

A woman with blonde hair is wearing a black VR headset. The background is dark with large, faint, stylized numbers (0, 5, 1, 2, 3, 4) and a large purple-to-pink gradient arc at the top. A horizontal purple-to-pink gradient bar is at the bottom. The text 'Back-End+' is written in a large, white, sans-serif font, with the '+' sign in a purple-to-pink gradient. The text is centered and partially overlaps the woman's face and the VR headset.

# Back-End+



# Programação Orientada a Objetos

## Fundamentos Técnicos e Científicos

Lógica booleana, árvore de decisão, estruturas de laço, conceito de recursividade, estrutura de dados (vetor, matriz, fila, pilha...), variável e constante e sistema de controle de versão

x x x x

# Padrão MVC (Model-View-Controller)

O que é o Padrão MVC?

O Padrão MVC é uma abordagem arquitetural que organiza o código de uma aplicação em três componentes principais: Model, View e Controller. Cada componente tem uma responsabilidade específica na estrutura da aplicação.

---

x x x x

x x x x

# Padrão MVC (Model-View-Controller)

O que é o Padrão MVC?

O Padrão MVC é uma abordagem arquitetural que organiza o código de uma aplicação em três componentes principais: Model, View e Controller. Cada componente tem uma responsabilidade específica na estrutura da aplicação.

---

x x x x

x x x x

# Padrão MVC (Model-View-Controller)

Conceito:

Camada de interação do usuário (View)

Camada de manipulação (model)

Camada de controle (controller)

---

x x x x



x x x x

# Padrão MVC (Model-View-Controller)

## Model:

- Representa a camada de dados e lógica de negócios.
- Responsável pelo acesso e manipulação dos dados.
- Não possui conhecimento sobre a interface do usuário.

## View:

- Apresenta os dados ao usuário e lida com a interação do usuário.
- Não possui lógica de negócios; apenas exibe informações.

---

x x x x

x x x x

# Padrão MVC (Model-View-Controller)

## Controller:

- Recebe entradas do usuário e atualiza o Model ou a View conforme necessário.
- Gerencia o fluxo de dados entre o Model e a View.

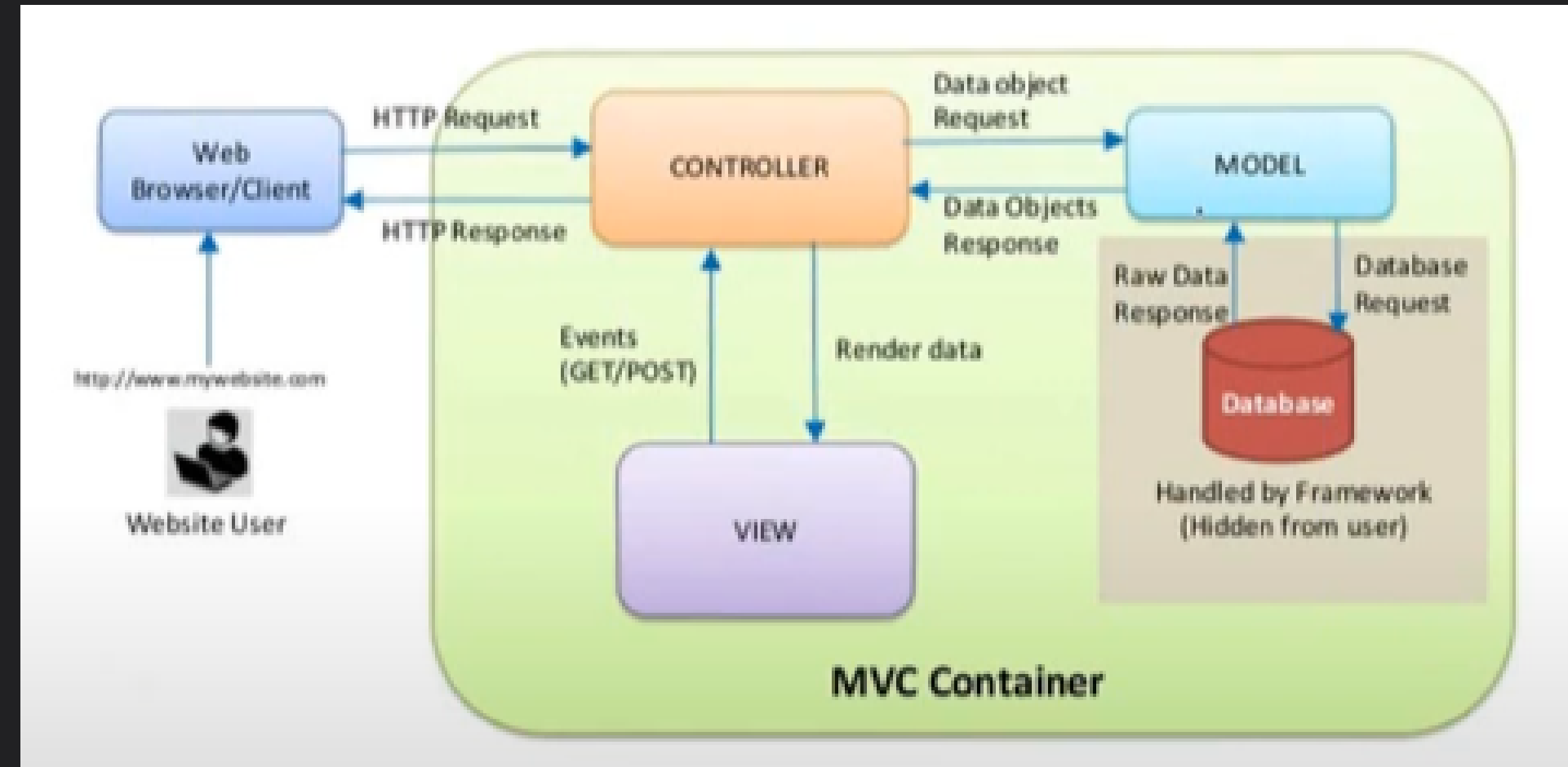
---

x x x x

x x x x

# Padrão MVC (Model-View-Controller)

Exemplo:



x x x x



x x x x

# Padrão MVC (Model-View-Controller)

Vantagens do Padrão MVC:

- **Separação de Responsabilidades:** Cada componente tem uma função específica, o que facilita a manutenção e escalabilidade do código.
- **Reutilização de Código:** A separação de Model, View e Controller permite a reutilização de componentes em diferentes partes da aplicação.

---

x x x x

x x x x

# Classes e Objetos

O que são Classes, Objetos e Método?

- **Classe:**
  - Uma estrutura que define o comportamento e as propriedades de um objeto.
  - Serve como um modelo para criar objetos.
- **Objeto:**
  - Uma instância de uma classe.
  - Possui propriedades (atributos) e comportamentos (métodos).

---

x x x x

x x x x

# Classes e Objetos

## Método:

- Função associada a uma classe.
- Representa o comportamento do objeto.
- Pode modificar o estado do objeto ou realizar operações específicas.

---

x x x x

x x x x

# Herança e Polimorfismo

## Herança:

- Mecanismo que permite criar uma nova classe baseada em uma classe existente.
- A nova classe herda atributos e métodos da classe existente (classe base ou superclasse).

## Polimorfismo:

- Habilidade de uma classe derivada de uma superclasse para fornecer uma implementação específica de um método que já está definido na superclasse.
- Permite que objetos de diferentes classes sejam tratados de maneira uniforme.

---

x x x x

x x x x

# Classes Abstratas e Métodos Abstratos

## Classe Abstrata:

- Uma classe que não pode ser instanciada e geralmente contém métodos abstratos.

## Método Abstrato:

- Um método que é declarado na classe, mas não possui implementação.
- Deve ser implementado nas subclasses derivadas.

---

x x x x

x x x x

# Interface

- Define um conjunto de métodos que uma classe deve implementar.
- Uma classe pode implementar várias interfaces.

---

x x x x