

Design Patterns

O que é Design Patterns ?

Design Patterns, ou padrões de projeto, são soluções provadas e reutilizáveis para problemas recorrentes no desenvolvimento de software. São como plantas de construção, que você pode personalizar para resolver problemas específicos no seu código. Eles não são código pronto, mas sim um conceito geral que fornece uma estrutura para abordar problemas comuns.

- **Soluções Reutilizáveis:** Design Patterns oferecem soluções testadas e comprovadas para problemas comuns em software.
- **Plantas de Construção:** São como plantas de construção que você pode adaptar para a sua solução.
- **Não são Código Pronto:** Não são blocos de código que você simplesmente copia e cola.
- **Conceito Geral:** Fornecem uma estrutura de pensamento para resolver problemas específicos.

O que é MVC ?

(Model-View-Controller) em Linguagens de
Programação

MVC é um padrão arquitetural amplamente utilizado no desenvolvimento de software, especialmente em aplicações web. Ele separa a aplicação em três componentes principais interconectados:

Model (Modelo)

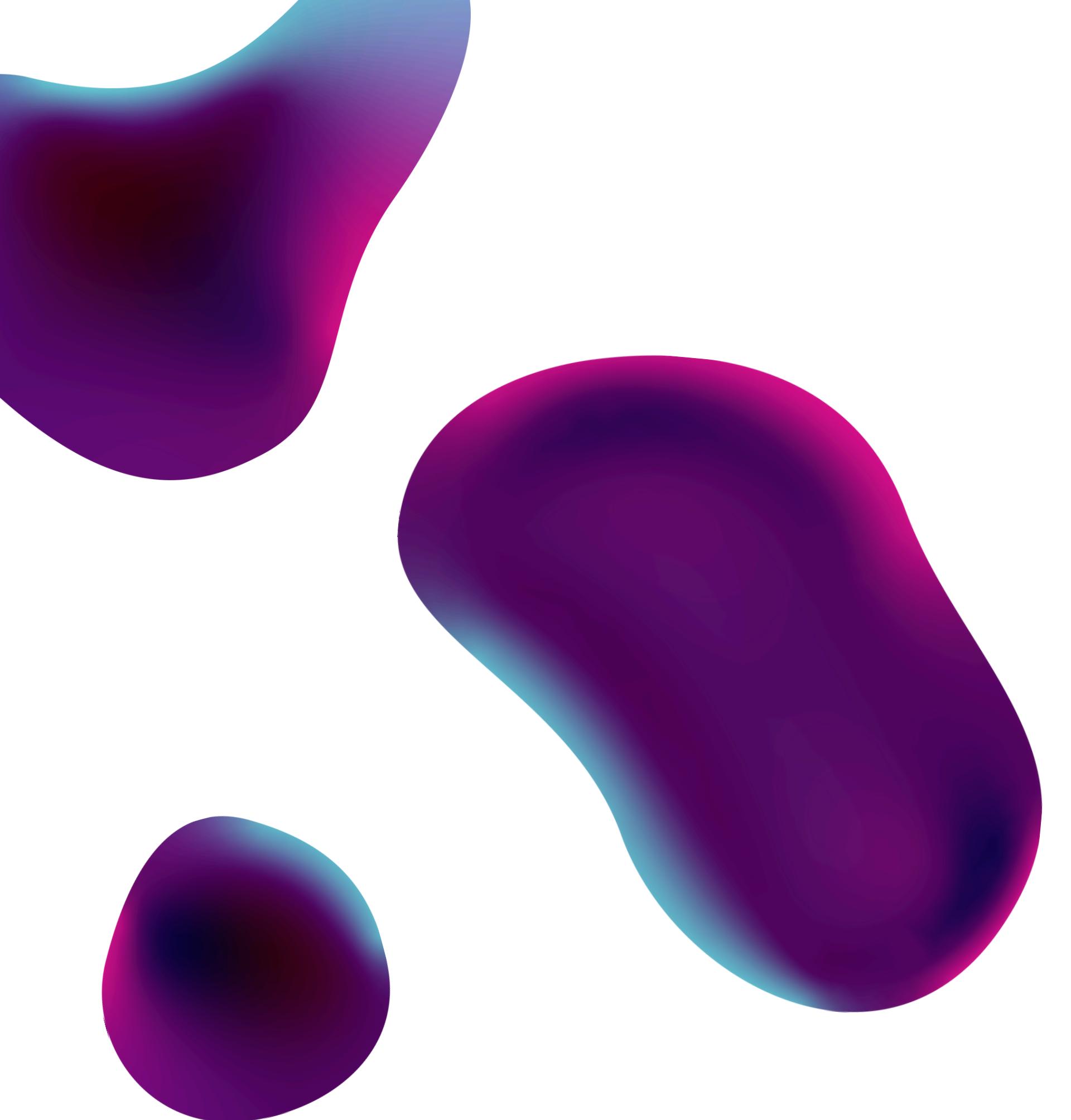
- Representa os dados e a lógica de negócios
- Responsável por acessar e manipular dados (banco de dados, APIs, etc.)
- Não conhece a View ou o Controller

View (Visualização)

- Representa a interface do usuário (UI)
- Exibe os dados do Model para o usuário
- Geralmente composta por templates/arquivos de apresentação

Controller (Controlador)

- Recebe as entradas do usuário (requisições)
- Coordena a interação entre Model e View
- Processa dados e decide qual View mostrar



Como o MVC Funciona
?

Como o MVC Funciona

1. O usuário interage com a interface (View)
2. A View envia ações/eventos para o Controller
3. O Controller processa a entrada e interage com o Model
4. O Model atualiza seus dados e notifica o Controller
5. O Controller seleciona a View apropriada
6. A View obtém dados atualizados do Model e renderiza a interface

Benefícios do MVC

Benefícios do MVC

- Separação de conceitos: Cada componente tem uma responsabilidade clara
 - Reutilização de código: Models e Views podem ser reutilizados
 - Manutenibilidade mais fácil: Mudanças em um componente afetam minimamente os outros
 - Desenvolvimento paralelo: Diferentes equipes podem trabalhar em componentes distintos
- Testabilidade: Componentes isolados são mais fáceis de testar

Variações do MVC

- **MVVM (Model-View-ViewModel):** Usado em frameworks como Angular e Knockout.js
- **MVP (Model-View-Presenter):** Versão mais testável do MVC
- **HMVC (Hierarchical MVC):** Para aplicações mais complexas

o que são ORMs ?

o que são ORMs ?

ORM significa Object-Relational Mapping (ou Mapeamento Objeto-Relacional, em português). É uma técnica usada na programação para facilitar a interação entre linguagens de programação orientadas a objetos (como Python, Java, C#) e bancos de dados relacionais (como MySQL, PostgreSQL, SQLite, etc.).

O que um ORM faz?

Em vez de escrever comandos SQL diretamente para manipular dados no banco, com um ORM você trabalha com objetos da linguagem. O ORM faz a "ponte" entre o modelo de dados em objetos e as tabelas do banco de dados.

Frameworks Modernos

Nesta seção, vamos explorar diferentes frameworks modernos, como Laravel, Django, FastAPI, Microsoft e NextJS, suas arquiteturas e os casos de uso mais eficazes.

Nesta seção, vamos explorar diferentes frameworks modernos, como Laravel, Django, FastAPI, Microsoft e NextJS, suas arquiteturas e os casos de uso mais eficazes.



Laravel

Laravel é o framework PHP mais popular, adotando o padrão MVC com foco em produtividade, elegância e uma sintaxe expressiva. Ele oferece recursos completos como ORM (Eloquent), autenticação, filas, jobs, cache e Blade (sistema de templates). Ideal para aplicações web robustas com backend tradicional.

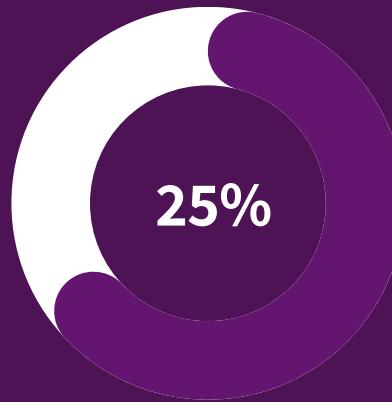
Django (Python) também usa um padrão semelhante (MTV) e foca na rapidez de desenvolvimento com segurança. Sua estrutura é mais rígida e menos “mágica” que Laravel, mas muito confiável para aplicações robustas.

FastAPI (Python) é minimalista e extremamente rápido, focado exclusivamente em APIs REST. Não possui um sistema de templates nativo como o Blade, mas é ideal para microsserviços e backends modernos.

Next.js (JavaScript) é um framework full-stack baseado em React. Ao contrário do Laravel, foca no front-end com suporte a rotas API no backend, ideal para aplicações web modernas (SPA/SSR).



DESEMPENHO E POPULARIDADE



FRAMEWORK DE ALTA POPULARIDADE

Django é amplamente usado na indústria, oferecendo robustez, recursos e uma comunidade ativa, ideal para projetos complexos.



POPULAR ENTRE FRONT-END

NextJS é o preferido de desenvolvedores React, facilitando a criação de interfaces ricas e responsivas com SEO otimizado.



UTILIZAÇÃO CONSTANTE

Laravel mantém uma base sólida de usuários, essencial em muitos servidores e aplicativos legados, garantindo suporte contínuo.



EMERGENTE

FastAPI está rapidamente conquistando o mercado por sua eficiência e suporte a tipos, sendo excelente para aplicações modernas.



Comparação Técnica

Esta tabela resume as diferenças técnicas entre vários motores de templates populares, focando em suas linguagens, tipos e recursos principais.

MOTOR	TIPO	LINGUAGEM DE PROGRAMAÇÃO	PRINCIPAIS RECURSOS
Jinja2	Servidor	Python	Fácil de usar, extensível
Django	Servidor	Python	Framework completo, prontos para uso
Vue	Cliente	JavaScript	Reativo, simples, ótimo para SPAs
React	Cliente	JavaScript	Componentes reutilizáveis, rápido
Angular	Cliente	JavaScript	Framework robusto, ideal para apps complexas
Laravel	Servidor	PHP	MVC, fácil integração com bancos de dados

Conclusão e Recomendações

- Jinja2 é ideal para projetos em Python.
- Django fornece uma solução completa para apps web.
- React e Vue são ótimas escolhas para SPA.
- Angular é adequado para projetos complexos.
- Laravel combina poder e simplicidade com PHP.