

CRUD MVC com PHP

Prof. Daniel Di Domenico



O que já sabemos

- Envio de informações do cliente para o servidor
- Verbos/métodos HTTP:
 - GET
 - POST
- Captura dos valores no PHP:
 - Superglobais:
 - `$_GET`
 - `$_POST`
- Formulários
- Validações
- Orientação à objetos
- Persistência de dados



Objetivo das aulas

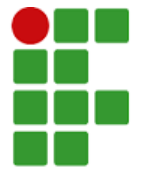
- Vamos desenvolver um CRUD com PHP:
 - **CRUD:** Create, Read, Update and Delete
 - Arquitetura MVC (*Model, View, Controller*)
 - Orientação a Objetos
- Ferramentas necessárias (pilha LAMP)
 - Apache com PHP instalado
 - Suporte a PDO e MySQL
 - Banco de dados MySQL
 - phpMyAdmin ou outro cliente com suporte a MySQL
 - Visual Studio Code (VSCode)



Arquitetura MVC

- Forma de estruturar um projeto, **dividindo a implementação em camadas** com funções específicas
- Vantagens do MVC:
 - Facilita a manutenção e alteração do código (organização)
 - Isolamento das regras de negócio da lógica de apresentação
 - Diminui o acoplamento e aumenta a coesão das classes
 - Possibilita o reaproveitamento de classes e partes da implementação em projetos futuros
 - As camadas são independentes
 - Padrão que pode ser utilizado em diversos tipos de projetos
 - Desktop, Web e Mobile

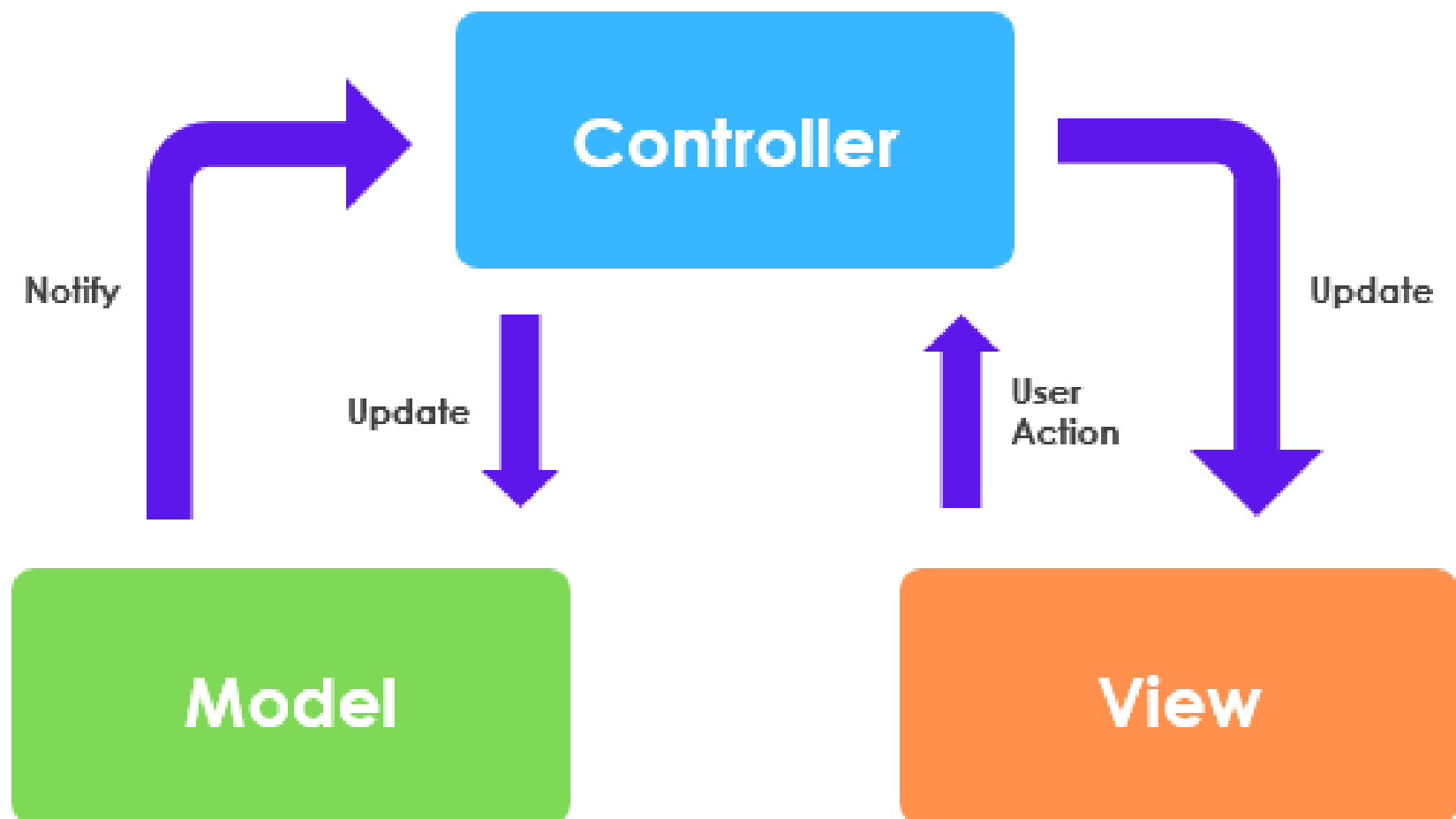
Camadas do MVC



**INSTITUTO
FEDERAL**

Paraná

Campus
Foz do Iguaçu



Camadas do MVC

- **Model:** gerencia os dados do sistema, representando o domínio da aplicação
 - Pode também incluir as regras de negócio (não é obrigatório)
- **View:** camada que apresenta as informações do sistema de forma visual ao usuário
 - Interage com o usuário por meio de botões, campos e mensagens
 - Permite entradas (questionamentos) e saídas (respostas)
- **Controller:** liga as requisições enviadas pelo View com as respostas do Model
 - Permite a comunicação entre o Model e o View
 - É a única camada que **conhece quem é o responsável por executar a operação** que retornará uma resposta ao usuário

MVC

- É possível construir uma aplicação somente com essas 3 camadas (Model, View e Controller)?
 - Geralmente **NÃO**
- A principal premissa do MVC é estruturar a aplicação em **camadas**
 - Podem haver outras, como:
 - Acesso ao banco de dados (camada DAO)
 - Serviço para validações de dados (camada Service)

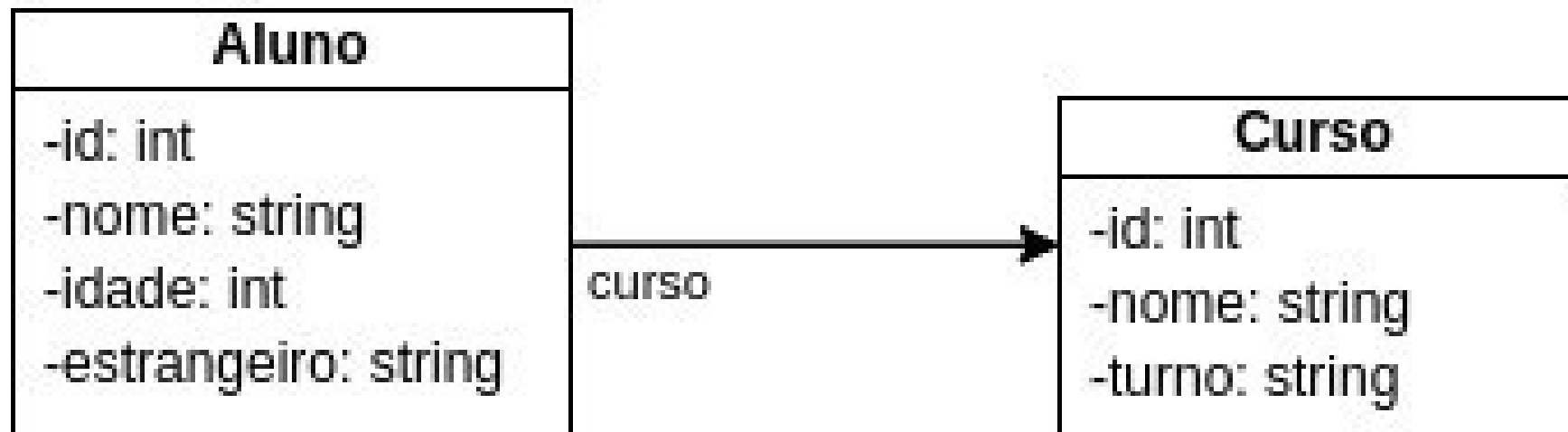
CRUD

- **CRUD** é uma interface de um sistema que contempla as operações de:
 - **Create** (inserir):
 - Criar ou adicionar novas entradas
 - **Read** (listar ou busca):
 - Ler, recuperar ou visualizar entradas existentes
 - **Update** (atualizar):
 - Atualizar ou editar entradas existentes
 - **Delete/Destroy** (excluir)
 - Remover entradas existentes

Passos do CRUD

- **1:** Criar base de dados no MySQL
 - Script SQL disponibilizado pelo professor
- **2:** Criar estrutura do projeto em PHP no padrão MVC
 - Estrutura inicial disponibilizada pelo professor
- **3:** Criar conexão com a base de dados utilizando PDO
- **4:** Implementar o CRUD de Alunos

Modelo de classes proposto

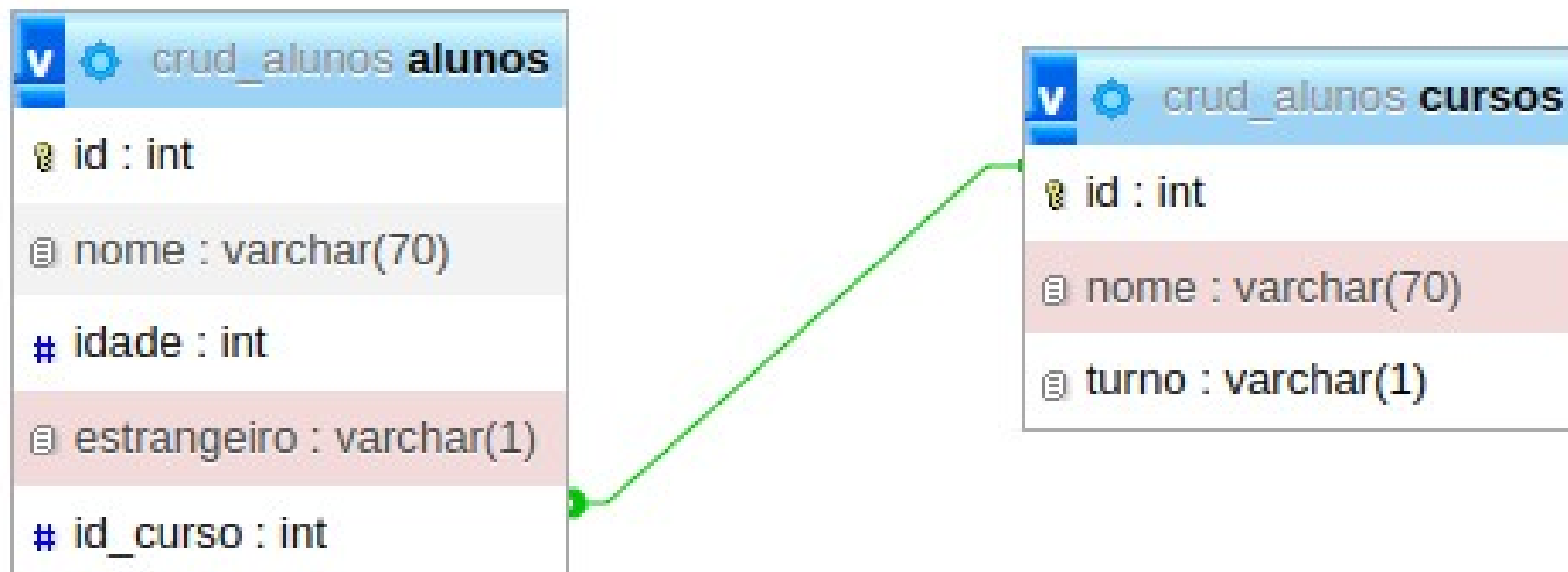


Modelo relacional da base de dados proposto



**INSTITUTO
FEDERAL**
Paraná

Campus
Foz do Iguaçu



Comando úteis PHP

- **__DIR__**: constante que retorna o caminho arquivo atual
 - Será utilizada para incluir arquivos em outras páginas
- **die("mensagem")**: função que exibe a mensagem de erro na tela, interrompendo a execução do *script* PHP
- **try...catch(\$e)**: tratamento de exceções
- **?<tipo>**: permite definir que um tipo de dados aceite nulo
 - Muito utilizado para definir tipos em classes

Implementação

- Implementação do CRUD será baseado nos modelos propostos:
 - O mesmo poderá ser incrementado futuramente
 - Ex.: novas classes, novas camadas, validações, login....
- Veremos conceitos de orientação a objetos e persistência desses objetos
- Em cada aula:
 - Novas implementações serão realizadas para completarmos o CRUD com todas as operações