

Olá!

Você tem até as 18:00 **horas de hoje (05/10/2015)** para entregar o seu teste, conforme descrito abaixo.

Ferramentas:

- JDK 8
- Eclipse for Java EE
- Wildfly Java EE7 Full & Web

Frameworks:

- Spring 4.0 / MVC / Data
- Hibernate
- Interface web com Bootstrap

Banco de Dados:

- Postgres
- O script das tabelas está no final deste e-mail, após a assinatura.

Escopo:

- Sistema para controle de empréstimo de livros, que deverá possibilitar o cadastro de vários livros e pessoas que o tomarão emprestados.

Regras de negócio:

- Um livro não pode ser emprestado se o mesmo já estiver com outra pessoa;
- Todos os livros têm classificação por idade e não poderá ser emprestado se a pessoa não tiver a idade mínima cadastrada no livro.
- Uma pessoa não pode ter mais que dois livros emprestado ao mesmo tempo.
- O sistema deverá permitir cadastrar as seguintes informações:
 - Livro (Nome, Escritor, Ano edição, Classificação (A partir de que idade))
 - Pessoa (Nome, Data de Nascimento, CPF)
 - Empréstimo (Data de Empréstimo, Pessoa, Livro)

Regras de implementação:

- Persistir as informações no BD enviado, utilizando os frameworks solicitados
- Utilizar MVC
- Utilizar testes unitários para avaliar as regras;

Entrega:

- O teste deverá ser enviado em um e-mail contendo o link para o repositório deste projeto em sua conta pessoal no GitHub.
- Não deverá constar em nenhum lugar do projeto (estruturas de pastas, nomes de arquivos, código fonte, etc) a palavra "Desbravador".

Critérios de avaliação:

- Sistema funcionando e atendendo os requisitos descritos acima
- Utilização correta dos conceitos de Orientação a Objetos
- Padrão na construção das classes e nas nomenclaturas
- Criação de testes unitários.

Boa sorte!

Atenciosamente,

Script do banco de dados:

```
-- Este script criará o schema "testeds1" no usuário/banco corrente e
-- em seguida criará as tabelas necessárias.
-- A estrutura criada não deve ser alterada!
```

```
CREATE SCHEMA testeds1;
```

```
CREATE TABLE testeds1.livro
( idlivro bigserial NOT NULL,
  nome character varying(100),
  escritor character varying(100),
  anoedicao smallint,
  classificacao smallint,
  CONSTRAINT pk_livro PRIMARY KEY (idlivro));
```

```
CREATE TABLE testeds1.pessoa
( idpessoa bigserial NOT NULL,
  nome character varying(100) NOT NULL,
  datanascimento date,
```

```
    cpf character varying(14),
    CONSTRAINT pk_pessoa PRIMARY KEY (idpessoa));

CREATE TABLE testedsl.emprestimo
( idemprestimo bigserial NOT NULL,
  dataemprestimo date NOT NULL,
  idpessoa bigint NOT NULL,
  idlivro bigint NOT NULL,
  datahoradevolucao time without time zone,
  CONSTRAINT pk_emprestimo PRIMARY KEY (idemprestimo),
  CONSTRAINT fk_livro FOREIGN KEY (idlivro)
    REFERENCES testedsl.livro (idlivro) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT fk_pessoa FOREIGN KEY (idpessoa)
    REFERENCES testedsl.pessoa (idpessoa) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION);
```