



Projeto Integrador – Desenvolvimento Web

Eliomar Gomes Campos

Curso Técnico em Informática
Educação a Distância
2017



EXPEDIENTE

Professor Autor

Eliomar Gomes Campos

Design Instrucional

Deyvid Souza Nascimento
Renata Marques de Otero
Terezinha Mônica Sinício Beltrão

Revisão de Língua Portuguesa

Letícia Garcia

Diagramação

Izabela Cavalcanti

Coordenação

Anderson Elias

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.

Setembro, 2017



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**. Esse material também está disponível nas videoaulas 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 é uma maneira 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 **locadora de veículos**, que apenas cadastra, lista e exclui os dados dos **funcionários, cargos, clientes, veículos** e locações.

O sistema deverá possuir páginas para cadastros de novos **funcionários, cargos, clientes, veículos e locações**, 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, clientes, veículos e locações** realizadas, tais informações deverão ser



buscadas no banco de dados.

1.3 O Sistema

O sistema deverá possuir as seguintes funcionalidades:

- **Página de Veículos:** esta página deverá possuir um formulário para preenchimento dos campos: **Modelo** (text), **Cor** (text), **Ano** (number), **Placa** (text), **Tipo** (combobox com os valores: *Hach Médio, Hatch Compacto, Sedam, Suv e Motocicleta*), **Status** (combobox com os valores: *Livre, Alugado, Desativado, Manutenção*). 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 veículos cadastrados. Um botão para excluir deverá ficar ao lado de cada veículo 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 1** apresenta tais características dessa página.

The screenshot shows a web application for a vehicle rental company. The top navigation bar includes links for Home, Veículos (selected), Locações, Clientes, Funcionários, Cargos, and Eliomar Campos. The main title is 'Locadora de Veículos' with the subtitle 'AQUILOCA'. Below the title is a banner image of a road through a green landscape. The page is divided into two main sections: 'Novo Veículo' (New Vehicle) and 'Lista de Veículos' (List of Vehicles).

Novo Veículo: This section contains input fields for 'Modelo', 'Cor', 'Ano', 'Placa', 'Tipo' (dropdown menu with 'Hach Médio' selected), 'Status' (dropdown menu with 'Livre' selected), and a 'Cadastrar' (Register) button.

Lista de Veículos: This section displays a table with the following data:

Código	Modelo	Cor	Ano	Placa	Tipo	Status	Ação
1	Bravo	Vermelho	2012	NXZ 2653	Hach Médio	Livre	Excluir
4	Titan 125	Vermelho	2015	FHG 3432	Motocicleta	Alugado	Excluir

Figura 1 – Página Veículos.

Fonte: Autor.

Descrição: a figura apresenta como deverá ser a página de Veículos. Ela possui um formulário para cadastro de um novo veículo e uma tabela contendo a lista de todos os veículos cadastrados no banco de dados. Para cada linha da tabela há um botão para excluir o registro desejado.



- **Página de Locações:** esta página deverá possuir um formulário para preenchimento dos campos: **Data da Locação** (date), **Data da Devolução**, **Status** (combobox com os valores: *Aberta*, *Fechada* e *Cancelada*), **Funcionário** (combobox com os nomes dos funcionários cadastrados no banco de dados), **Cliente** (combobox com os nomes dos clientes cadastrados no banco de dados). O combobox **Veículo** deverá listar apenas os veículos com **status Livre** (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 listadas todas as locações cadastradas. Um botão para excluir deverá ficar ao lado de cada locação listada. 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.

Figura 2 – Página Locações.

Fonte: autor.

Descrição: a figura apresenta como deverá ser a página de Locações, esta possui um formulário para cadastro de uma nova locação e uma tabela contendo a lista de todas as locações cadastradas no banco de dados. Para cada linha da tabela, há 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	
1	Gerente	Responsável por tudo	R\$ 1300,55	Excluir
2	Secretária	Responsável pelo atendimento ao público	R\$ 950	Excluir

Figura 3 – Página Cargos.

Fonte: autor.

Descrição: a figura apresenta como deverá ser a página de cargos. Esta possui um formulário para cadastro de um novo cargo e uma tabela contendo a lista de todos os cargos cadastrados no banco de dados. Para cada linha da tabela há 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 Admissão** (date) 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. 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.



Home Véículos Locações Clientes **Funcionários** Cargos Eliomar Campos

Locadora de Veículos

AQUILOCA

Novo Funcionário

Nome:

CPF:

Data Admissão: dd/mm/aaaa

Cargo: Gerente

Cadastrar

Lista de Funcionários

Código	Nome	CPF	Data Admissão	Cargo	Ações
1	Eliomar Gomes Campos	00000000000	01/01/2017	Gerente	Excluir
2	Mayara Gomes Campos	00000000000	04/09/2017	Secretária	Excluir

Figura 4 – Página Funcionários.

Fonte: autor.

Descrição: a figura apresenta como deverá ser a página de funcionários. Ela possui um formulário para cadastro de novos funcionários e uma tabela contendo a lista de todos os que já estão cadastrados no banco de dados. Para cada linha da tabela há um botão para excluir o registro desejado.

- **Página de Cliente:** esta página deverá possuir um formulário para preenchimento dos campos: **Nome** (text), **CPF** (text), **Status** (combobox com os valores: *Ativo*, e *Bloqueado*) e **Pontos Fidelidade** (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 Clientes cadastrados. Um botão para excluir deverá ficar ao lado de cada Cliente 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 5** apresenta tais características dessa página.



Home Veículos Locações **Clients** Funcionários Cargos Eliomar Campos

Locadora de Veículos

AQUILOCA

Novo Cliente

Nome:

CPF:

Pontos Fidelidade:

Status:

Lista de Clientes

Código	Nome	CPF	Status	Pontos Fidelidade	
1	José Francisco	000000000000	Ativo	4896	<input type="button" value="Excluir"/>
2	Maria José	000000000000	Bloqueado	936854	<input type="button" value="Excluir"/>

Figura 5 – Página Clientes.

Fonte: autor.

Descrição: a figura apresenta como deverá ser a página de clientes. Ela possui um formulário para cadastro de um novo cliente e uma tabela contendo a lista de todos os clientes cadastrados no banco de dados. Para cada linha da tabela há 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**). Deverá ter ainda 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 “pais” (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 “pais” cadastrados. Um botão para excluir deverá ficar ao lado de cada “pai” 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.



Home Veículos Locações Clientes Funcionários Cargo Eliomar Campos

Locadora de Veículos

AQUILOCA

Nome Completo

Primeiro Nome

Último Nome

 Cadastrar

Lista de Parentes

Código	Primeiro Nome	Último Nome	
1	Eliomar	Campos	
3	Rafaela	Campos	

Figura 6 – Página Detecção de Plágio.

Fonte: 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. Há um botão para excluir o registro desejado para cada linha da tabela.

1.4 Arquitetura

O projeto já está iniciado com a organização desejada. Utilizando **PHP Orientado a Objetos**, você deve seguir os 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: **Veículos**, **Clientes**, **Locações** 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
			
Páginas responsáveis apenas pela forma como a informação são apresentadas ao usuário, ou seja, é o design do website, compõe arquivos de layout HTML, CSS e imagens Exemplo de arquivos: index.php view_funcionario.php view_cargo.php	Páginas responsáveis apenas por receber dados provenientes das páginas da view, e encaminhá-los para os métodos das classes de persistência. Exemplo de arquivos: incluir_funcionario.php excluir_funcionario.php incluir_cargo.php excluir_cargo.php	Classes que contém os métodos de acesso e manipulação dos dados no banco de dados. Por exemplo, cadastrar, remover, alterar e listar. Exemplo de arquivos: dao_funcionario.php dao_cargo.php conexao.php	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. Exemplo de arquivos: class_funcionario.php class_cargo.php

Tabela 1 – Arquitetura em camadas.

Fonte: autor

Descrição: a figura apresenta a estrutura de pastas de acordo com os padrões de projetos 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 disponibilizada pelo professor, você poderá estudar tais páginas 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 funcionários e cargos. 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 funcionário e cargo disponibilizados, para conseguir adicionar as demais, seguindo o mesmo raciocínio e padrão.

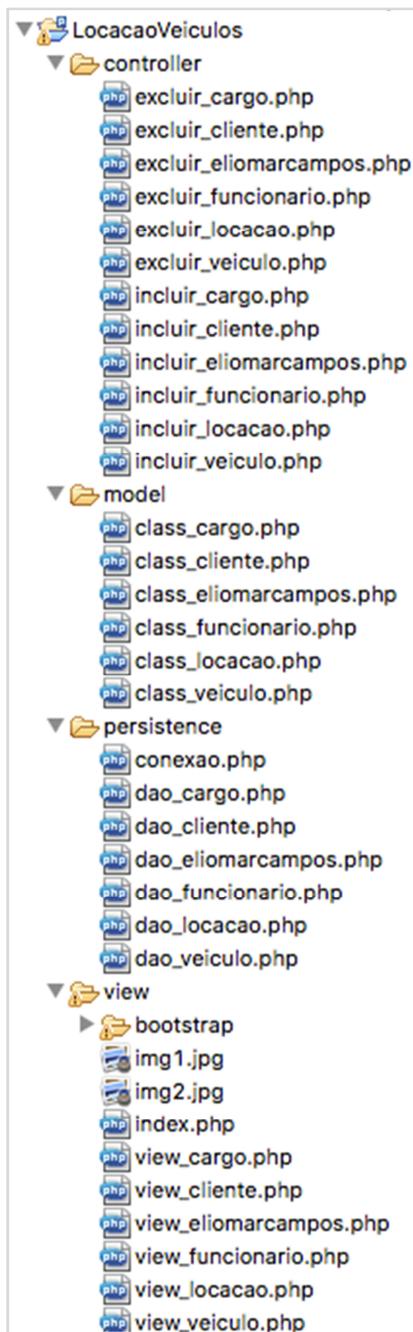


Figura 7 – Arquitetura em camadas do projeto completo.

Fonte: 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.

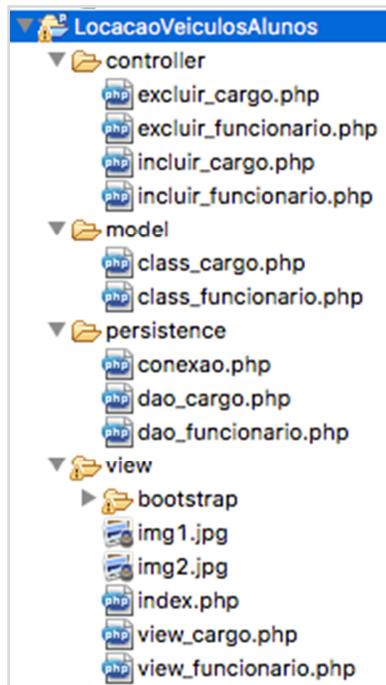


Figura 8 – Arquitetura em camadas do projeto incompleto.

Fonte: 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:

a) Nome da base de dados: **aluguelveiculos**

b) Tabela **veiculo**:

- `veiculo_id` [BIGINT, NOT NULL, AUTO_INCREMENT, e PRIMARY KEY]
- `veiculo_modelo` [VARCHAR(100)]
- `veiculo_cor` [VARCHAR(45)]
- `veiculo_ano` [VARCHAR(45)]
- `veiculo_placa` [VARCHAR(45), NOT NULL]
- `veiculo_tipo` [VARCHAR(45)]



- `veiculo_status` [VARCHAR(45), NOT NULL]

c) Tabela **cliente**:

- `cliente_id` [BIGINT, NOT NULL, AUTO_INCREMENT, e PRIMARY KEY]
- `cliente_nome` [VARCHAR(100), NOT NULL]
- `cliente_cpf` [VARCHAR(11), NOT NULL]
- `cliente_status` [VARCHAR(45)]
- `cliente_pontos_fidelidade` [BIGINT]

d) Tabela **funcionario**:

- `funcionario_id` [BIGINT, NOT NULL, AUTO_INCREMENT, e PRIMARY KEY]
- `funcionario_nome` [VARCHAR(100), NOT NULL]
- `funcionario_cpf` [VARCHAR(11), NOT NULL]
- `funcionario_data_admissao` [DATETIME]
- `funcionario_cargo` [VARCHAR(45)]

e) Tabela **locacao**:

- `locacao_id` [BIGINT, NOT NULL, AUTO_INCREMENT, e PRIMARY KEY]
- `locacao_data` [DATETIME, NOT NULL]
- `locacao_data_devolucao` [DATETIME, NOT NULL]
- `locacao_status` [VARCHAR(45), NOT NULL]
- `locacao_funcionario_id` [BIGINT, NOT NULL, e FOREIGN KEY REFERENCES funcionario]
- `locacao_veiculo_id` [BIGINT, NOT NULL, e FOREIGN KEY REFERENCES veiculo]
- `locacao_cliente_id` [BIGINT, NOT NULL, e FOREIGN KEY REFERENCES cliente]



f) Tabela **cargo**:

- `cargo_id` [BIGINT, NOT NULL, AUTO_INCREMENT, e PRIMARY KEY]
- `cargo_nome` [VARCHAR(45), NOT NULL]
- `cargo_descricao` [VARCHAR(100)]
- `cargo_salariobase` [DOUBLE]

g) Tabela **eliomarcampos**: aqui deverá ser o seu primeiro e último nome.

- `eliomarcampos_id` [BIGINT, NOT NULL, AUTO_INCREMENT, e PRIMARY KEY]
- `eliomarcampos_primeiro_nome` [VARCHAR(45)]
- `eliomarcampos_ultimo_nome` [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:

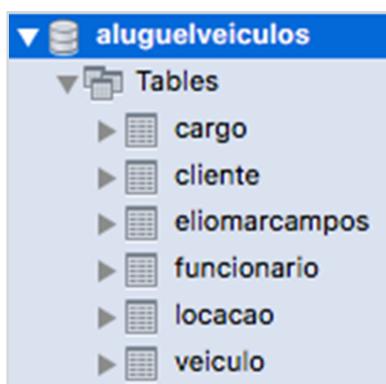


Figura 9 – Estrutura do banco de dados.

Fonte: 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. Porém, recomendamos fortemente uma dessas duas citadas.
- **XAMPP:** 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.

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_cliente.php, class_locacao.php, class_veiculo.php e class_eliomarcampos.php (este deverá ser com o seu nome).
DAO ou Persistence	Serão analisadas as classes de persistência ou de 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_cliente.php, dao_locacao.php, dao_veiculo.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, pois 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_cliente.php e excluir_cliente.php, incluir_locacao.php e excluir_locacao.php, incluir_veiculo.php e excluir_veiculo.php, e incluir_eliomarcampos.php e excluir_eliomarcampos.php (estes dois últimos deverão ser com o seu nome).
View	Serão analisadas as páginas de design e interação com o usuário, mais conhecidas como front-end. Tais páginas serão responsáveis também por enviar os dados de formulários para as classes de controle. Serão analisados os arquivos: view_cliente.php, view_locacao.php, view_veiculo.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.

1.7 Arquivos para Envio

O projeto deve ser enviado compactado em .ZIP ou .RAR. Lembre-se de que o 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 deverão 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.

