

## **Trabalho - Aplicação Web**

### **Introdução**

Neste semestre faremos um trabalho único que será dividido em partes. Desde o início é importante que você escolha bem o objetivo do trabalho para que cada etapa venha a complementar a seguinte e, ao final, você tenha uma aplicação Web completa.

O trabalho irá contemplar os seguintes tópicos que serão vistos durante a disciplina:

- Framework Front End Web
- Controle de Sessão de usuário
- Abordagem AJAX
- Web Services REST para construção de APIs
- Protocolo Open Authorization (OAuth)
- Representação externa de dados com JSON
- Publicação de aplicações em ambiente de nuvem (Microsoft Azure)

A divisão do trabalho será feita da seguinte forma:

- Trabalho 1 (10 pontos)
  - Definição de um template a ser usado por todas as páginas do site
  - Aplicação de framework Frontend (Bootstrap, Materialize, Foundation)
  - Implementação e teste de responsividade com layout sendo compatibilizado com telas de celulares
- Trabalho 2 (10 pontos)
  - Implementação de Web Services REST fornecendo recursos mantido no banco de dados
  - Aplicação da abordagem Ajax em pelo menos uma tela de cadastro
  - Inclusão de funcionalidade de login com controle de sessão de usuário
- Trabalho Final (20 pontos)
  - Integração da aplicação com alguma rede social/provedor de dados via Open Authorization (OAuth)
  - Implementação dos demais cadastros e funcionalidades da aplicação
  - Implementação do controle de segurança para as funcionalidades da aplicação

### **Instruções gerais**

- Os grupos são limitados a 4 pessoas
- As entregas devem ser feitas no SGA e no ambiente do Microsoft Azure

## Requisitos e restrições

O trabalho deverá entregar uma solução completa para um problema a escolha do grupo no contexto apresentado pelo Prof. William Regis (Nutrição) com um conjunto de requisitos técnicos mínimos apresentados a seguir. A solução deverá fazer sentido não podendo ser apenas um conjunto de cadastros de tabelas sem uma utilidade prática real.

### Funcionais

- 1) Controle de acesso a usuários por perfil
  - a. Cada usuário deve ser identificado como, pelo menos, administrador ou usuário final
  - b. O sistema deve verificar o perfil do usuário logado para conceder acessos diferentes em função do perfil
- 2) Funcionalidade de login que permita a integração com pelo menos 1 das redes sociais (Facebook, Google, Twitter) ou provedor de recursos via OAuth.
- 3) Funcionalidades gerais envolvendo pelo menos 6 tabelas, a saber:
  - a. Cadastro de usuário com permissão apenas para o administrador
  - b. Outros 5 casos de uso envolvendo as demais tabelas contemplando cadastros, transações ou consultas que incluam pelo menos 2 filtros cada uma.
- 4) Menu de navegação com todas as funcionalidades do sistema incluídas.

### Não Funcionais

- 1) O sistema deverá estar hospedado no Microsoft Azure ou nuvem similar com acesso aberto para o professor durante todo o período da disciplina.
- 2) Estrutura de banco de dados contando com
  - a. Pelo menos 6 tabelas de entidades principais
  - b. Pelo menos outras 3 tabelas secundárias de relacionamentos e domínios
- 3) Serviços Web padrão REST a serem utilizados pelas interfaces de cadastro, transações e consultas.
- 4) Utilização da estratégia AJAX nas telas para dinamizar as operações do CRUD, as transações e os filtros das consultas.
- 5) Utilização de framework para o frontend (bootstrap, materialize ou foundation)
- 6) Layout responsivo, com adequação da aplicação para smartphones

## Entregas

As entregas do trabalho estão organizadas em 3 etapas, sendo 2 etapas de 10 pontos e uma final de 20 pontos.

### Trabalho 1

Nesta primeira parte, crie uma estrutura de pastas para o projeto conforme apresentado na Figura 1 e monte um layout padrão a ser utilizado em todo o site. Para esse propósito, é fornecido no SGA um **template padrão** como orientação de boas práticas a serem seguidas.

Este ambiente inicial deverá contemplar um cabeçalho, um menu de opções (já prevendo as diversas funcionalidades que o seu sistema terá), uma estrutura de corpo para o conteúdo principal e um rodapé. O layout deverá, ainda, ser

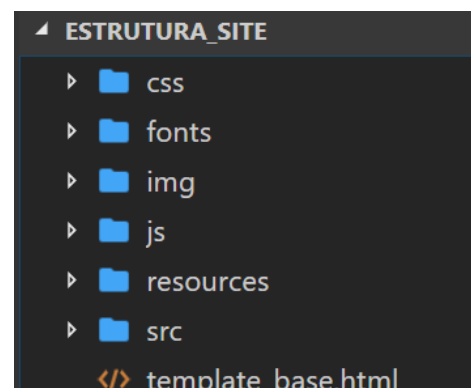


Figure 1 - Estrutura de pastas

responsivo e a visualização adequada tanto com ambientes desktop quanto dispositivos móveis.

Para a implementação da responsividade, é permitido que o grupo utilize no seu projeto um framework de frontend voltado para a estilização da aplicação, podendo ser o Bootstrap, o Foundation ou o Materialize, a escolha do grupo. Neste trabalho, o aluno deverá montar as telas de login, a tela construída na etapa 3 e mais uma outra tela utilizando as definições de estilo do framework.

O trabalho deverá ser publicado em um ambiente de nuvem, para o qual sugerimos o ambiente do Microsoft Azure que provê acesso gratuito a todos os alunos da PUC.

A relação que se segue, apresenta algumas referências importantes para estudo e consulta que vão auxiliar no desenvolvimento do trabalho 1. São elas:

- Document and website structure  
[https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction\\_to\\_HTML/Document\\_and\\_website\\_structure](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Document_and_website_structure)
- The Secrets of Semantic HTML5 for Document Structure — a Guide  
<https://www.semrush.com/blog/semantic-html5-guide>
- HTML Document Meta  
<https://html.com/document/metadata>
- HTML CheatSheet  
<https://htmlcheatsheet.com>

## Trabalho 2

Nesta etapa, você deverá criar um Web Service na sua aplicação que retorne dados em formato JSON a serem utilizados no cliente que deverá interagir por meio de requisições Ajax. A título de exemplo, em aplicações que lidam com clientes, pode ser entregue um Web Service que retorne a lista de clientes com base em alguns parâmetros como, por exemplo, cidade de residência. A tela a ser construída deverá solicitar a cidade de residência que será enviada na requisição. Assim que receber o retorno, em formato JSON, a tela deverá listar os clientes de alguma maneira.

Você poderá utilizar a tecnologia de backend que desejar. Um exemplo simples de como criar um Web Service utilizando o mini-framework Slim do PHP se encontra no endereço que se segue:

<https://github.com/rommelcarneiro/AulasWeb/tree/master/Lab-Slim>

Nesta etapa, vamos abordar, ainda, o controle de sessões e o acesso a banco de dados de forma simplificada. O grupo deverá desenvolver um conjunto de páginas que identifique se o aluno já efetuou o login, consultando dados de senha em um banco de dados (tabela usuário), e em caso positivo apresente uma página inicial. Caso o usuário ainda não tenha efetuado login, deverá apresentar uma tela para que se identifique.

## Trabalho Final (20 pontos)

Nesta etapa, a entrega deve contemplar aplicação completa com todos os requisitos especificados neste documento, incluindo o que já foi feito nos trabalhos 1 e 2. Verifique os

requisitos funcionais e não funcionais para compreender o que precisa ser feito e entregue. Nesta etapa, todas as funcionalidades deverão constar do trabalho.

O trabalho a ser entregue, deverá permitir ao usuário efetuar login na aplicação a partir de um provedor do protocolo Open Authorization (Facebook, Google Accounts, etc). Assim que o usuário fizer a autenticação no provedor OAuth, o sistema deve permitir acesso com as credenciais validadas junto ao provedor e registrar o usuário no banco de dados incluindo informações sobre o provedor utilizado e o e-mail do usuário. Para isto será necessário incluir estes campos na tabela de usuário.

O Open Authorization é aberto e permite a implementação de 4 fluxos distintos: (1) Código de Autorização, (2) Autorização Implícita, (3) Credenciais do Usuário, (4) Credenciais do Cliente. Cada fluxo corresponde a um ambiente e necessidade distintos. Neste projeto, adotamos o fluxo de código de autorização.

Para auxiliar na implementação desta parte do projeto, disponibilizo um exemplo do funcionamento do OAuth para que possam estudar e incorporar no seu projeto. O link para o exemplo é o que se segue:

<https://github.com/rommelcarneiro/AulasWeb/tree/master/Lab-OAuth>

O sistema deverá, ainda, implementar um controle de autorização de telas em que, dependendo do perfil do usuário ele, o sistema deve permitir ou não o acesso às funcionalidades. Deve ser implementado minimamente dois perfis: usuário e administrador.

Aproveite para fazer uma verificação completa da sua aplicação testando performance e problemas de compatibilidade

- Compatibilidade (W3C Markup Validator): <https://validator.w3.org/>
- Desempenho (Google Page Speed): <https://developers.google.com/speed/pagespeed/>