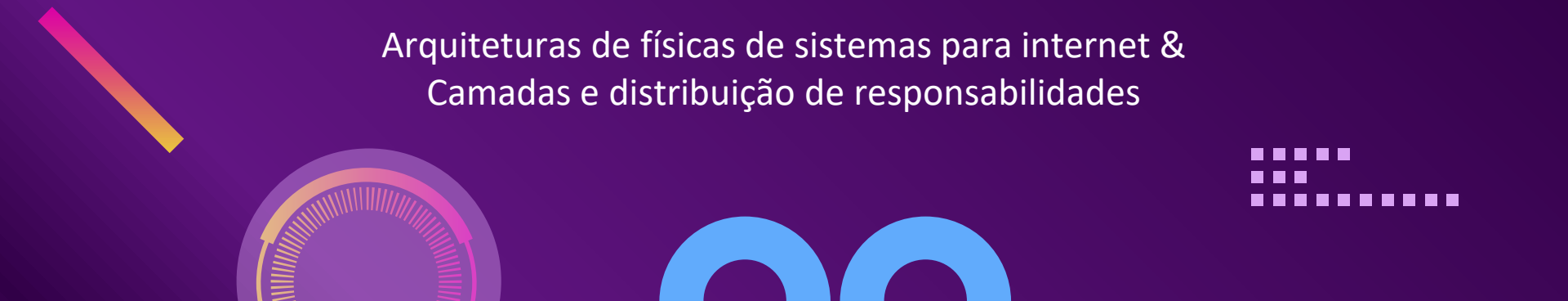




# Ciência da Computação

## Projeto de Sistemas para WEB

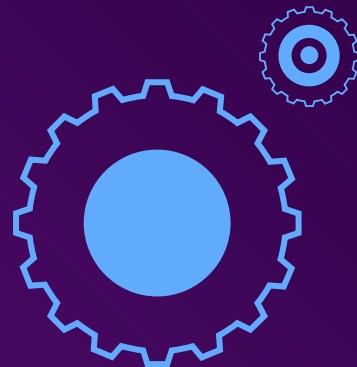


Arquiteturas de físicas de sistemas para internet &  
Camadas e distribuição de responsabilidades



# Objetivo da aula

Compreender definição de Cloud Computing e Arquitetura de Aplicações.



# Programação da aula



## Teoria

08h00 ~ 09h40



## Prática

13h00 ~ 14h40



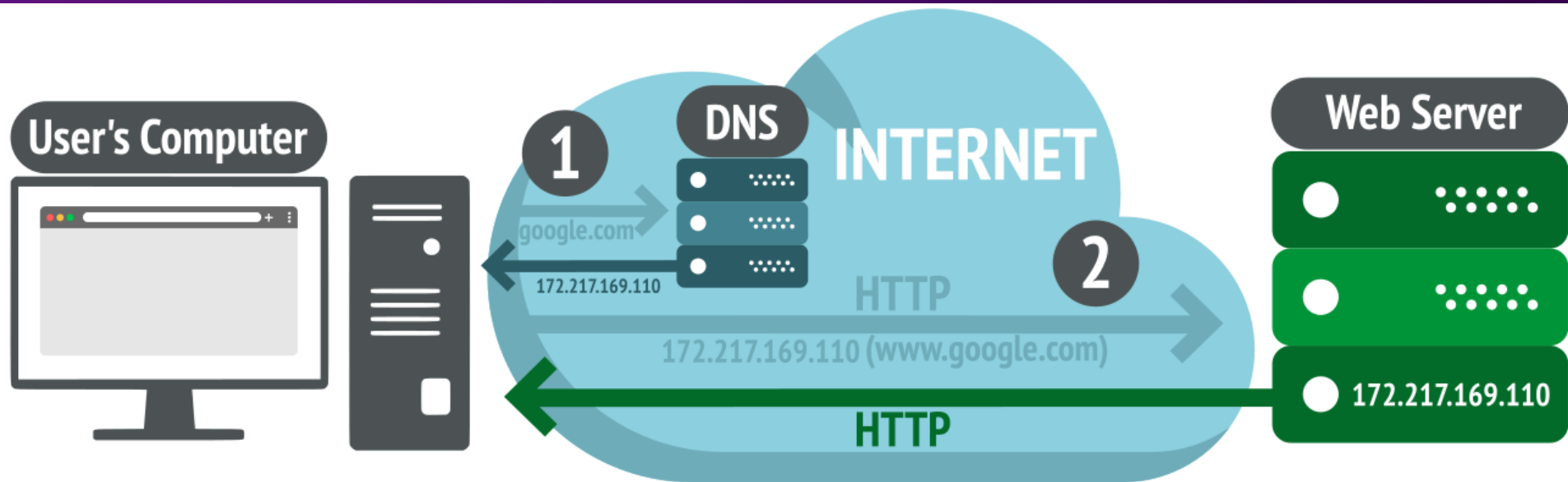
## Hoje é 14 de abril





# Cloud Computing?

SaaS, PaaS, IaaS, XaaS...





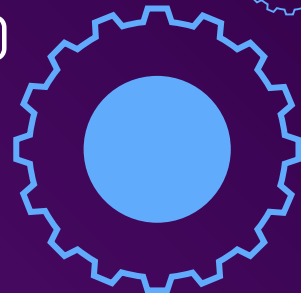
# Cloud computing



"Os gastos mundiais de usuários finais com **serviços de nuvem pública** devem crescer 20,4% em 2022, chegando a uma receita total de US\$ 494,7 bilhões. O valor é acima do registrado em 2021 (que foi de US\$ 410,9 bilhões), e abre caminho para os US\$ 600 bilhões em 2023."



<https://itforum.com.br/noticias/usuarios-de-nuven-publica-gastarao-quase-us-500-bi-em-2022/>





# Cloud computing

“É o que preveem os analistas do **Gartner** em um estudo divulgado essa semana. A pesquisa indica que o segmento de infraestrutura como serviço (**IaaS**) deve ter o maior crescimento em 2022, com alta de 30,6% nos gastos.

<https://itforum.com.br/noticias/usuarios-de-nuvem-publica-gastarao-quase-us-500-bi-em-2022/>

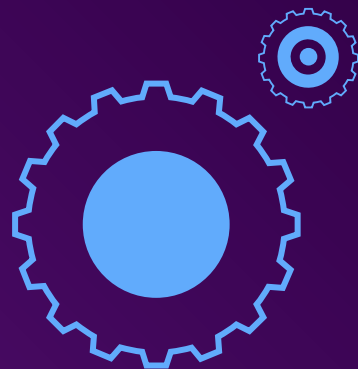




# Cloud computing

"Na sequência aparecem desktop como serviço (**DaaS**), com 26,6% de crescimento, e plataforma como serviço (**PaaS**), que deve chegar a 26,1%."

<https://itforum.com.br/noticias/usuarios-de-nuvem-publica-gastarao-quase-us-500-bi-em-2022/>

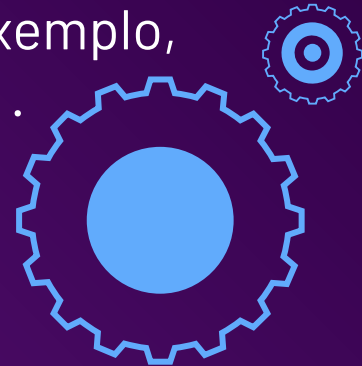




# Cloud computing?

A computação em nuvem é um modelo computacional que tem como finalidade permitir que as pessoas acessem de qualquer lugar (via internet), convenientemente e sob demanda, a um conjunto de recursos de computação configuráveis (por exemplo, redes, servidores, armazenamento, aplicativos e serviços).

<https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>



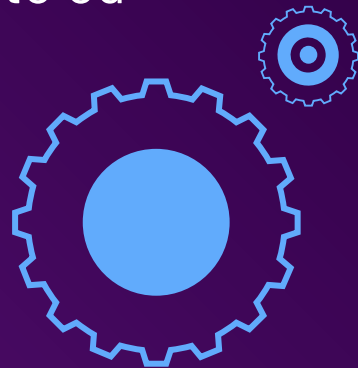




# Cloud computing?

Uma vez compartilhados na rede, estes recursos computacionais podem ser provisionados e liberados rapidamente com o mínimo de esforço de gerenciamento ou interação com o provedor de serviços.

<https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>





# Cloud computing

## Essential Characteristics

**On-demand self-service:** O consumidor pode provisionar unilateralmente recursos de computação, como tempo de servidor e armazenamento em rede, conforme sua necessidade, automaticamente, sem exigir interação humana com cada provedor de serviços.

<https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>





# Cloud computing

## Essential Characteristics

**Broad network access:** Os recursos estão disponíveis na rede e são acessados por meio de mecanismos padrão que promovem o uso por plataformas heterogêneas de Thin ou Thick Client (telefones celulares, tablets, laptops, wearables e estações de trabalho).

<https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>



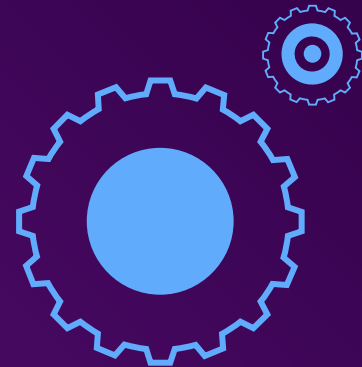


# Cloud computing

## Essential Characteristics

**Resource pooling:** os recursos de computação do provedor são agrupados para atender a vários consumidores usando um modelo multilocatário, com diferentes recursos físicos e virtuais atribuídos e reatribuídos dinamicamente de acordo com a demanda do consumidor.

<https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>





# Cloud computing

## Essential Characteristics

Há um senso de independência de localização, pois o cliente geralmente não tem controle ou conhecimento sobre a localização exata dos recursos computacionais fornecidos; Exemplos de recursos incluem armazenamento, processamento, memória e largura de banda de rede.

<https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>



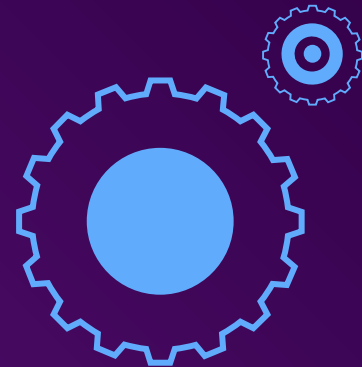


# Cloud computing

## Essential Characteristics

**Rapid elasticity:** Os recursos podem ser provisionados e liberados de forma elástica, em alguns casos automaticamente, para expandir rapidamente e sob demanda.

<https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>





# Cloud computing

## Essential Characteristics

**Measured service:** Os sistemas em nuvem controlam e otimizam automaticamente o uso de recursos, aproveitando uma capacidade de medição em algum nível de abstração apropriado para o tipo de serviço (por exemplo, armazenamento, processamento, largura de banda e contas de usuário ativas).

<https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>



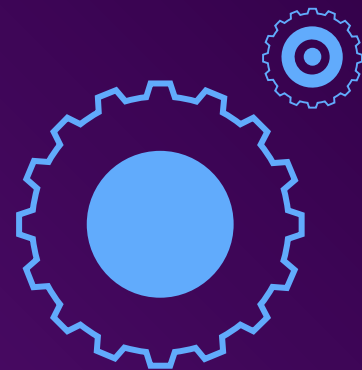


# Cloud computing

## Essential Characteristics

O uso de recursos pode ser monitorado, controlado e relatado, proporcionando transparência tanto para o provedor quanto para o consumidor do serviço utilizado.

<https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>





# Cloud computing

## Service Models

01

SaaS

Software as a Service

02

PaaS

Platform as a Service

03

IaaS

Infrastructure as a Service



# Cloud computing

## Deployment Models

01

Private cloud

02

Community  
cloud

03

Public cloud

04

Hybrid cloud

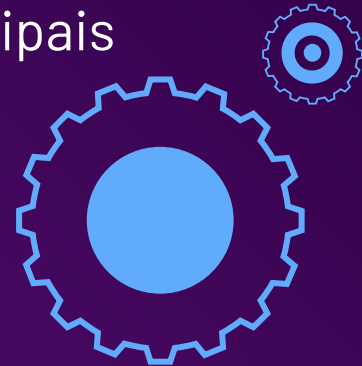




# Cloud computing

## Exercícios

Como exercício de fixação, peço que você caro aluno(a), faça uma pesquisa sobre os modelos de serviços e modelos de implantação de computação em nuvem. Esta pesquisa deverá resultar em um relatório a ser entregue, contendo as principais características e casos de implementação (quais recursos computacionais) que cada modelo entrega.

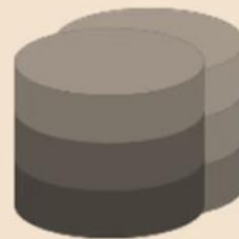
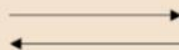
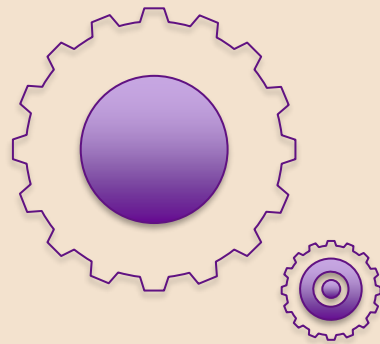




# Camadas de Distribuição

MVC

Model-View-Controller



View

Controller

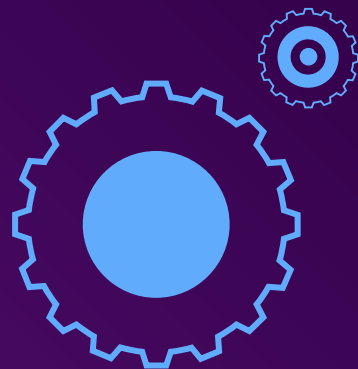
Model



# MVC Pattern

“O **MVC (Model-View-Controller)**, é um padrão de arquitetura de software formulado na década de 1970, focado no reuso de código e na separação de conceitos em três camadas interconectadas.”

<https://pt.wikipedia.org/wiki/MVC>

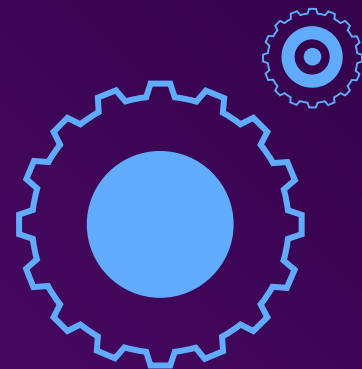




# MVC Pattern

No padrão MVC, cada camada possui responsabilidades distintas onde as alterações feitas no layout (**Visão**), não afetam a manipulação de dados (**Modelo**), que por sua vez poderão ser reorganizados sem alterar o layout.

<https://pt.wikipedia.org/wiki/MVC>

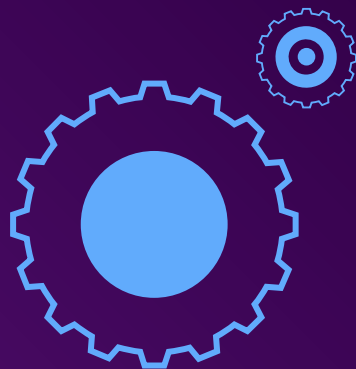




# mvc Pattern

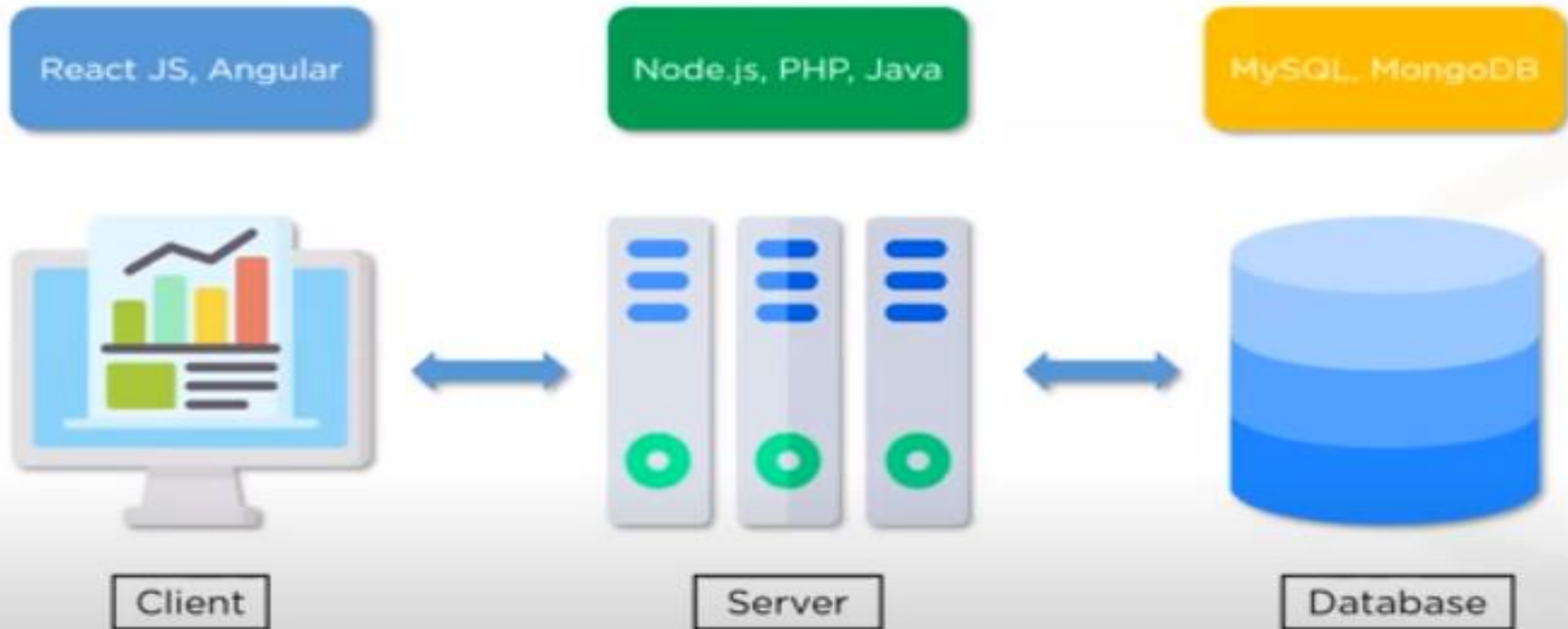
A camada de **Controle** por sua vez, interliga estas duas camadas (Visão e Modelo), gerenciando o fluxo da aplicação.

<https://pt.wikipedia.org/wiki/MVC>





# Camadas de Distribuição







# MVC Framework

A arquitetura de sistemas MVC é baseada em três elementos:



**Figura 6.1: Estrutura do MVC**

Fonte: Equipe produção CEAD/IFES (2011)



# MVC Framework Model

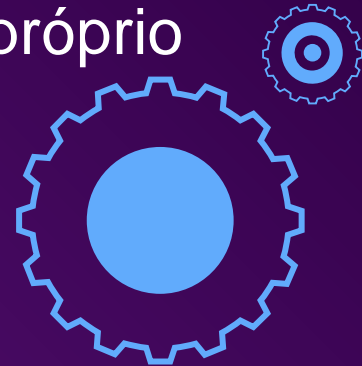
1. O **modelo** representa os dados da aplicação e as regras de negócio que governam o acesso e a modificação dos dados.





# MVC Framework Model

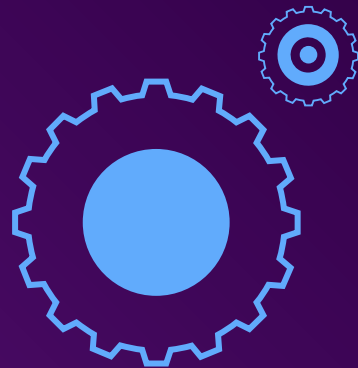
2. O **modelo** fornece ao controlador a capacidade de acessar as funcionalidades da aplicação encapsuladas pelo próprio modelo.





# MVC Framework Model

3. O **modelo** pode ser subdividido em “regras do negócio” e “persistência dos dados”.





# MVC Framework

## View

1. A **visão** renderiza o conteúdo de uma parte particular do **modelo** e encaminha para o **controlador** as ações do usuário.

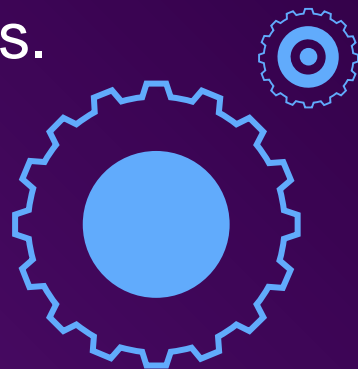




# MVC Framework

## View

2. A **visão** acessa também os dados do **modelo** via **controlador** e define como esses dados devem ser apresentados.

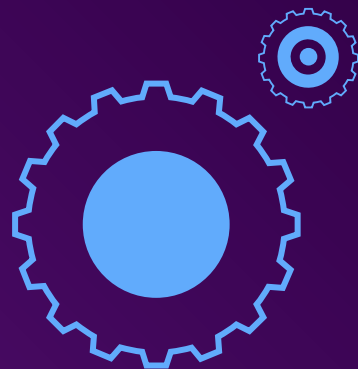




# MVC Framework

## View

3. A **visão** apresenta os dados para o usuário sem se preocupar com a origem deles.

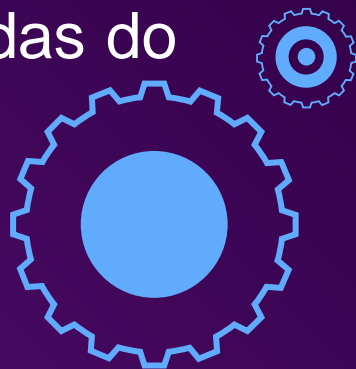




# MVC Framework

## Controller

1. O **controlador** define o comportamento da aplicação. É ele que interpreta as ações do usuário e as mapeia para chamadas do **modelo**.







# MVC Framework

## Controller

2. Com base na ação do usuário e no resultado do processamento do **modelo**, o **controlador** seleciona uma **visualização** a ser exibida como parte da resposta à solicitação do usuário.

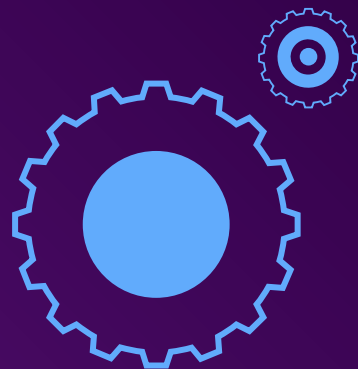




# MVC Framework

## Controller

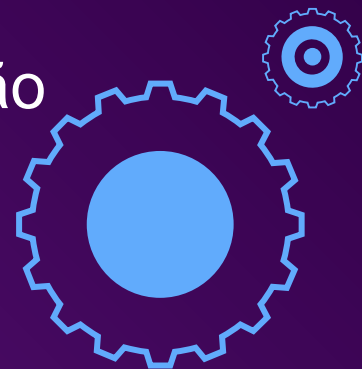
3. Há, normalmente, um **controlador** para cada conjunto de funcionalidades relacionadas.





# MVC Framework benefits

- reaproveitamento de código;
- facilidade de manutenção;
- integração de equipes e/ou divisão de tarefas;

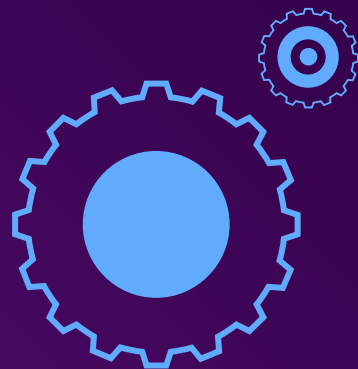




# MVC Framework

## benefits

- camada de persistência independente;
- implementação de segurança;
- aplicação escalável.
- facilidade na alteração da interface da aplicação;





# MVC Framework

## benefits - code reuse

Como característica fundamental do benefício do **reaproveitamento do código** é: problemas que já foram solucionados no passado, não devem ser resolvidos novamente.

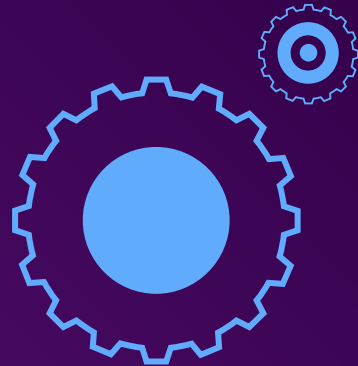




# MVC Framework

## benefits – Ease of Maintenance

- Gerenciar múltiplos visualizadores usando o mesmo modelo, facilitando a manutenção, testes e atualização de sistemas múltiplos;

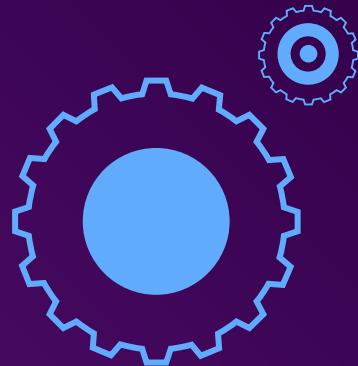




# MVC Framework

## benefits – Ease of Maintenance

- Como a parte visual é separada do modelo de negócio, é possível alterar a parte visual sem alterar o sistema todo.





# MVC Framework

benefits – integration of teams  
and/or division of tasks

- É possível ter desenvolvimento em paralelo para o modelo, visualizador e controle, pois são independentes, ganhando em produtividade.



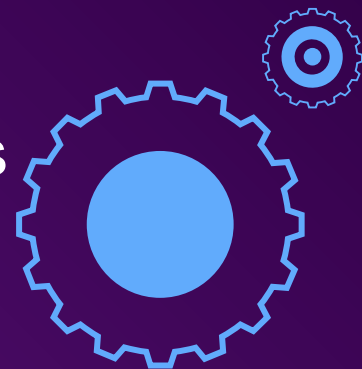




# MVC Framework

## DISADVANTAGES

- requer uma quantidade maior de tempo para analisar e modelar o sistema;
- requer pessoal especializado;
- não é aconselhável para pequenas aplicações.



# Grato!

Possui alguma dúvida?

jeveson.santos@ufv.br  
UFV – IEF

<http://lattes.cnpq.br/5592960118083061>

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, infographics & images by **Freepik**

Please keep this slide for attribution

