



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

## 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 (clientes).

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 para exibição de conteúdo. Um exemplo de *layout* é apresentado em [www.w3schools.com/html/html\\_layout.asp](http://www.w3schools.com/html/html_layout.asp) (não é obrigatório o uso deste *layout* - é apenas um exemplo).

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

1. Uma página principal para exibição das informações da clínica (Home);
2. Uma página para exibição de fotos e vídeos da clínica (Galeria);
3. Uma tela para que os clientes possam contatar a clínica, deixando suas mensagens tais como sugestões, perguntas, etc. (Contato);
4. Uma página para que os clientes possam realizar o agendamento de consultas médicas (Agendamento).

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

1. Cadastramento de funcionários da clínica;
2. Listagem dos contatos realizados pelos clientes (sugestões, perguntas, etc.);
3. Listagem dos agendamentos de consultas realizados pelos clientes;
4. Listagem dos funcionários cadastrados.

Recomenda-se a leitura completa deste documento antes de iniciar o desenvolvimento do website.

## Recursos e Tecnologias

---

O website deve ser desenvolvido utilizando as seguintes tecnologias: HTML5, CSS, JavaScript, JSON, jQuery, Bootstrap, AJAX, PHP e MySQL.

## Hospedagem

---

O website deve ser hospedado gratuitamente nos servidores do *awardspace* ([www.awardsapace.com](http://www.awardsapace.com)) conforme apresentado nos slides de aula. Cada grupo deve criar uma conta no *awardsapace.com* e utilizá-la para enviar os arquivos do website, criar e manipular o banco de dados.

## Observação Importante

---

Os alunos envolvidos em qualquer tipo de plágio, total ou parcial, **entre equipes**, de trabalhos feitos em 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**.

## Organização dos Arquivos

---

Os arquivos do website devem ser bem organizados e estruturados. Arquivos de imagem, código JavaScript, CSS, etc., devem ser armazenados em pastas próprias. O website deve ser desenvolvido visando uma manutenção prática e eficiente.

## Parte Pública do Website – Especificações Detalhadas

---

O website deve possuir uma barra de navegação com opções para acessar os conteúdos: Home, Galeria, Contato e Agendamento. Utilize os recursos adequados do *framework* Bootstrap para criar a barra de navegação. Explore os recursos da CSS (criando classes de estilos, etc.) e da biblioteca jQuery para prover uma melhor apresentação das páginas.

### Página Home (acesso público)

Cada grupo deverá criar um nome para a clínica fictícia. 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;
- Sua missão;
- Seus Valores;
- Uma foto principal ou logomarca.

Utilize jQuery para possibilitar que o conteúdo de algum desses tópicos seja ocultado e reexibido quando o usuário clicar sobre o mesmo. Utilize um efeito de sua preferência.

### Página Galeria (acesso público)

Nesta página deverão ser exibidas pelo menos 8 imagens em formato de tabela (2 linhas, 4 colunas) e um vídeo do YouTube:

- Utilize a biblioteca jQuery para fazer com que as imagens da galeria sejam apresentadas progressivamente quando a página for carregada.
- Utilize JavaScript (puro, sem jQuery) para que, ao passar o mouse sobre qualquer imagem da galeria, uma borda vermelha envolvendo a imagem seja apresentada para destacá-la. Quando o usuário retirar o ponteiro do mouse da imagem, a borda sobre aquela imagem deve desaparecer (será necessário criar funções JavaScript e utilizá-las em conjunto com os atributos de eventos HTML **onmouseenter** e **onmouseleave**).

## Página Contato (acesso público)

Deve exibir um formulário com *layout* vertical (utilize Bootstrap) para que o usuário do website deixe sua mensagem para a clínica. O formulário deve ter campos apropriados para coletar os seguintes dados:

- Nome do cliente;
- E-mail do cliente;
- Motivo do contato (Reclamação/Sugestão/Elogio/Dúvida)
- Mensagem.

Os dados coletados devem ser adequadamente armazenados em uma tabela de nome **Contato** no banco de dados do servidor. Utilize os atributos de validação de formulários da linguagem HTML para exigir o preenchimento de todos os campos. O formulário deverá se apresentar de maneira semelhante ao formulário a seguir:



Envie sua mensagem

Nome

E-mail

Mensagem

Enviar

## Página Agendamento (acesso público)

Deve possibilitar o agendamento on-line de consultas médicas, permitindo que o usuário indique:

- a especialidade médica para a qual deseja marcar a consulta;
- um médico daquela especialidade (que tenha sido previamente cadastrado);
- a data da consulta;
- e um horário disponível para a consulta.

O procedimento de agendamento deve ser realizado explorando a tecnologia AJAX. Assim que o usuário selecionar a especialidade médica, o website deve buscar no servidor os nomes de todos os

médicos cadastrados que possuem aquela especialidade (veja a seção **Cadastro de Funcionário** deste documento). Com base nos dados retornados, a página deve atualizar automaticamente um campo do tipo **select** para que sejam apresentados ao usuário todos os nomes dos médicos daquela especialidade. Por fim, quando o médico for escolhido, o website deve apresentar a lista de horários disponíveis daquele médico na data indicada (se houver).

Tal procedimento deve ser desenvolvido de acordo com as seguintes especificações:

- a) O formulário de agendamento de consulta deverá se apresentar de forma elegante e amigável (utilize Bootstrap) e deve possuir os seguintes campos:
  - **Especialidade médica desejada.** Campo do tipo **select** listando as especialidades disponíveis *de acordo com os médicos cadastrados no banco de dados*;
  - **Nome do médico especialista.** Campo do tipo **select** que deverá listar os nomes dos médicos da especialidade previamente escolhida (de acordo com o banco de dados);
  - **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: 8:00, 9:00, 10:00, 11:00, etc.);
  - **Nome do paciente.** Campo textual para que o paciente informe o seu nome;
  - **Telefone do paciente.** Campo textual para que o paciente informe seu telefone.
- b) Os dados do agendamento devem ser armazenados de maneira adequada em uma tabela de nome **Agenda** do banco de dados. A tabela deve conter as colunas: *codAgendamento*, *data*, *hora* (inteiro), *codFuncionario* e *codPaciente*. Dicas:
  - O campo *codAgendamento* deverá ser a chave primária;
  - Os campos *data*, *hora* e *codFuncionario*, juntos, devem compor uma chave do tipo UNIQUE (não se deve permitir duas consultas agendadas para o mesmo dia, horário e médico);
  - O campo *codFuncionario* deve ser uma chave estrangeira conectando com **Funcionário**;
  - O campo *codPaciente* deve ser uma chave estrangeira conectando com a tabela **Paciente** (veja a dica a seguir);
- c) Os dados do paciente em si (código, nome e telefone) não devem ser armazenados na tabela **Agenda**. Eles devem ser armazenados separadamente em uma tabela de nome **Paciente**;
- d) Quando a especialidade médica for selecionada pelo usuário, uma requisição AJAX deverá buscar no banco de dados do servidor os nomes dos médicos daquela especialidade. Utilize o evento **onchange** do elemento **select** para disparar a requisição AJAX. Ao receber esses nomes do servidor, utilize JavaScript/DOM para montar dinamicamente a lista de opções contendo os nomes dos médicos (campo **select**). Um novo item de um campo **select** pode ser adicionado dinamicamente utilizando o próprio JavaScript. Veja o exemplo a seguir:

```
var campoSelect = document.getElementById("idDoCampoSelect");
var option = document.createElement("option");
option.text = "Texto da nova opção";
option.value = "Value da nova opção";
campoSelect.add(option);
```

- e) Quando o nome do médico for selecionado pelo usuário, uma requisição AJAX deverá buscar no banco de dados todos os horários já agendados para o médico na data especificada pelo usuário.

Ao receber o retorno do servidor, um código JavaScript deverá filtrar as opções de horários, eliminando os horários já agendados anteriormente. O código JavaScript deverá preencher adequadamente o campo “*Horário disponível para consulta*” de tal forma que o usuário visualize apenas os horários livres.

- f) Utilize o conceito de transação ao realizar a inserção dos dados do agendamento nas duas tabelas do banco de dados (Agenda e Paciente);

## Parte Restrita do Website – Especificações Detalhadas

---

A parte pública do website deve exibir um botão **Login** na barra de navegação (à direita). Quando o usuário clicar nesse botão, um pequeno formulário com os campos *login* e *senha* deverá ser apresentado através de um *modal* (utilize Bootstrap). Caso os valores informados sejam válidos, o website deverá abrir, em uma nova aba, a página principal da parte do sistema de acesso restrito. Para fins de simplificação, a validação do *login* e *senha* pode ser feita através de uma simples consulta na tabela **Usuário** (com dois campos, login e senha). Não é necessário criar uma interface para cadastro de usuários. Os dados podem ser inseridos manualmente na tabela utilizando SQL.

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**, **Listar Funcionários**, **Listar Contatos**, **Listar Agendamentos**.

### Cadastro de Funcionário

Esta página deve exibir um formulário para cadastramento dos funcionários da clínica. O formulário deve se apresentar de maneira organizada e elegante (com layout horizontal, utilizando Bootstrap) e deve ter três partes (Dados pessoais, Documentos e Endereço). Os seguintes dados devem ser coletados:

#### Dados pessoais:

- Nome do funcionário;
- Data de nascimento;
- Sexo: campo do tipo *radio*;
- Estado civil: caixa de seleção;
- Cargo: caixa de seleção com as opções *médico*, *enfermeiro*, *secretário*, *outro*.
- Especialidade médica: esta opção deverá aparecer apenas quando o usuário selecionar *médico* para o campo *Cargo* (utilize CSS com JavaScript/jQuery).

#### Documentos:

- CPF
- RG;
- Outro;

#### Endereço:

- |   |                |
|---|----------------|
| • CEP;                                  | • Complemento; |
| • Tipo de logradouro (rua/avenida/prça) | • Bairro;      |
| • Logradouro;                           | • Cidade;      |
| • Número;                               | • Estado;      |

Os dados coletados devem ser adequadamente armazenados no banco de dados por um script PHP. A tecnologia AJAX deve ser utilizada para facilitar o preenchimento do endereço. Assim que o usuário digitar o último caractere do CEP, os campos *logradouro*, *bairro* e *cidade* devem ser automaticamente preenchidos (uma requisição AJAX deve ser feita ao próprio servidor para resgatar tais dados de uma tabela do banco de dados).

Utilize JavaScript para validar a data de nascimento do funcionário. O formulário não deve ser submetido quando a data de nascimento for maior do que a data atual.

O endereço do funcionário deve ser armazenado em tabela separada dos demais dados do funcionário (criar uma tabela **EnderecoFunc** no banco de dados e fazer o devido vínculo com a tabela **Funcionario** utilizando uma chave estrangeira);

Os dados devem ser inseridos no banco de dados nas respectivas tabelas (**Funcionario** e **EnderecoFunc**) utilizando o conceito de transação.

## Listagem dos Funcionários

Os seguintes dados de todos os funcionários cadastrados devem ser apresentados na forma de uma tabela. Utilize PHP e Bootstrap para que os dados sejam apresentados de maneira elegante e organizada.

- Nome
- Sexo
- Cargo
- RG
- Logradouro
- Bairro
- Cidade

## Listagem dos Contatos

Os dados de todos os contatos feitos pelos usuários devem ser apresentados na forma de uma tabela.

## Listagem dos Agendamentos

Listar, para todos os agendamentos realizados, os seguintes dados:

- O nome e a especialidade do médico que prestará o atendimento;
- A data e a hora da consulta;
- O nome e o telefone do paciente que agendou a consulta.

Os agendamentos devem ser listados por ordem alfabética do nome do médico e, posteriormente, pela data das consultas.

## Criação de Domínio

---

Cada grupo deverá registrar um domínio comercial para o website desenvolvido. Há domínios a partir de R\$ 6,99 no site *GoDaddy*. Após registro do domínio comercial, as devidas configurações devem ser

feitas (no próprio site de registro de domínio) para que o usuário seja redirecionado para o endereço do subdomínio gratuito criado no *awardspace.com*.

## Outras Especificações e Restrições

---

O website deve ser responsável. Ele deve se apresentar de maneira adequada e legível mesmo quando acessado a partir de dispositivos móveis;

- *Templates* e/ou *layouts* prontos não devem ser utilizados;
- Outros *frameworks* e tecnologias não devem ser utilizados;
- Os formulários devem ser validados no servidor para prevenir ataques do tipo *HTML/SQL Injection*;
- Os códigos comuns referentes aos cabeçalhos, barras de navegação e rodapés das páginas não devem ser explicitamente repetidos para cada página. Há diversas formas de compartilhar esses elementos com todas as páginas que os utilizam. Uma delas é utilizando a declaração *PHP include*. Veja o exemplo a seguir:

```
<!DOCTYPE html>
<html>
<head>
<title></title>
<body>

<?php include "header.php"; ?>
<?php include "navbar.php"; ?>

<div id="conteudoDaPagina">

</div>

<?php include "footer.php"; ?>

</body>
</html>
```

## Da Entrega e Avaliação das Atividades

---

O projeto deverá ser entregue em duas etapas:

1. Na primeira etapa é preciso entregar toda a parte de *front-end* do website: páginas de acesso público e páginas restritas, formulários, galeria de imagens, etc. Entretanto, nesta etapa não é necessário efetivar as operações propriamente ditas (não é necessário efetivar os cadastros, efetivar o agendamento, buscar horários livres e médicos, etc.). Também não é necessário realizar o *login* propriamente dito (embora a interface para *login* já tenha que existir). A pasta raiz do website deve ser compactada e enviada por e-mail ([furtado@ufu.br](mailto:furtado@ufu.br)) para o professor até o dia 26/09/2018.
2. Na segunda etapa é preciso implementar toda a parte de *back-end* do website, incluindo toda a lógica de programação *server-side* para efetivar as operações. Nesta etapa deve-se focar nas operações com o banco de dados, criação dos serviços para acesso com AJAX, efetivação do *login*,

uso de sessões, etc. O projeto completo deverá ser entregue e apresentado ao professor na data especificada no cronograma de atividades de aula. No momento da apresentação, os alunos deverão mostrar o sistema em funcionamento on-line, nos servidores do *awardspace*. O website será acessado por meio do domínio registrado.

**Valor:** 20 pontos.

**Alguns 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;
- Usabilidade, elegância, segurança e robustez do website;
- Cumprimento dos prazos.