

# Frontend Web

## React

Criar um projeto React que contemple as seguintes funcionalidades:

1. Tela de Login
  - a. Campo de Login
  - b. Campo de Senha
  - c. Botão de visualizar Senha
  - d. Checkbox de Gravar Senha
    - i. Caso o usuário marque essa opção, armazenar no storage para que no próximo acesso já apareça preenchido, e caso ele desmarque limpar o storage.
  - e. Botão de Entrar
    - i. Ao clicar deve chamar o endpoint de Login.
    - ii. Redirecionar para Home caso o Login esteja correto
  - f. Botão de Cadastrar-se
2. Tela de Cadastrar-se
  - a. Campo de Login
  - b. Campo de Senha
  - c. Campo de Confirmar Senha
  - d. Botões para visualizar as senhas digitadas
  - e. Botão de Salvar
    - i. Deve validar se a Senha e Confirmar Senha são iguais
    - ii. Ao clicar deve chamar o endpoint de Cadastro
    - iii. Exibir uma mensagem de cadastro realizado com sucesso
3. Tela Home
  - a. Exibir lista de Skills que o usuário adicionou
    - i. A lista será composta pela imagem(url), nome da skill, level(version) e descrição
    - ii. O level deve ser possível editar na própria lista
    - iii. Também deve ter um botão de excluir a Skill da Lista
  - b. Botão de Adicionar Skill
    - i. Ao clicar no botão deve abrir uma modal de cadastro. Nessa modal terá:
      1. Uma combo para o usuário escolher a skill
        - a. O conteúdo dessa combo vai ser retornada de um endpoint
      2. Botão de Salvar
      3. Botão de Cancelar
  - c. Botão de Logout
4. Segurança
  - a. Não permitir o usuário acessar a página Home sem estar logado
5. Capriche no layout, pois o design também irá contar na avaliação
6. O projeto deve ser compartilhado no github num repositório público

# Frontend Mobile

## React Native

Criar um projeto React Native que contemple as seguintes funcionalidades:

1. Tela de Login
  - a. Campo de Login
  - b. Campo de Senha
  - c. Botão de visualizar Senha
  - d. Checkbox de Gravar Senha
    - i. Caso o usuário marque essa opção, armazenar no storage para que no próximo acesso já apareça preenchido, e caso ele desmarque limpar o storage.
  - e. Botão de Entrar
    - i. Ao clicar deve chamar o endpoint de Login.
    - ii. Redirecionar para Home caso o Login esteja correto
  - f. Botão de Cadastrar-se
2. Tela de Cadastrar-se
  - a. Campo de Login
  - b. Campo de Senha
  - c. Campo de Confirmar Senha
  - d. Botões para visualizar as senhas digitadas
  - e. Botão de Salvar
    - i. Deve validar se a Senha e Confirmar Senha são iguais
    - ii. Ao clicar deve chamar o endpoint de Cadastro
    - iii. Exibir uma mensagem de cadastro realizado com sucesso
3. Tela Home
  - a. Exibir lista de Skills que o usuário adicionou
    - i. A lista será composta pela imagem(url), nome da skill, level(version) e descrição
    - ii. O Level deve ser possível editar na própria lista
    - iii. Também deve ter um botão de excluir a Skill da Lista
  - b. Botão de Adicionar Skill
    - i. Ao clicar no botão deve abrir uma modal de cadastro. Nessa modal terá:
      1. Uma combo para o usuário escolher a skill
        - a. O conteúdo dessa combo vai ser retornada de um endpoint
      2. Botão de Salvar
      3. Botão de Cancelar
  - c. Botão de Logout
4. Capriche no layout, pois o design também irá contar na avaliação
5. O projeto deve ser compartilhado no github num repositório público.

# Backend

## Spring Boot

Criar um projeto Spring Boot que contemple as seguintes funcionalidades:

1. Serviço de Login
  - a. Esse serviço deve receber login e senha e verificar se está igual ao da base
  - b. A senha deverá ser criptografada
  - c. Deverá retornar um token para acesso aos demais serviços
2. Serviço de Cadastro
  - a. Esse serviço receberá o login e senha para ser cadastrado na base de dados
  - b. A senha deve ser armazenada criptografada
3. Serviço de Listagem de Skills
  - a. Esse serviço deve receber o id do usuário e retornar todas as skills associadas a ele e seu respectivo level para preenchimento da interface.
4. Serviço de Associar Skill
  - a. Esse serviço deve receber o usuário, a skill e o level para persistir na base de dados
5. Serviço de Atualizar Associação de Skill
  - a. Esse serviço deve receber o id da associação da skill e o level para atualização na base de dados
6. Serviço de Excluir Associação de Skill
  - a. Esse serviço deve receber o id da associação da skill e excluir da base de dados
7. Apenas o Serviço de Login deve ser público. Os demais devem ter segurança JWT e ser acessados apenas com um token válido
8. O projeto deve conter o Spring Fox para gerar documentação automática dos serviços pelo swagger.
9. Lembre-se de seguir as boas práticas de API RESTFul
10. O projeto deve ser compartilhado no github num repositório público

# Banco de dados

## PostgreSql

1. Façam a modelagem do banco que comporte os requisitos solicitados, utilizando constraints corretamente, e salvem os scripts em um arquivo chamado SistemaSkill.sql;
2. O script deverá ser salvo juntamente no repositório para fins de avaliação sobre a modelagem do banco;
3. Criem as devidas *sequences* para utilização de ids auto incremento em todas as tabelas, incluindo qualquer associativa que por ventura seja necessária;
4. Os ids devem usar a sequence criada;