**Relatório de desenvolvimento de API e telas de cadastro para o painel de senhas do Registro Acadêmico**

**Disciplina:** Programação para Internet II

**Professor:** Fábio Henrique M. Oliveira

**Alunos:** Daniel Barros Candeira

Levi Alves

**Problema**

O registro acadêmico (RA) é o responsável por lidar com a papelada relativa aos alunos (matrícula, rematrícula, trancamento de curso etc.) do Instituto Federal de Brasília (IFB). Contudo, não há nenhuma forma de organização ao se tratar das filas de atendimento, ao chegar no local (RA) deve-se identificar quem é o último da fila, e lembrar que será atendido após essa pessoa.

**Organização do Projeto**

A fim de sanar o problema, os alunos do 5º (quinto) semestre do curso Sistemas para Internet (TSI), na disciplina de programação para Internet II, receberam a tarefa de desenvolver um sistema de organização de filas com o uso de senhas como um projeto da disciplina.

A turma foi divida em duas equipes, a primeira ficou responsável por desenvolver a parte do sistema a ser utilizada pelas pessoas que serão atendidas (*front-end*), a outra ficou responsável por desenvolver o que será usado pelos funcionários do RA (*backend*).

As atividades a serem realizadas foram divididas em três Sprints (pontos de entrega), na primeira foram entregues os desenhos e a implementação das telas do “usuário” do RA pela equipe do *front-end*.

A equipe do *backend* ficou responsável pela implementação das telas de cadastro de status, categoria e tipo da senha, e as funções getSequencia (retorna a parte numérica da ultima senha cadastrada), getTipo (retorna os tipos de senha cadastrados no banco), getCategoria (retorna as categorias de atendimento registradas no banco), postSenha (cadastra uma nova senha no banco), na API .

Na segunda Sprint a equipe do *front-end* ficou responsável pela implementação do painel de atendimento, onde as senhas chamadas pelo atendente seriam mostradas, e integrar a aplicação a API no Elelctron.

E o *backend* ficou responsável pela implementação das telas de cadastro de campus, guichê e atendente, bem como a tela de atendimento, e uma notificação push que enviaria última senha chamada pelo atendente para o painel de atendimento. Na terceira e última Sprint as equipes devem escrever um relatório das atividades realizadas e realizar uma apresentação do que foi concluído.

**Tecnologias utilizadas**

A equipe responsável apenas pela parte do sistema da pessoa a ser atendida utilizou o framework javaScript *Electron*, para o desenvolvimento da parte do atendente foram utilizados Django e Django rest-framework frameworks python para desenvolvimento web.

Python Django é um framework que vem ganhando força no mercado atualmente devido ao formato MVC (Model-View-Controller), e as facilidade provenientes do python.

Django rest-framework, como o nome já diz é um framework para utilização da tecnologia **rest**, que funciona como uma camada intermediária entre a aplicação e o banco de dados, usando requisições HTTP para a troca de informações.

As requisições HTTP mais utilizadas são GET, POST, PUT, DELETE, e são usadas respectivamente para resgatar, publicar, atualizar e deletar informações, que no caso do **rest** são realizadas no banco de dados.

**Etapas de desenvolvimento do backend**

A fim de implementar as telas de cadastro de tipo, categoria e status da senha, e cadastro de guichês, atendentes e campus, foram criados os modelos no django para o banco de dados, o modelo do banco de dados utilizado foi disponibilizado pelo professor da disciplina.

As telas de cadastro foram criadas com utilização do HTML5 e CSS, após a implementação das telas de cadastro foram implementadas as funções da API, que seriam utilizadas para se “comunicar” com o que foi desenvolvido pela equipe do *front-end*.

A API permite o acesso aos tipos, categorias, status e senhas cadastrados no banco, e ao cadastro de uma nova senha, para acessar os itens disponibilizados no banco é necessário realizar um cadastro com nome, e-mail e senha de usuário, após o cadastro o usuário da API tem acesso a um *token* (chave de autenticação) que o permite acessar as funções disponíveis pela API.