



**Universidade Federal de Uberlândia**  
**Faculdade de Computação – Prof. Daniel A Furtado**  
**Projeto Final de Programação para Internet**

Leia este documento até o fim, **sem pular uma única linha**, antes de implementar **uma única linha**! ☺

### INTRUÇÕES GERAIS

- Esta atividade deve ser realizada em equipes de no máximo 3 alunos;
- Tecnologias permitidas: HTML, CSS, JavaScript, Bootstrap 5, PHP, Ajax, JSON e MySQL;
- **Não é permitido o uso** de tecnologias adicionais como jQuery, React, Node.js, Angular, etc.
- Projetos utilizando outras tecnologias serão anulados;
- Esteja atento às **observações sobre plágio** apresentadas no final deste documento;
- Projetos com implementações utilizando trechos de códigos retirados de sites da Internet ou retirados de projetos de semestres anteriores, serão anulados;
- O website não deve conter qualquer conteúdo de caráter imoral, desrespeitoso, pornográfico, discurso de ódio, desacato, etc.;
- O website deve ser bem estruturado utilizando uma pasta raiz e subpastas de acordo com as instruções das videoaulas sobre HTML;
- O projeto deve ser entregue até a data definida pelo professor em aula síncrona;
- O projeto deve ser apresentado ao professor por toda a equipe em data a ser agendada. A avaliação é individual. TODOS os membros da equipe precisam conhecer em detalhes TODOS os recursos implementados, seja de *front-end* ou *back-end*. Durante a apresentação, perguntas específicas poderão ser feitas de qualquer parte do projeto a qualquer membro da equipe e as respostas serão utilizadas como parte da avaliação individual dos membros;
- O website implementado deve ser responsivo (principalmente os formulários);
- O website implementado deve prevenir ataques do tipo *cross-site scripting* (XSS) e *SQL Injection*;
- A comunicação com o MySQL deve ser feita utilizando o **PHP Data Objects** (PDO) do PHP;
- Nos campos de formulário, não adicione campos extras além dos que são pedidos;
- Todas as requisições AJAX devem retornar os dados no formato JSON;
- As requisições AJAX devem utilizar o XMLHttpRequest ou a API Fetch (com ou sem *async/await*);
- O website deve ser implementado utilizando os recursos gratuitos (PHP, MySQL, etc.) do [awardspace.com](https://www.awardspace.com) e disponibilizados no endereço:

[www.subdominiocadastradoparaostrabalhos.exemplodedominio.com/clinica](http://www.subdominiocadastradoparaostrabalhos.exemplodedominio.com/clinica)

## Descrição Geral

Desenvolva um website para uma clínica médica. O website deve ser organizado em duas partes:

- uma de acesso restrito, voltada para a equipe interna da clínica;
- e outra voltada para o público em geral.

Todas as páginas do website devem possuir um layout contendo quatro partes bem definidas: um cabeçalho, uma barra de navegação, um rodapé e uma parte principal para exibição de conteúdo.

A parte do website voltada para o público em geral deve conter:

1. Uma página principal para exibição das informações da clínica (Home);
2. Uma página para exibição de fotos da clínica (Galeria - utilize fotos fictícias);
3. Uma página para cadastro de endereços (Novo Endereço);
4. Uma página de login;
5. Uma página para que os clientes possam realizar o agendamento de consultas médicas (Agendamento).

A parte do website de acesso restrito deverá conter telas/páginas para:

1. Cadastramento de funcionários da clínica;
2. Cadastramento de pacientes;
3. Listagem dos funcionários cadastrados;
4. Listagem dos pacientes cadastrados;
5. Listagem dos endereços cadastrados;
6. Listagem de todos os agendamentos de consultas realizados pelos clientes;
7. Listagem dos agendamentos de consultas apenas do funcionário logado caso ele seja um médico.

## 1 Parte de Acesso Público

---

A parte de acesso público deve possuir uma barra principal de navegação contendo links para acesso às páginas descritas a seguir e um botão/link que possibilite ao usuário efetuar login e ter acesso à parte restrita do website.

### 1.1 Página Home

Cada grupo deverá criar um nome fictício para a clínica. A página principal do website deverá exibir, de maneira elegante e bem organizada, os seguintes dados sobre a clínica (fique à vontade para elaborar um texto fictício para cada item):

- Nome da clínica;
- Breve descrição;
- Seus Valores;
- Uma foto principal ou logomarca.

### 1.2 Página Galeria

Esta página deve apresentar fotos, logos, etc., relacionados à clínica médica. O layout é livre.

### 1.3 Página Novo Endereço

Deve apresentar um formulário para cadastro de endereço contendo os seguintes campos: **CEP**, **logradouro**, **cidade** e **estado**. Os dados devem ser inseridos adequadamente na tabela “Base de Endereços AJAX” do banco de dados, conforme esquema apresentado no final deste documento.

### 1.4 Página de Login

A página de login deverá exibir um pequeno formulário com os campos **e-mail** e **senha** para que os **funcionários** da clínica possam *logar* e ter acesso à parte restrita do website. O formulário de login deve ser apresentado de maneira bem estruturada e elegante (um exemplo é o formulário do exercício feito anteriormente no trabalho de CSS). **A validação dos dados de login deve ser feita em segundo plano, com AJAX.** Mensagens adequadas devem ser apresentadas caso os dados estejam incorretos (por meio da manipulação da estrutura DOM, sem que a página seja recarregada).

Para validar os dados de login utilize o campo **E-mail** da tabela **Pessoa** juntamente com o campo **SenhaHash** da tabela **Funcionário** (veja diagrama no final do documento).

### 1.5 Agendamento de Consulta

O procedimento de agendamento de consulta médica deve ser realizado explorando a tecnologia AJAX. O formulário de agendamento deve ser responsivo e deve se apresentar de maneira amigável, contendo os seguintes campos e funcionalidades:

**Especialidade médica desejada.** Campo do tipo **select** listando, dinamicamente, as especialidades disponíveis, *de acordo com os médicos correntemente cadastrados no banco de dados*. Não deve aparecer neste **select** uma especialidade médica sem que tenha um respectivo médico cadastrado.

**Nome do médico especialista.** Campo do tipo **select** que deverá ser carregado dinamicamente com AJAX depois que a especialidade médica for selecionada (uma requisição assíncrona deve buscar no servidor os nomes dos médicos cadastrados naquela especialidade);

**Data da consulta.** Campo do tipo **date** para que o usuário escolha a data do agendamento;

**Horário disponível para consulta.** Campo do tipo **select** para que o usuário escolha uma opção de horário disponível na data previamente selecionada. Os horários de agendamento devem ser exatos, no período das 8h às 17h (exemplos: 8, 9, 10, 11, etc.).

Quando a data da consulta for informada, uma requisição AJAX deverá buscar no banco de dados todos os horários já agendados para o médico em questão na data selecionada. Ao receber o retorno do servidor, um código JavaScript deverá filtrar as opções de horários eliminando os horários já agendados anteriormente e preenchendo o campo *select* com apenas os horários “disponíveis” (para simplificar, basta montar uma lista com todos os inteiros no intervalo de 8 a 17 e posteriormente remover os números “ocupados” de acordo com a lista de horários ocupados retornados pela requisição AJAX).

O formulário de agendamento deve possuir também uma região para que o paciente possa informar os seus dados pessoais essenciais: **nome**, **e-mail** e **sexo**.

Os dados do agendamento devem ser armazenados de maneira adequada em uma tabela de nome **Agenda** do banco de dados, conforme apresentado no esquema de dados no final deste documento. Para fins de simplificação, os dados do usuário que está realizando o agendamento (nome, e-mail e sexo) devem ser armazenados na própria tabela **Agenda**. Suponha que uma equipe interna da clínica posteriormente fará o cadastro formal do paciente utilizando a respectiva funcionalidade da parte restrita do website.

## 2 Parte Restrita do Website

---

A parte pública do website deve exibir um botão/link **Login** na barra de navegação (à direita). Quando o usuário clicar nesse botão e efetuar login com sucesso, o website deverá abrir, em uma nova aba, a página principal da parte restrita do sistema.

A parte restrita do website deverá ter uma barra de navegação diferente daquela feita para a parte pública, pois deverá exibir as opções:

- **Novo Funcionário**
- **Novo Paciente**
- **Listar Funcionários**
- **Listar Pacientes**
- **Listar Endereços**
- **Listar todos Agendamentos**
- **Listar meus Agendamentos** (esta última opção deverá aparecer apenas caso o funcionário logado seja um médico).

### 2.1 Cadastro de Funcionário

Um formulário adequado deve ser exibido para cadastrar os seguintes dados básicos de todos os funcionários: **nome**, **sexo**, **email**, **telefone**, **CEP**, **logradouro**, **cidade**, **estado**, **data de início do contrato de trabalho**, **salário** e **senha**.

Deve haver também uma opção para indicar se o funcionário é um médico. Caso o funcionário seja um médico, também devem ser cadastrados os dados: **Especialidade** e **CRM**. A página de cadastro deve exibir os campos **Especialidade** e **CRM** apenas quando o usuário escolher a opção “funcionário médico”. Esses campos devem ficar inicialmente ocultos e devem ser apresentados dinamicamente utilizando JavaScript quando essa opção for selecionada.

Os dados devem ser inseridos por um script PHP nas tabelas **Pessoa**, **Funcionário** e **Médico** do banco de dados utilizando o conceito de generalização/especialização, com as devidas chaves primárias e estrangeiras. A operação deve ser feita como uma **transação** no banco de dados utilizando o conceito de commit/rollback.

O preenchimento dos campos de endereço do funcionário deve ser facilitado utilizando AJAX. Assim que o usuário preencher o CEP, uma requisição AJAX deve buscar no **próprio** servidor, na tabela “Base de Endereços AJAX”, os demais dados do endereço relativo ao CEP indicado (logradouro, cidade e estado). Um código JavaScript deverá completar os campos do formulário automaticamente conforme

dados retornados pela requisição. Se o CEP não estiver cadastrado no servidor, os campos não precisam ser preenchidos automaticamente.

**OBS 1:** esta funcionalidade deve ser implementada. Serviços/APIs de busca de endereço de terceiros não devem ser utilizadas. Caso seja, o recurso será desconsiderado na avaliação do trabalho.

**OBS 2:** os dados do endereço do funcionário precisam ser inseridos normalmente na tabela **Pessoa**. Não faça qualquer ligação com os dados de endereço cadastrados na tabela “Base de Endereços AJAX”. Esta tabela é apenas auxiliar e tem o único objetivo de prover o serviço de busca de endereço pelo CEP para facilitar o preenchimento do formulário.

## 2.2 Cadastro de Paciente

A página de cadastro de paciente deve oferecer um formulário para cadastro das seguintes informações do paciente: **nome, sexo, email, telefone, CEP, logradouro, cidade, estado, peso, altura e tipo sanguíneo**.

Os dados devem ser inseridos adequadamente no banco de dados nas tabelas **Pessoa** e **Paciente** por um script PHP. A operação deve ser feita como uma **transação**.

## 2.3 Listagens dos Dados

A parte restrita deverá apresentar opções para que o funcionário possa visualizar, de maneira estruturada, os dados cadastrados. A listagem pode ser feita utilizando tabelas HTML ou o sistema de grade do Bootstrap. Deverá haver opções para as seguintes listagens de dados:

- Listagem dos funcionários cadastrados;
- Listagem dos pacientes cadastrados;
- Listagem dos endereços auxiliares (base de endereços AJAX);
- Listagem de todos os agendamentos de consultas realizados pelos clientes;
- Listagem dos agendamentos de consultas apenas do funcionário logado caso ele seja um médico.

## 2.4 Login e sessões PHP

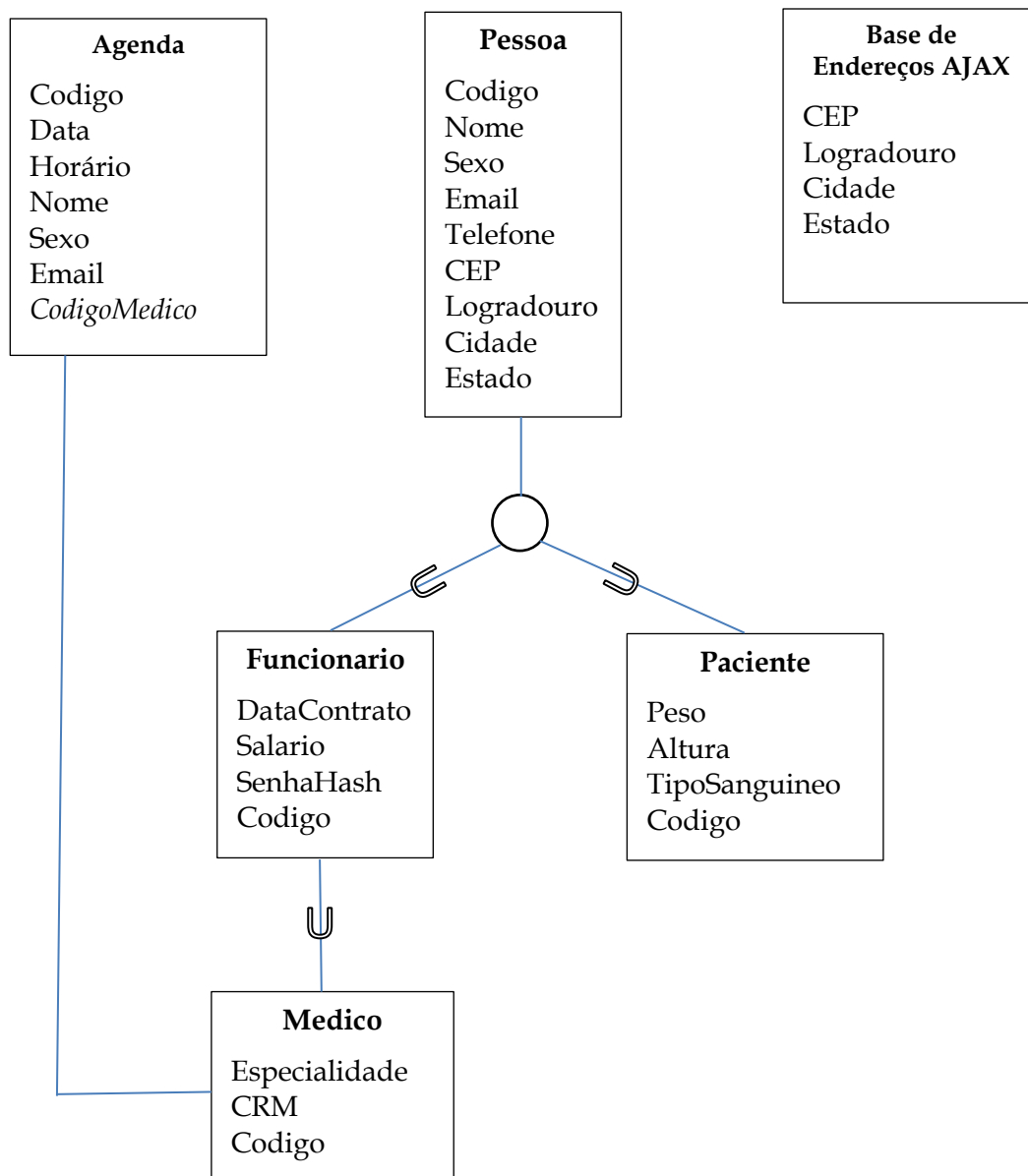
O sistema deverá implementar adequadamente o conceito de sessão em PHP para manter o usuário *logado* durante o acesso à parte restrita. O sistema não deverá permitir o acesso à scripts PHP da parte restrita sem que o usuário tenha permissão para tal (sem que esteja devidamente logado).

### 3 Banco de dados e tabelas

---

#### Observações/Dicas:

- O campo **Codigo** em **Pessoa** deve ser chave primária;
- O campo **Codigo** em **Funcionário** deve ser chave primária e também chave estrangeira conectando com **Pessoa**;
- O campo **Codigo** em **Medico** deve ser chave primária e também chave estrangeira conectando com **Funcionario**;
- O campo **Codigo** em **Paciente** deve ser chave primária e também chave estrangeira conectando com **Pessoa**;
- O campo **Codigo** em **Agenda** deve ser chave primária;
- O campo **CodigoMedico** em **Agenda** deve ser chave estrangeira conectando com **Medico**.



## 4 Outras Informações Importantes

---

### 4.1 Sobre Eventuais Plágios

Os alunos envolvidos em qualquer tipo de plágio, total ou parcial, seja entre equipes ou de trabalhos de semestres anteriores ou de materiais disponíveis na Internet (exceto os materiais de aula disponibilizados pelo professor), serão duramente penalizados (art. 196 do Regimento Geral da UFU). Todos os alunos envolvidos terão seus **trabalhos anulados** e receberão **nota zero**.

### 4.2 Outros critérios de avaliação:

- Otimização/eficiência das operações implementadas;
- Uso adequado dos recursos conforme solicitado neste documento e apresentado nos materiais de aula;
- Domínio dos conceitos e técnicas durante a apresentação oral do projeto;
- Uso adequado dos elementos semânticos da HTML;
- Atendimento à especificação da HTML5 (páginas validadas em **validator.w3.org**);
- Usabilidade, elegância, segurança, responsividade e robustez do website;
- Cumprimento dos prazos.

## 5 Entrega

---

O sistema deve ser colocado online até a data de entrega limite especificada pelo professor nas aulas síncronas. Além da disponibilização online, **também é necessário** compactar a pasta raiz do sistema web (contendo todos os arquivos e recursos necessários para o seu funcionamento tais como arquivos HTML, CSS, JavaScript, imagens, scripts PHP, etc.) e enviar o arquivo compactado, até a data limite de entrega, pelo sistema SAAT do professor ([www.furtado.prof.ufu.br](http://www.furtado.prof.ufu.br)).

Caso o arquivo compactado possua mais de 10 MB de tamanho, poderá ser necessário reduzir a resolução das imagens para que esse limite não seja ultrapassado.

A equipe também deverá incluir na pasta raiz do sistema um arquivo de texto contendo o código SQL que permita criar todas as tabelas do banco de dados para uso com o sistema.