**1. Connection.php (Database)**

**Função geral:**

Responsável por **criar a conexão com o banco de dados PostgreSQL** usando variáveis de ambiente do arquivo .env.

**Funções:**

* **connect()**  
  → Lê as variáveis DB\_HOST, DB\_NAME, DB\_USER, DB\_PASS, DB\_PORT do .env.  
  → Cria um objeto **PDO** configurado para lançar exceções (ERRMODE\_EXCEPTION).  
  → Retorna a conexão pronta para uso nos models.  
  → Em caso de erro, exibe a mensagem e encerra a execução.

**2. Prato.php (Model)**

**Função geral:**

Gerencia as **operações CRUD da tabela pratos** no banco de dados.

**Funções:**

* **\_\_construct()**  
  → Cria a conexão com o banco através da classe Connection.
* **create($data)**  
  → Insere um novo prato no banco (nome, descricao, preco, categoria).  
  → Retorna o **ID do novo prato** criado.
* **getAll()**  
  → Busca todos os pratos existentes na tabela.  
  → Retorna um array com todos os registros.
* **find($id)**  
  → Busca um prato específico pelo id.  
  → Retorna os dados do prato ou false se não existir.
* **update($id, $data)**  
  → Atualiza as informações de um prato específico.  
  → Retorna a **quantidade de linhas afetadas** (geralmente 1 ou 0).
* **delete($id)**  
  → Exclui o prato pelo id.  
  → Retorna true se excluiu, false se o prato não existia.

**3. PratosController.php (Controller)**

**Função geral:**

Recebe as requisições HTTP, chama o model (Prato) e retorna as respostas em **formato JSON** com o **status HTTP** adequado.

**Funções:**

* **\_\_construct()**  
  → Instancia o model Prato para uso nas operações.
* **index()**  
  → Lista todos os pratos.  
  → Retorna status **200 (OK)** e o JSON com os dados.
* **show($id)**  
  → Busca um prato específico.  
  → Retorna **200 (OK)** se encontrado ou **404 (Not Found)** se não existir.
* **store($data)**  
  → Valida se nome e preco foram enviados.  
  → Cria um novo prato via model.  
  → Retorna **201 (Created)** com o ID do novo prato.
* **update($id, $data)**  
  → Atualiza um prato existente.  
  → Retorna:
  + **200 (OK)** se atualizou;
  + **404 (Not Found)** se não achou o prato;
  + **500 (Internal Server Error)** se houve falha.
* **destroy($id)**  
  → Exclui um prato pelo ID.  
  → Retorna **200 (OK)** se excluído ou **404 (Not Found)** se não existia.

**4. index.php (Roteador principal / Entry point)**

**Função geral:**

É o **ponto de entrada da API** — recebe a requisição HTTP e decide qual método do PratosController será executado.

**Passos principais:**

1. **Carrega o autoload e o .env**  
   → Usa vlucas/phpdotenv para ler as variáveis de ambiente.
2. **Define cabeçalhos HTTP (CORS e JSON)**  
   → Permite acesso de front-ends externos (ex: React).  
   → Responde automaticamente a requisições OPTIONS (pré-flight).
3. **Cria o controller (PratosController)**  
   → É a ponte entre a rota e o banco.
4. **Lê o método HTTP e a URL**  
   → Identifica se é GET, POST, PUT ou DELETE.  
   → Extrai o recurso (/pratos) e o ID (ex: /pratos/5).
5. **Executa a ação correspondente:**
   * GET /pratos → lista todos (index)
   * GET /pratos/{id} → mostra um prato (show)
   * POST /pratos → cria novo (store)
   * PUT /pratos/{id} → atualiza (update)
   * DELETE /pratos/{id} → remove (destroy)
6. **Retorna resposta JSON com código HTTP adequado.**

**5. .env**

**Função geral:**

Armazena as **configurações sensíveis e variáveis de ambiente** do sistema.

**Principais variáveis:**

* DB\_HOST, DB\_NAME, DB\_USER, DB\_PASS, DB\_PORT → Conexão com PostgreSQL.
* APP\_ENV, APP\_DEBUG → Configurações do ambiente da aplicação (desenvolvimento/produção).

**6. composer.json**

**Função geral:**

Gerencia as **dependências e autoload** do projeto PHP.

**Campos principais:**

* "require" → Define dependências, como vlucas/phpdotenv.
* "autoload" → Carrega automaticamente os arquivos do projeto (Connection.php e Prato.php).
* "name" e "description" → Informações do projeto.