La arquitectura de desarrollo que usaremos para nuestro back-end es **MVC** lo que nos permite separar tanto el modelo de negocio y datos de la interacción con el front-end (vista y controlador).
- El lenguaje de desarrollo para este curso será **Python**.



UNIVERSIDAD TECNOLÓGICA DE CHILE  
INSTITUTO PROFESIONAL  
CENTRO DE FORMACIÓN TÉCNICA