



# CSI477 – Sistemas para a WEB I

## Atividade Prática 1 – Entrega: 22/05/2018

Valor: 05 pontos

Prof. Fernando Bernardes de Oliveira

[fboliveira@ufop.edu.br](mailto:fboliveira@ufop.edu.br) – <https://github.com/fboliveira>

2018/01

### Instruções/Regras

- a) **Entrega: até 22/05/2018, 23:55 horas** (no horário oficial do Moodle/UFOP ☺).
- **Moodle:** poste o link (apenas) do **GitHub** com a atividade. Utilize o repositório criado pelo *GitHub Classroom*.
  - **GitHub:** código-fonte e afins.
- b) A atividade é **INDIVIDUAL**.
- c) *Para cada uma das regras desrespeitadas serão descontados 20% do total da atividade.*
- d) Em caso de caracterização de cópia entre alunos(as) e/ou da internet (sem a devida referência), **TODOS** os envolvidos terão **nota zero (0)**.
- e) Os dados devem ser **validados**, tanto do lado do **cliente** quanto do lado do **servidor**.
- f) Utilize algum *framework* para a interface, como o **Bootstrap**, e algum para agilizar o processo de validação do lado do cliente, como o **jQuery**.

### 1 Desenvolvimento: Sistema de Controle de Solicitações de Análises Laboratoriais

O objetivo deste sistema é controlar as solicitações de análises clínicas laboratoriais por pacientes. O diagrama do banco de dados é apresentado na Figura 1. Ele foi criado conforme as convenções definidas pelo **Laravel**. Todos os usuários da aplicação podem **visualizar** os procedimentos. Entretanto, para solicitar exames o paciente deve efetuar o acesso (*login*) para concluir a ação. O administrador do sistema pode cadastrar os procedimentos, listar os exames e os pacientes, além de ter um relatório totalizando os exames solicitados por procedimentos/paciente. Crie **um menu de acesso** na página inicial para cada área e para cada uma das opções (*Área Geral*, *Pacientes* e *Administrador*). Desenvolva os processamentos de acordo com a área de acesso/visão do sistema apresentadas a seguir.

O cadastro dos procedimentos (*procedures*) é gerenciado por usuários (*users*) do sistema. Os exames (*tests*) são solicitados pelos pacientes. Existem três tipos (*type*) para os usuários que devem ser observados. O **tipo 1** é o **administrador**, com permissão para realizar todas as operações CRUD (*Create*, *Read*, *Update* e *Delete*). Os procedimentos podem ser excluídos somente se não existirem

exames associados. O código do administrador (*id*) é associado na tabela de procedimentos (*procedures.user\_id*) para indicar o responsável pelo cadastro. Isso permite que outros administradores sejam criados. O **tipo 2** é **operador**, o qual tem permissão apenas para listar os exames e alterar apenas o preço. E, por final, o **tipo 3** é **paciente**, o qual solicita exames.

Crie interfaces personalizadas por meio de CSS para garantir uma experiência amigável ao usuário. Você pode, por exemplo, utilizar o *Bootstrap* de maneira apropriada para agilizar o processo, além de utilizar *templates* livres disponíveis na Internet.

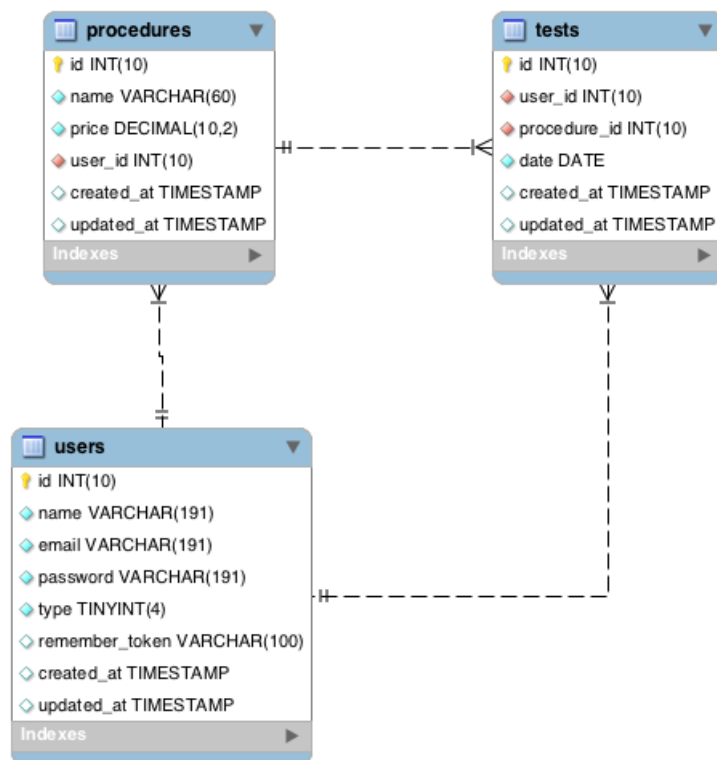


Figura 1 – Diagrama do Sistema de Controle de Análises Laboratoriais

### 1.1 Banco de Dados:

- O arquivo **analyses-v20180511.sql** contém o script para o banco de dados **analyses**.
- Não existem registros nas tabelas. Insira dados aleatórios, por exemplo, para facilitar os testes da aplicação.
- A tabela **users** armazena todos os tipos de usuários. Os procedimentos são armazenados na tabela **procedures** e os exames na tabela **tests**.
- Dados para conexão:

- *servidor*: localhost
- *usuário*: sysanalyses
- *banco*: analyses
- *senha*: 123456

### 1.2 Atividades

Crie o sistema conforme as áreas específicas:

### 1.2.1 Área Geral

Nesta área os **procedimentos** podem ser visualizados, sendo apresentados em ordem **ascendente** do **nome**.

### 1.2.2 Área do Paciente

A **Área do Paciente** é de acesso restrito. Assim, ele deve efetuar o *login* no sistema informando seu usuário e senha. Um novo paciente ainda tem a opção de se cadastrar. A área de solicitações **não pode** ser acessada se o cliente não estiver logado no sistema. Faça esse controle utilizando, por exemplo, sessão.

O paciente pode realizar solicitações de procedimentos, além de visualizar os procedimentos agendados. Para fazer a solicitação, o **paciente** seleciona o **procedimento** que deseja realizar. Ele pode solicitar mais de um procedimento, e não existem restrições para o número de solicitações.

As funcionalidades desta área são:

- a) **Novo paciente** – realizar cadastro de um novo paciente.
- b) **Acesso** à área do paciente – *login*.
- c) Inclusão de **solicitação de exames** – o paciente seleciona o procedimento e informa a data prevista para a realização. Ele pode ainda fazer a alteração e exclusão de procedimentos solicitados.
- d) Lista de **exames** solicitados em **ordem decrescente de data**, e os **nomes dos procedimentos em ordem alfabética**. A quantidade de exames e o valor total devem ser apresentados ao final do relatório.

### 1.2.3 Área Administrativa

A administração do sistema é realizada nesta área. O **administrador** (**type == 1**) pode realizar todo o processo de cadastro do procedimento (CRUD). A exclusão só será permitida caso não existam procedimentos solicitados. O **tipo 2** é **operador**, o qual tem permissão apenas para listar os exames e alterar apenas o preço.

## Referências

DEITEL, H. M.; DEITEL, P. J. *AJAX, Rich Internet Applications e Desenvolvimento*. [S.l.]: Editora Pearson, 2008.

FLANAGAN, D. *Javascript: o Guia Definitivo*. 4. ed. [S.l.]: Bookman, 2004.

FREEMAN, E.; FREEMAN, E. *Use A Cabeça! HTML com CSS e XHTML*. 2. ed. [S.l.]: Editora Alta Books, 2008.

NIXON, R. *Learning PHP, MySQL, and JavaScript - A Step-by-Step Guide to Creating Dynamic Websites*. [S.l.]: O'Reilly, 2009. I-XVII, 1-505 p. ISBN 978-0-596-15713-5.

W3C. *World Wide Web Consortium (W3C)*. 2017. [Http://www.w3.org/](http://www.w3.org/).