

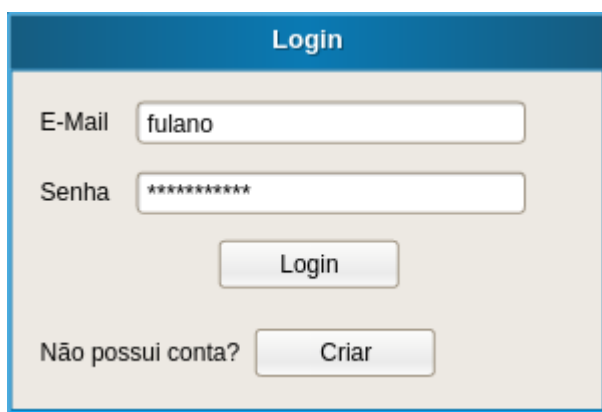


**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA
PARAÍBA
CAMPUS CAJAZEIRAS
DISCIPLINA DE PROGRAMAÇÃO ORIENTADA A OBJETOS
PROFESSOR: PAULO FREITAS**

TRABALHO PRÁTICO

O projeto consiste na criação de um sistema de agendas eletrônicas, semelhante ao Google Calendar, por exemplo. Nesse documento serão listadas algumas funcionalidades e mostrados alguns protótipos que devem auxiliar na construção da ferramenta. Lembrando que novas funcionalidades podem ser adicionadas de acordo com a necessidade das aplicações.

Ao abrir o programa o usuário deve ser direcionado à tela de Login. Somente usuários cadastrados e autenticados podem ter acesso ao sistema. Um exemplo dessa tela é mostrado a seguir:

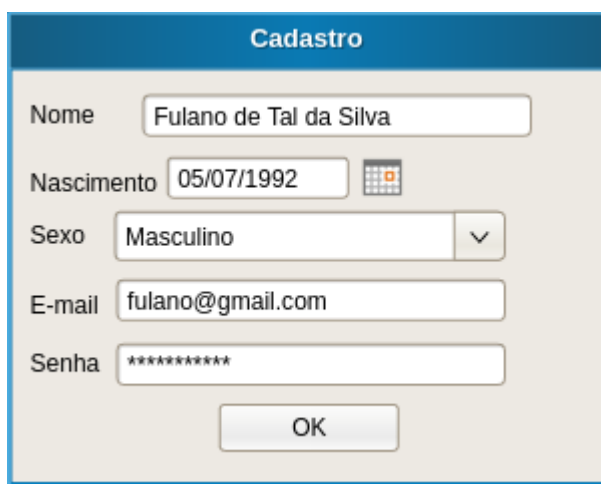


The image shows a login form with a blue header bar containing the word "Login". Below the header, there are two input fields: "E-Mail" with the text "fulano" and "Senha" with masked characters "*****". Below these fields is a "Login" button. At the bottom, there is a link "Não possui conta?" followed by a "Criar" button.

Se o usuário optar por criar uma nova conta, ele será redirecionado à página de cadastro. Nessa página serão solicitados alguns dados pessoais. Existem alguns detalhes importantes:

- Não deve ser possível informar datas inválidas. Exemplo: 31/02/2017.
- Não é possível cadastrar mais de um usuário para um mesmo e-mail.

Um exemplo da página de cadastro é mostrado pelo protótipo a seguir:



The image shows a registration form with a blue header bar containing the word "Cadastro". Below the header, there are several input fields: "Nome" with the text "Fulano de Tal da Silva", "Nascimento" with the date "05/07/1992" and a calendar icon, "Sexo" with a dropdown menu showing "Masculino", "E-mail" with the text "fulano@gmail.com", and "Senha" with masked characters "*****". Below these fields is an "OK" button.

Após fazer login o usuário chegará na página inicial. Nela serão mostrados todos os compromissos cadastrados para os próximos 30 dias por meio de uma tabela. Um mesmo usuário pode possuir diversas agendas, exemplo: Pessoal, Faculdade e Trabalho. Então, deve ser possível filtrar quais de quais agendas deseje-se mostrar os compromissos, e isso é feito selecionando no checkBox a agenda e clicando em atualizar.

Data	Hora	Compromisso
16/03/2017	14:00	Dentista
18/03/2017	07:00	Atividade POO
21/03/2017	16:00	Projeto

Ao clicar em novo compromisso, o usuário irá para a tela de cadastro de compromissos, onde serão solicitados alguns dados. Alguns campos são obrigatórios: Data, Hora, Descrição e Agenda. Para escolher a agenda deve ser utilizado um comboBox que lista todas as agendas cadastradas para aquele usuário. Um exemplo dessa tela é apresentado no protótipo a seguir:

Novo Compromisso

Data: 18/03/2017

Hora: 18:00:00

Descrição: Jogar bola

Local: XYZ Society

Agenda: Pessoal

OK

Na opção de gerenciar agendas, deverão ser listadas todas as agendas disponíveis e permitir a criação de novas agendas e edição das já existentes. Não deve ser possível haver mais de uma agenda com o mesmo nome.

Já em gerenciar compromissos, deve ser possível buscar compromissos de um intervalo de tempo. Devem ser mostrados os compromissos de todas as agendas. Ao clicar em um desses, deve ser possível editar ou excluir o mesmo.

A interface 'Gerenciar Compromissos' possui um cabeçalho azul com o título. Abaixo dele, há dois campos de texto para 'Início' (16/03/2017) e 'Fim' (31/03/2017), cada um com um ícone de calendário. Entre eles está um botão 'Atualizar'. Abaixo disso, há uma tabela com três colunas: 'Data', 'Hora' e 'Compromisso'.

Data	Hora	Compromisso
16/03/2017	14:00	Dentista
18/03/2017	07:00	Atividade POO
21/03/2017	16:00	Projeto

A entrega será dividida em duas etapas:

- Em **12/04/2017**, deverão ser apresentadas e entregues todas as classes de modelo da aplicação, bem como deve ser possível realizar o CRUD (Create, Read, Update e Delete) das seguintes entidades: Usuário, Agenda e Compromisso. O cadastro deve ser desenvolvido utilizando coleções e o Javadoc das classes de cadastro deverão ser feitos.
- Em **10/05/2017**, deverá ser entregue o projeto concluído, já com a interface gráfica e realizando a persistência de todas as entidades em arquivos e em algum banco de dados à sua escolha.

Observações:

- O trabalho poderá ser desenvolvido individualmente ou em duplas;
- Utilizar o GitHub para controle de versão. Caso o projeto seja desenvolvido em dupla, ambos devem interagir com o repositório.
- Em caso de plágio será atribuída nota 0 a todos os envolvidos;
- Seguir os padrões de nomenclatura do Java;
- Buscar o máximo de coesão e o mínimo de acoplamento entre as classes do projeto;
- Na versão final, sempre que trabalhar com datas, utilizar algum componente gráfico de calendário para receber as datas. Existem inúmeros, basta procurar;
- Na versão final, ao realizar o cadastro de usuário, a aplicação deve enviar um e-mail de confirmação para o endereço que foi informado pelo usuário.

Também existem muitas bibliotecas que fazem o envio de e-mails, basta procurar;

- Os protótipos apresentados nesse documento devem servir como um simples esboço. Não é obrigatório seguir essa interface à risca, fiquem a vontade para **criarm**, pois a interface também será critério de avaliação.