



# **Projeto Integrador 2017.2**

## **Módulo de Desenvolvimento Web**

Elias Gomes Campos

**Curso Técnico em Informática**

Educação a Distância

2018





## **EXPEDIENTE**

### **Professor Autor**

Elias Gomes Campos

### ***Design Instrucional***

Annara Perboire

Hugo Cavalcanti

### **Revisão de Língua Portuguesa**

Letícia Garcia

### **Diagramação**

Hugo Cavalcanti

### **Coordenação Executiva**

George Bento Catunda

### **Coordenação Geral**

Paulo Fernando de Vasconcelos Dutra

Conteúdo produzido para os Cursos Técnicos da Secretaria Executiva de Educação Profissional de Pernambuco, em convênio com o Ministério da Educação.

**Março, 2018**



## 1. Projeto Integrador - Desenvolvimento Web

### 1.1 Apresentação

O Projeto Integrador do módulo de Desenvolvimento Web 2016.2 oferece uma oportunidade ao estudante de aperfeiçoar os conhecimentos vistos em módulos separados e exercitados nas atividades semanais.

Planejamos este projeto para unificar todos os conhecimentos vistos anteriormente na confecção de um produto, que reúna o que foi visto e exercitado nas competências já apresentadas. Assim, será necessária a leitura atenta do conteúdo dos cadernos de estudo das disciplinas de **Banco de Dados, Lógica de Programação, Desenvolvimento Web com PHP e Orientação a Objeto com PHP**, e também assistir às video aulas disponibilizadas no AVA. Utilize-os para tirar dúvidas, o que é preciso para a realização da atividade se encontra neles, mas se ainda assim restar dúvidas, utilize o fórum do AVA e pesquise na internet outras formas de exposição do conteúdo.

Este projeto beneficia o aluno de várias formas. Em uma primeira perspectiva, ele vem como uma forma de fortalecer o que foi visto nas disciplinas do módulo de forma separada, unindo esses conteúdos para uma melhor fixação do conhecimento. Em outra perspectiva, se o estudante não conseguiu a média em uma competência, a nota correspondente àquela competência pode ser alterada, caso a nota do Projeto Integrador seja maior.

### 1.2 O produto

Suponha que você foi contratado para participar do desenvolvimento de uma aplicação web para uma **Pizzaria**, que apenas cadastra, lista e exclui os dados dos **funcionários, cargos, pizzas e comandas**.

O sistema deverá possuir páginas para cadastros de novos **funcionários, cargos, pizzas e comandas**, com formulários para preenchimento dos dados que serão enviados para armazenamento no banco de dados. O sistema deverá também possuir páginas que exibam a lista de todos os **funcionários, cargos, pizzas e comandas** realizadas, tais informações deverão ser buscadas no banco de dados.



### 1.3 O Sistema

O sistema deverá possuir as seguintes funcionalidades:

- **Página de Pizzas:** Esta página deverá possuir um formulário para preenchimento dos campos: **Nome** (text), **Ingredientes** (text), **Valor** (number), **Status** (combobox com os valores: *disponível* e *indisponível*), e **Tipo** (combobox com os valores: *salgada*, *doce*, *fit* e *vegetariana*). Ao clicar no botão Cadastrar, serão invocadas as páginas de controle e persistência - para salvar os dados preenchidos no banco de dados. Logo abaixo do formulário, e na mesma página, deverão ser listados todos as pizzas cadastradas. Um botão para excluir deverá ficar ao lado de cada Pizza listada e, ao clicar no botão excluir, serão invocadas as páginas de controle e persistência - para excluir o respectivo registro no banco de dados. A **Figura 1** apresenta tais características dessa página.

The screenshot displays the 'Pizzeria Italiana' web interface. At the top, a navigation bar includes links for Home, Pizzas, Comandas, Funcionários, Cargos, and the user name Eliomar Campos. Below the navigation bar is a header image of pizzas with the text 'Pizzaria Italiana'. The main content area is titled 'Nova Pizza' and contains a form with the following fields: 'Nome' (text input), 'Ingredientes' (text input), 'Valor' (text input), 'Status' (dropdown menu with 'Disponível' selected), and 'Tipo' (dropdown menu with 'Salgada' selected). A blue 'Cadastrar' button is located below the form. Below the form is a section titled 'Lista de Pizzas' containing a table with the following data:

Código	Nome	Ingredientes	Valor	Status	Tipo	
3	Margherita	Tomates assados, muçarela, manjericão e azeite	R\$ 22	Disponível	Salgada	<button>Excluir</button>
4	Frango	Frango desfiado, molho de tomate, queijo muçarela, milho verde, pimentão, azeitonas e orégano.	R\$ 25	Indisponível	Salgada	<button>Excluir</button>
5	Brie com pera	queijo brie com pera	R\$ 30	Disponível	Vegetariana	<button>Excluir</button>

Figura 1 – Página Pizzas.

Fonte: Autor.

Descrição: A figura apresenta como deverá ser a página de pizzas, esta possui um formulário para cadastro de uma nova pizza e uma tabela contendo a lista de todas as pizzas cadastradas no banco de dados. Associado a cada linha da tabela, um botão para excluir o registro desejado.



- **Página de Comandas:** esta página deverá possuir um formulário para preenchimento dos campos: **Data da Compra** (date), **Forma de Pagamento** (combobox com os valores: *À Vista*, *Débito*, *Crédito*, e *Cheque*), **Status** (combobox com os valores: *Paga*, *A Pagar*, e *Cancelada*), **Funcionário** (combobox com os nomes dos funcionários cadastrados no banco de dados) e **Pizza** (combobox com os nomes das pizzas cadastradas no banco de dados). O combobox **Pizza** deverá listar apenas as pizzas com status disponível (**é obrigatório implementar esta regra!**). Ao clicar no botão Cadastrar, serão invocadas as páginas de controle e persistência - para salvar os dados preenchidos no banco de dados. Logo abaixo do formulário, e na mesma página, deverão ser listados todas as comandas cadastradas. Um botão para excluir deverá ficar ao lado de cada Comanda listada e, ao clicar no botão excluir, serão invocadas as páginas de controle e persistência - para excluir o respectivo registro no banco de dados. A **Figura 2** apresenta tais características dessa página.

The screenshot displays the 'Pizzeria Italiana' web application. At the top, there is a navigation bar with links: Home, Pizzas, Comandas, Funcionários, Cargos, and Eliomar Campos. Below the navigation bar is a header image of pizzas with the text 'Pizzaria Italiana'. The main content area is titled 'Nova Comanda' and contains a form for creating a new order. The form fields are: 'Data Compra' (a date input field with the placeholder 'dd/mm/aaaa'), 'Forma de Pagamento' (a dropdown menu with 'À Vista' selected), 'Status' (a dropdown menu with 'Paga' selected), 'Funcionário' (a dropdown menu with 'Arnold schwarzenegger' selected), and 'Pizza' (a dropdown menu with 'Margherita' selected). Below the form is a blue button labeled 'Cadastrar'. Underneath the form is a section titled 'Lista de Comandas' containing a table with the following data:

Código	Data Compra	Forma de Pagamento	Status	Funcionário	Pizza	
4	15/03/2017	À Vista	A pagar	Arnold schwarzenegger	Margherita	<button>Excluir</button>
6	08/03/2017	Crédito	Paga	Severino Costa Filho	Brie com pera	<button>Excluir</button>

Figura 2 – Página Comandas.

Fonte: Autor.

Descrição: A figura apresenta como deverá ser a página de comandas, esta possui um formulário para cadastro de uma nova comanda e uma tabela contendo a lista de todas as comandas cadastradas no banco de dados. Associado a cada linha da tabela, um botão para excluir o registro desejado.



- **Página de Cargo:** esta página deverá possuir um formulário para preenchimento dos campos: **Nome** (text), **Descrição** (text) e **Salário Base** (number). Ao clicar no botão Cadastrar, serão invocadas as páginas de controle e persistência - para salvar os dados preenchidos no banco de dados. Logo abaixo do formulário e, na mesma página, deverão ser listados todos os Cargos cadastrados. Um botão para excluir deverá ficar ao lado de cada Cargo listado. Ao clicar no botão excluir, serão invocadas as páginas de controle e persistência - para excluir o respectivo registro no banco de dados. A **Figura 3** apresenta tais características dessa página.

Código	Nome	Descrição	Salário Base	
5	Estagiário	Contrato de Experiência	R\$ 450	Excluir
6	Gerente	Responsável de forma geral	R\$ 5500	Excluir

Figura 3 – Página Cargos.

Fonte: Autor.

Descrição: A figura apresenta como deverá ser a página de cargos, que possui um formulário para cadastro de um novo cargo e uma tabela contendo a lista de todos os cargos cadastrados no banco de dados. Associado a cada linha da tabela, um botão para excluir o registro desejado.

- **Página de Funcionário:** esta página deverá possuir um formulário para preenchimento dos campos: **Nome** (text), **CPF** (text), **Data da Admissão** (date), **Status** (combobox com os valores: *Ativo*, *Demitido*, *Férias* e *Licença Médica*), e **Cargo** (combobox com os nomes dos cargos cadastrados no banco de dados). Ao clicar no botão Cadastrar, serão invocadas as páginas de controle e





persistência - para salvar os dados preenchidos no banco de dados. Logo abaixo do formulário e, na mesma página, deverão ser listados todos os funcionários cadastrados. Um botão para excluir deverá ficar ao lado de cada funcionário listado e, ao clicar no botão excluir, serão invocadas as páginas de controle e persistência - para excluir o respectivo registro no banco de dados. A **Figura 4** apresenta tais características dessa página.

Código	Nome	CPF	Data Admissão	Status	Cargo	
6	Severino Costa Filho	000.000.000-00	09/03/2017	Ativo	Estagiário	Excluir
7	Arnold schwarzenegger	000.000.000-00	22/03/2017	Férias	Gerente	Excluir

Figura 4 – Página Funcionários.

Fonte: o autor.

Descrição: a figura apresenta como deverá ser a página de funcionários. Esta possui um formulário para cadastro de um novo funcionário e uma tabela contendo a lista de todos os funcionários cadastrados no banco de dados. Associado a cada linha da tabela, um botão para excluir o registro desejado.

- **Página de Detecção de Plágio:** a fim de evitar trabalhos totalmente copiados de outros colegas alunos, o menu dessa página deverá possuir o seu primeiro e último nome (**e somente o primeiro e último**). Um formulário para preenchimento dos campos: **Primeiro Nome** (texto) e **Último Nome** (texto). Ao clicar no botão Cadastrar, serão invocadas as páginas de controle e persistência - para salvar os dados preenchidos no banco de dados. Todas as classes dessa página deverão ser nomeadas com o seu primeiro e último nome. Você deverá cadastrar apenas os nomes





dos seus “parentes” (que possuam o seu último nome), tais dados podem ser fictícios, o importante é conter o seu último nome nos cadastros. Logo abaixo do formulário e, na mesma página, deverão ser listados todos os “parentes” cadastrados. Um botão para excluir deverá ficar ao lado de cada “parente” listado. Ao clicar no botão excluir, serão invocadas as páginas de controle e persistência - para excluir o respectivo registro no banco de dados. A **Figura 5** apresenta tais características dessa página.

The screenshot shows a web application for 'Pizzeria Italiana'. The navigation bar includes links for Home, Pizzas, Comandas, Funcionários, Cargos, and Eliomar Campos. The main header features the text 'Pizzaria Italiana' over a background image of pizzas. Below the header, there is a registration form for 'Nome Completo' with fields for 'Primeiro Nome' and 'Último Nome', and a 'Cadastrar' button. Underneath the form is a section titled 'Lista de Parentes' containing a table with three columns: 'Código', 'Primeiro Nome', and 'Último Nome'. Each row in the table has an 'Excluir' button next to it.

Código	Primeiro Nome	Último Nome	
4	Eliomar	Campos	Excluir
5	Rafaela	Campos	Excluir
7	Mayara	Campos	Excluir

Figura 5 – Página Detecção de Plágio.

Fonte: o autor.

Descrição: a figura apresenta como deverá ser a página de Detecção de Plágio, esta possui um formulário para cadastro de um novo “parente” e uma tabela contendo a lista de todos os “parentes” cadastrados no banco de dados. Associado a cada linha da tabela, um botão para excluir o registro desejado.

## 1.4 Arquitetura

O projeto já está iniciado com a organização desejada. Utilizando **PHP Orientado a Objetos**, você deve seguir estes mesmos requisitos para completar o projeto. As páginas referentes ao cadastro, listagem e exclusão dos **Funcionários e Cargos** já estão implementadas, bastando você apenas



completar o projeto adicionando as páginas de Pizzas, Comandas e Detecção de Plágio.

Como você já viu em disciplinas anteriores, na orientação a objetos cada classe tem apenas uma responsabilidade. Dessa forma, o trabalho fica mais organizado e fácil de entender, entre outros benefícios. Portanto, este projeto foi organizado de acordo com os padrões de projetos (mais conhecido como *design pattern*). Para isso, organizamos as classes em quatro partes: **visualização**, **controle**, **modelo** e **persistência**, ou como são mais conhecidas: **view**, **controller**, **model** e **persistence**, respectivamente. A **Tabela 1** relembra as responsabilidades de cada parte.





View	Controller	Persistence	Model
			
<p>Páginas responsáveis apenas pela forma como a informação é apresentada ao usuário, ou seja, é o design do website, composto de arquivos de layout HTML, CSS e imagens.</p> <p><b>Exemplo de arquivos:</b> index.php view_funcionario.php view_cargo.php</p>	<p>Páginas responsáveis apenas por receber dados provenientes das páginas da view e encaminhá-los para os métodos das classes de persistência.</p> <p><b>Exemplo de arquivos:</b> incluir_funcionario.php excluir_funcionario.php incluir_cargo.php excluir_cargo.php</p>	<p>Classes que contêm os métodos de acesso e manipulação dos dados no banco de dados. Por exemplo, cadastrar, remover, alterar e listar.</p> <p><b>Exemplo de arquivos:</b> dao_funcionario.php dao_cargo.php conexao.php</p>	<p>Classes que caracterizam e definem os objetos de cadastro do sistema. Tais objetos são utilizados principalmente pelas classes de persistência para enviar os dados contidos neles para as respectivas tabelas do banco de dados.</p> <p><b>Exemplo de arquivos:</b> class_funcionario.php class_cargo.php</p>

Tabela 1 – Arquitetura em camadas.

Fonte: o autor

Descrição: a figura apresenta a estrutura de pastas de acordo com os padrões de projeto, e descreve cada categoria: visualização, controle, persistência e modelo; citando exemplos de nomeação de seus respectivos arquivos.

**Observação:** As classes de persistência também são conhecidas como DAO (Data Access Object). São classes que preenchem os atributos das classes do modelo com dados trazidos de alguma fonte, que no nosso caso é o banco de dados. As classes DAO vão guardar os comandos SQL necessários para manipular o banco.



Como a aplicação já foi parcialmente desenvolvida pelo professor, você poderá estudar as páginas já construídas para entender melhor estas responsabilidades. A **Figura 6** apresenta a estrutura de pastas, arquivos e classes do projeto completo. A **Figura 7** apresenta a estrutura de pastas, arquivos e classes que serão disponibilizados para você. Observe na **Figura 7** que está sendo disponibilizado e mantido para você o mesmo padrão, porém, apenas contendo o cadastro, listagem e exclusão de filmes. Dessa forma, você deverá apenas acrescentar as demais páginas e classes. Na hora de acrescentar, é importante seguir os mesmos nomes do projeto completo da **Figura 6**, com exceção dos arquivos e classes de detecção de plágio, que deverão ser nomeados com o seu primeiro e último nome. Portanto, verifique, analise, e entenda as páginas e as classes de filme disponibilizadas, para conseguir adicionar as demais, seguindo o mesmo raciocínio e padrão.

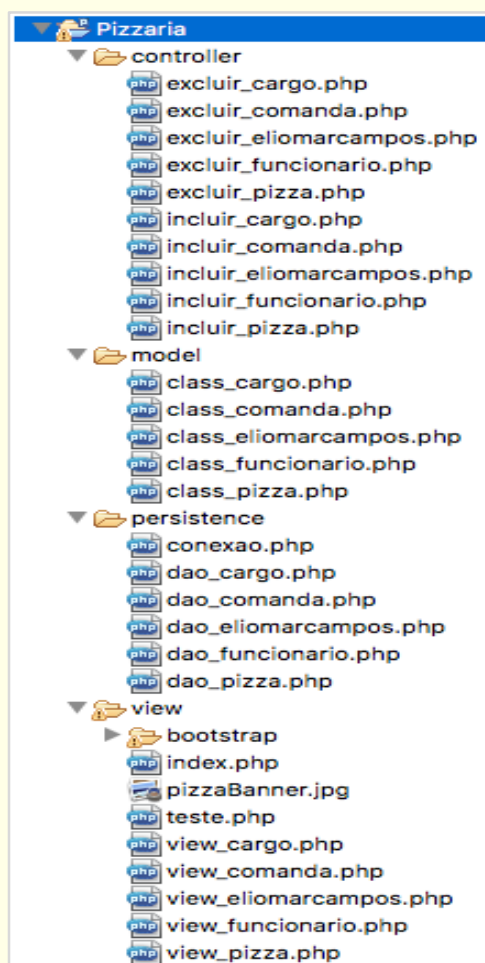


Figura 6 – Arquitetura em camadas do projeto completo.

Fonte: o autor

Descrição: a figura apresenta como deverá ficar a estrutura de pastas e os nomes dos arquivos do projeto completo a ser entregue pelo aluno ao professor.

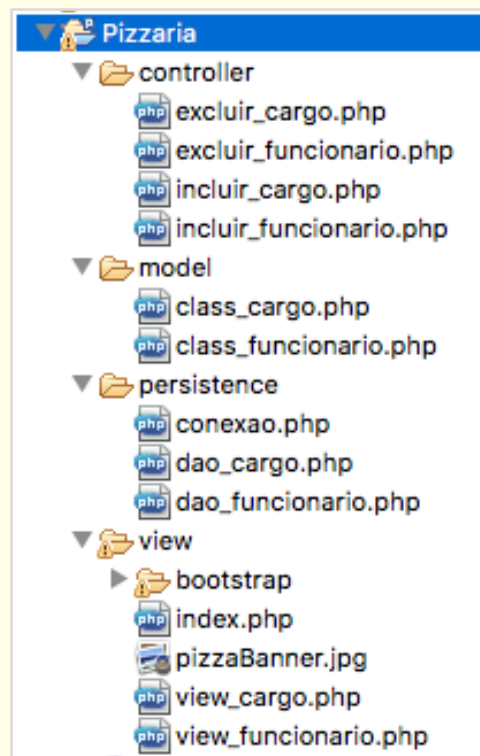


Figura 7 – Arquitetura em camadas do projeto incompleto.

Fonte: o autor

Descrição: a figura apresenta a estrutura de pastas e os nomes dos arquivos do projeto incompleto a ser entregue pelo professor para auxiliar o aluno.

## 1.5 Banco de Dados

Você deverá criar a base de dados, tabelas e os respectivos campos, seguindo as especificações seguintes:

- Nome da base de dados: **pizzaria**
- Tabela **pizza**:
  - `codigo` [INT, NOT NULL, AUTO\_INCREMENT, e PRIMARY KEY]
  - `nome` [VARCHAR(45), NOT NULL]
  - `ingredientes` [VARCHAR(100)]
  - `valor` [FLOAT]
  - `status` [VARCHAR(45)]



- `tipo` [VARCHAR(45)]
- Tabela **comanda**:
  - `codigo` [INT, NOT NULL, AUTO\_INCREMENT, e PRIMARY KEY]
  - `datacompra` [DATETIME, NOT NULL]
  - `formapagamento` [VARCHAR(45), NOT NULL]
  - `status` [VARCHAR(45), NOT NULL]
  - `codigofuncionario` [INT, NOT NULL, e **FOREIGN KEY REFERENCES funcionario**]
  - `codigopizza` [INT, NOT NULL, e **FOREIGN KEY REFERENCES pizza**]
- Tabela **cargo**:
  - `codigo` [INT, NOT NULL, AUTO\_INCREMENT, e PRIMARY KEY]
  - `nome` [VARCHAR(45), NOT NULL]
  - `descricao` [VARCHAR(100)]
  - `salariobase` [FLOAT]
- Tabela **funcionario**:
  - `codigo` [INT, NOT NULL, AUTO\_INCREMENT, e PRIMARY KEY]
  - `nome` [VARCHAR(45), NOT NULL]
  - `cpf` [VARCHAR(45)]
  - `dataadmissao` [DATETIME]
  - `status` [VARCHAR(45)]
  - `codigocargo` [INT, e **FOREIGN KEY REFERENCES cargo**]
- Tabela **eliomarcampos**: aqui deverá ser o seu primeiro e último nome.
  - `codigo` [INT, NOT NULL, AUTO\_INCREMENT, e PRIMARY KEY]



- `primeiroNome` [VARCHAR(45)]
- `ultimoNome` [VARCHAR(45)]

**ATENÇÃO:** É obrigatória a implementação das chaves estrangeiras (FOREIGN KEY) nos campos das tabelas que possuem tal especificação.

Portanto, o banco de dados ficará com a seguinte estrutura apresentada na **Figura 8** abaixo:

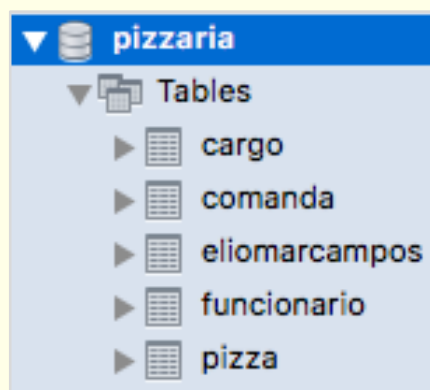


Figura 8 – Estrutura do banco de dados.

Fonte: o autor

Descrição: a figura apresenta como deverá ficar a estrutura das tabelas do banco de dados a ser entregue pelo aluno.

## 1.6 Ambiente de Desenvolvimento

- **MySQL Server:** deverá ser utilizado o MySQL Server como sistema gerenciador do banco de dados (SGBD). A fim de facilitar o seu trabalho, no projeto disponibilizado, já existe uma classe de conexão pronta, basta modificar os valores de conexão ao servidor (IP, usuário, senha e base de dados).
- **MySQLWorkbench** ou **phpMyAdmin:** a ferramenta para facilitar o gerenciamento dos dados é opcional, você poderá escolher até mesmo outras. No entanto, recomendamos fortemente uma dessas duas citadas.
- **XAMPP:** esse deverá ser o servidor web apache utilizado. Poderia ser também o WAMP ou outros, mas recomendamos o XAMPP para seguirmos um padrão e facilitar nosso trabalho.
- **Aptana Studio:** a IDE para desenvolvimento do código php é opcional, pode ser Notepad++



ou outros.

## 1.7 Resumo da proposta

A **Tabela 2** abaixo, resume todas as especificações solicitadas no projeto:

REQUISITO	VERIFICAÇÃO
Banco de Dados	Serão analisados os comandos de criação de cada tabela no arquivo SQL. Por isso, você deverá enviar o arquivo de backup do banco de dados (em .sql ou em .txt).
Model	Serão analisadas as classes do modelo, responsável pela definição e preenchimento dos dados nos objetos. Seus atributos devem ser privados e os métodos gets e sets para manipulação desses atributos devem ser públicos. Serão analisados os arquivos: class_pizza.php, class_comanda.php, e class_eliomarcampos.php (este deverá ser com o seu nome).
DAO ou Persistence	Serão analisadas as classes de persistência ou manipulação direta com o banco de dados. Essas classes deverão possuir os métodos para cadastro, listagem de todos os registros e exclusão de um registro. Serão analisados os arquivos: dao_pizza.php, dao_comanda.php e dao_eliomarcampos.php (este deverá ser com o seu nome).
Controller	Serão analisadas as classes que recebem os valores dos formulários quando clicamos nos botões para cadastrar ou excluir. Essas classes deverão invocar os métodos das classes de persistência, para cadastrar ou excluir. Elas apenas recebem os valores e criam os objetos com esses valores, porém o acesso direto ao banco, para cadastrar e excluir, só quem tem são as classes de persistência. Serão analisados os arquivos: incluir_pizza.php, excluir_pizza.php, incluir_comanda.php, excluir_comanda.php, incluir_eliomarcampos.php e excluir_eliomarcampos.php (estas duas últimas deverão ser com o seu nome).
View	Serão analisadas as páginas de design e interação com o usuário, mais conhecidas com front-end. Tais páginas serão responsáveis também por enviar os dados de formulários para as classes de controle. Serão analisados os arquivos: view_pizza.php, view_comanda.php e view_eliomarcampos.php (este deverá ser com o seu nome). Caso queira, poderá alterar as imagens e o CSS, mas NÃO modifique a estrutura do HTML de modo algum. Qualquer alteração estética NÃO terá acréscimo na pontuação.

Tabela 2 – Resumo das solicitações

Fonte: o autor

Descrição: A tabela descreve o resumo de cada solicitação do projeto que deverá ser realizado pelo aluno e avaliado pelos tutores virtuais.





## 1.8 Arquivos para Envio

O projeto deve ser enviado compactado em .ZIP ou .RAR. Lembre-se de que este arquivo não pode exceder 10 Mb. O nome do arquivo compactado deve ser o seu nome, por exemplo **EliomarGomesCampos.zip**. Ele deve conter a pasta base do projeto com os arquivos PHP na mesma estrutura da **Figura 6**. Acrescente o arquivo .sql do banco (ou em .txt) contendo os comandos de criação e valores das tabelas (as ferramentas de SGBDs já possuem funções que geram esse arquivo, geralmente é a opção exportar). Este arquivo deve ser nomeado **banco.sql**. Não deve ser enviado nenhum outro arquivo que não foi solicitado.

**ATENÇÃO:** O Projeto Integrador é individual. Projeto enviado sem a página de detecção de plágio, de acordo com o seu primeiro e último nome, ou seja, com o nome de outro colega, será considerado plágio e, portanto, receberá nota zero. Assim também para essa mesma página, **os valores preenchidos no banco, os nomes das classes, arquivos, variáveis, também deverá conter o seu último nome**. Portanto, não se esqueça de enviar o projeto com a página de detecção de plágio, pois, sem ela, o projeto receberá nota zero.

