



UNIVERSIDADE FEDERAL DE RORAIMA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO
DCC802 - PROJETO E IMPLEMENTAÇÃO DE SISTEMAS
(2024.1) PROF. MARCELO HENRIQUE OLIVEIRA HENKLAIN

Documentação Projeto para Acompanhamento de Inserção de Alunos Egressos Empreendedores

Aluno: Lucas Prado Ribeiro - Matrícula: 2020018187

Introdução

O objetivo deste projeto é criar uma ferramenta para consultar os dados dos alunos egressos da UFRR e verificar se estão associados a algum CNPJ de empresa, gerando estatísticas a partir desses dados.

Modelo Incremental

- Primeiro incremento: Download de CNPJs
 - Desenvolver um script para baixar os CNPJs da Receita Federal.
 - Armazenar os dados em um banco de dados PostgreSQL ou no Google Drive.
- Segundo incremento: Projeto Web em Django
 - Criar um sistema web para cadastrar os alunos egressos, incluindo CPF, nome e data de formatura.
 - Implementar a verificação automática do CNPJ associado ao CPF do aluno.
- Terceiro incremento: Dashboard de Estatísticas
 - Desenvolver um dashboard para exibir gráficos e estatísticas relacionados aos alunos empreendedores e seus vínculos com empresas.

Interface Gráfica (GUI)

- Tela de Cadastro de Alunos Egressos
 - Campos para inserção de CPF, nome do aluno e data de formatura.
 - Botão "Cadastrar" para submeter os dados.
- Gráficos (Dashboard) com Estatísticas

- Exibir gráficos interativos com informações estatísticas, como número de alunos empreendedores, tipos de empresas vinculadas, etc.

Funcionalidades Principais

- Cadastro de Alunos Egressos
 - Captura de informações como CPF, nome e data de formatura.
 - Validação dos dados antes do cadastro.
- Consulta Automática de CNPJs
 - Integração com a Receita Federal para consulta dos CNPJs vinculados aos CPFs dos alunos.
- Download Automático de Tabela de CNPJs
 - Realizar o download automático da tabela de CNPJs da Receita Federal para armazenamento no PostgreSQL ou Google Drive.
- Análise e Geração de Estatísticas
 - Utilizar a biblioteca pandas para análise dos dados coletados.
 - Gerar estatísticas relevantes, como número de alunos empreendedores e tipos de empresas associadas.

Fluxo de Trabalho

- O usuário acessa o sistema e realiza o cadastro do aluno egresso.
- As informações são validadas e consultadas automaticamente na Receita Federal.
- Os dados são analisados para geração das estatísticas, que são exibidas no dashboard.

Requisitos Técnicos

- Linguagem de Programação: Python.
- Bibliotecas: Django Framework para a interface web, requests para consultas web, pandas para análise de dados.
- Banco de Dados: PostgreSQL ou integração com o Google Drive para armazenamento dos CNPJs e outras informações necessárias.

Teste de software unitário automatizado

Verifica o comportamento funcional de componentes individuais, geralmente ao nível de classe e função.

Django fornece um *framework* de teste com uma baixa hierarquia de classes construída na biblioteca padrão [unittest](#) de Python. Apesar do nome, este

framework de teste é adequado para testes unitários e de integração. O *framework* Django adiciona métodos e ferramentas de API para ajudar a testar o comportamento web e específico do Django. Isso permite você simular requisições, inserir dados de teste e inspecionar as saídas do seu aplicativo

Construção do teste

- Derivar de qualquer uma das classes base de teste de Django (ou *unittest*) e então escrever métodos separados para verificar se a funcionalidade específica funciona como esperado (testes usam métodos "assert" para testar se a expressão resulta em valores True ou False, ou se os dois valores são iguais, etc.). Quando iniciar a execução de um teste, o framework executa os métodos de teste escolhidos em suas classes derivadas. Os métodos de teste são executados independentemente, com configuração comum e/ou comportamento *tear-down* definido na classe, como mostrado abaixo.

Execução dos testes

- A maneira mais fácil para executar todos os testes é usar o comando:

```
python3 manage.py test
```

Isso descobrirá todos arquivos nomeados com o padrão test*.py no diretório atual e executará todos testes definidos usando as classes base apropriadas. Por padrão, os testes irão reportar individualmente apenas falhas no teste, seguidos por um resumo do teste.

Boas Práticas

Controle de versão

- Como boa prática que pode ser adotada para melhorar o desenvolvimento e a manutenção do sistema, temos o controle de versão (Git e Github) para gerenciar o código-fonte do projeto. Isso permite rastrear alterações, colaborar com outros desenvolvedores, gerenciar branches e reverter alterações conforme necessário.

Estratégias de levantamento de requisitos

- As partes interessadas do projeto estão listada em:

Representantes da UFRR (egressos), e os usuários finais do sistema (administradores, professores, alunos), especificamente, o professor Marcelo Henklain e o professor Acauan Ribeiro.

Levantamento de acordo com pontos de vista

- Nesse método, a primeira etapa consiste em identificar as diferentes perspectivas. Isso é feito ouvindo os próprios consumidores e analisando as informações fornecidas por eles.

Para isso serão realizadas reuniões e levantamentos de questões.

A informação obtida com os usuários deverá ser tratada para criar modelos de pontos de vista. Esses modelos são agrupados entre si, seguindo uma hierarquia que identifica, especialmente, o que todos os clientes de diversas opiniões têm em comum. Isso quer dizer que o que for comum a todos tem uma hierarquia mais alta, tendo mais prioridade no desenvolvimento.

Prototipagem

- Protótipo tem por objetivo explorar aspectos críticos dos requisitos de um produto, implementando de forma rápida um pequeno subconjunto de funcionalidades deste produto. O protótipo é indicado para validar as informações coletadas do usuário na primeira etapa do levantamento, obter o feedback e realizar ajustes, se necessário. A partir disso, serão incrementadas as outras funcionalidades de acordo com o que o cliente desejar.

Perguntas

Prof Marcelo

- **“Quais as principais operações que o sistema deve ter ?”**

R = Estatísticas dos egressos que empreenderam, conforme TCC do Rodrigo.

- **“Quem utilizará o sistema ?”**

R = Inicialmente, apenas professores do curso de computação, com destaque para Prof. Marcelo, Prof. Luciano e Prof. Felipe Lobo. Os dois últimos por conta das funções de coordenação e chefe, respectivamente.

- **“Quem entregará as entradas do recurso (como dados dos egressos) ?”**

R = Nome, CPF, data de conclusão do curso e informações análogas serão fornecidas pelas DTI em csv.

- **“Quais serão os dados principais a serem mostrados ?”**

R = Ver TCC do Rodrigo e artigos de 2020 do Prof. Rodrigues da Unirio. De memória:

- % de quem empreendeu
- Nome da empresa
- CNAE da empresa
- % de empresas na área de computação
- Data de criação da empresa
- Se a empresa é só do aluno ou se é sociedade
- Se possível: faturamento da empresa.

- **“Quais os principais desafios que a atual solução representa ?”**

R = Desafio 01: automatizar coleta e processamento da informação, considerando que a cada 6 meses precisam ser atualizadas. A atualização pode ser manual, isto é, acionada por quem for rodar o script.

Prof Acauan

- **“Como seria um design de apresentação dos dados de maneira aceitável ?”**

R = Esta pergunta está pouco vaga. A apresentação dos dados deve ser clara e compreensível para quem for visualizar esses dados. Uma boa visualização conta uma história, eliminando elementos irrelevantes dos dados e ressaltando informações úteis. Pode-se resumir uma boa visualização de dados nesses pontos:

- * Tornar os dados atraentes e fáceis de entender
- * Identificar tendências e situações atípicas dentro de um conjunto de dados
- * Contar uma história encontrada nos dados
- * Reforçar um argumento ou opinião
- * Destacar um ponto importante em um conjunto de dados

- **“Quais são os requisitos de segurança e privacidade que devem ser considerados ao acessar e manipular os dados dos alunos egressos?”**

R = A LGPD não se aplica para fins acadêmicos (Art. 4º, inciso II, alínea b), com exceção dos artigos 7 e 11. Observe o que dizem os artigos:

“Art. 7º O tratamento de dados pessoais somente poderá ser realizado nas seguintes hipóteses:

IV – para a realização de estudos por órgão de pesquisa, garantida, sempre que possível, a anonimização dos dados pessoais;”

“Art. 11. O tratamento de dados pessoais sensíveis somente poderá ocorrer nas seguintes hipóteses:

I – quando o titular ou seu responsável legal consentir, de forma específica e destacada, para finalidades específicas;II – sem fornecimento de consentimento do titular, nas hipóteses em que for indispensável para:

c) realização de estudos por órgão de pesquisa, garantida, sempre que possível, a anonimização dos dados pessoais sensíveis; Ou seja, não há problema em coletar esses dados para fins de pesquisa, desde que eles sejam anonimizados. E lembre-se de não compartilhá-los com terceiros. Se você lidera uma equipe de pesquisa, oriente os membros dessa equipe para terem o mesmo cuidado.

- **“Quais são os recursos de visualização e relatórios que o cliente deseja disponibilizar para acessar as estatísticas geradas pelo sistema?”**

R = Isso vai depender do que você quer apresentar. Uma tabela é uma visualização, uma das mais utilizadas para apresentar dados. Um gráfico, uma imagem, um mapa. Tem vários tipos de visualizações que você deve analisar o tipo de dado que você quer apresentar e escolher a melhor que conta uma história ou tire uma dúvida de maneira clara.

- **“O sistema deve ser acessível em dispositivos móveis (smartphones e tablets) ou será utilizado principalmente em computadores desktop?”**

R = Indicado que ele seja acessível para smartphones. Já comece a implementação do sistema pensando na “responsividade” do mesmo, ou seja ele funcionar (não quebrar) nos tamanhos de tela mais utilizados. Smartphones e desktops.

Diagrama de relacionamento de entidade do banco de dados (inicial)

Apresenta os dados das tabelas criadas a partir do script que faz o download dos dados e cria um banco de dados postgresQL com eles. Com isso, também foi criada uma tabela “busca_cnpj” para armazenar as buscas realizadas e retornar os resultados encontrados na tabela.

cnpj
cnpj_busca
id
nome
cpf
cnpj_base
nome_fantasia
cd_motivo_situacao_cadastral
cd_cnae_principal
dt_inicio_atividade
st_uf

cnpj
tb_dados_simples
st_cnpj_base
st_opcao_simples
dt_opcao_simples
dt_exclusao_simples
st_opcao_mei
dt_opcao_mei
dt_exclusao_mei

cnpj
tb_qualificacao_socio
cd_qualificacao
st_qualificacao

cnpj
tb_motivo_situacao_cadastral
cd_motivo_situacao_cadastral
st_motivo_situacao_cadastral

cnpj
tb_cnae
cd_cnae
st_cnae

cnpj
tb_estabelecimento
st_cnpj_base
st_cnpj_ordem
st_cnpj_div
cd_matriz_filial
st_nome_fantasia
cd_situacao_cadastral
dt_situacao_cadastral
cd_motivo_situacao_cadastral
st_cidade_exterior
cd_pais
dt_inicio_atividade
cd_cnae_principal
cd_cnae_secundario
st_tipo_logradouro
st_logradouro
st_numero
st_complemento
st_bairro
st_cep
st_uf
cd_municipio
st_ddd1
st_telefone1
st_ddd2
st_telefone2
st_ddd_fax
st_fax
st_email
st_situacao_especial
dt_situacao_especial

cnpj
tb_municipios
cd_municipio
st_municipio

cnpj
tb_empresa
st_cnpj_base
st_razao_social
cd_natureza_juridica
cd_qualificacao
vl_capital_social
cd_porte_empresa
st_ente_federativo

cnpj
tb_socio
st_cnpj_base
cd_tipo
st_nome
st_cpf_cnpj
cd_qualificacao
dt_entrada
cd_pais
st_representante
st_nome_representante
cd_qualificacao_representante
cd_faixa_etaria

Resultados do Sistema Finalizado - Documentação

O projeto consiste em:

- Um script Python para baixar dados de CNPJs.
- Uma aplicação Django para acessar e gerenciar esses dados.
- Um banco de dados PostgreSQL para armazenar as informações.

Instalação e Requisitos

- Python 3.x
- PostgreSQL
- pip
- Git

Explicação do projeto

`script_cnpj_postgres.py`

Carregar variáveis de ambiente:

```
ENV_PATH = os.path.join(os.path.dirname(__file__), '..', '.env')
config = Config(RepositoryEnv(ENV_PATH))
```

O código carrega variáveis do arquivo `.env` para definir credenciais do banco de dados.

`get_latest_url():`

Busca a última URL no 'http://200.152.38.155/CNPJ/dados_abertos_cnpj/', Faz scraping da página de dados abertos e encontra o link mais recente de arquivos disponíveis para download.

Faz uma requisição HTTP à URL base e lê o conteúdo HTML da página para extrair uma tabela de links para arquivos `.zip`.

O DataFrame resultante é filtrado para incluir apenas arquivos `.zip`.

`create_table()`: Cria uma tabela no banco de dados Verifica se a tabela já existe e, se não, cria uma nova tabela baseada nos layouts definidos.

`settings.py`

Carrega arquivos front-end

STATIC_URL: Este define a URL pública para acessar os arquivos estáticos. No nosso caso, está definido como '/static/', o que significa que todos os arquivos estáticos serão acessíveis via /static/ no navegador (exemplo: <http://seusite.com/static/arquivo.css>).

models.py

Define a estrutura dos dados que você utiliza no seu sistema. Ele descreve como as tabelas do banco de dados serão representadas e mapeadas em classes Python. A class Busca é onde serão armazenados os socios buscados na aplicação e salvos no banco.

views.py

Contém a lógica que será executada quando certas URLs forem acessadas. Ele processa as requisições e gera as respostas que serão enviadas ao usuário

Instalação

Passo a passo

Clone o repositório no link https://github.com/Lucasx10/PIS_dados_cnpj:

```
git clone https://github.com/Lucasx10/PIS_dados_cnpj.git
cd PIS_dados_cnpj
```

Crie e ative um ambiente virtual:

```
python -m venv venv
source venv/bin/activate # Para Windows use: venv\Scripts\activate
```

Instale as dependências:

```
pip install -r requirements.txt
```

Configure o PostgreSQL:

- Crie um banco de dados PostgreSQL.

- Atualize as configurações do banco de dados no arquivo .env

Exemplo de configuração no arquivo .env.example:

Configuração

Script para Baixar Dados

Execute o script para baixar e armazenar CNPJs:

```
cd script/  
python script_cnpj_postgres.py
```

Script em python irá carregar os arquivos de cnpj dos dados públicos da Receita Federal no banco de dados PostgreSQL configurado. (Ele demora um pouco)

Execute as migrações do Django:

```
cd ..  
cd projeto_cnpj/  
python manage.py migrate
```

Uso do sistema

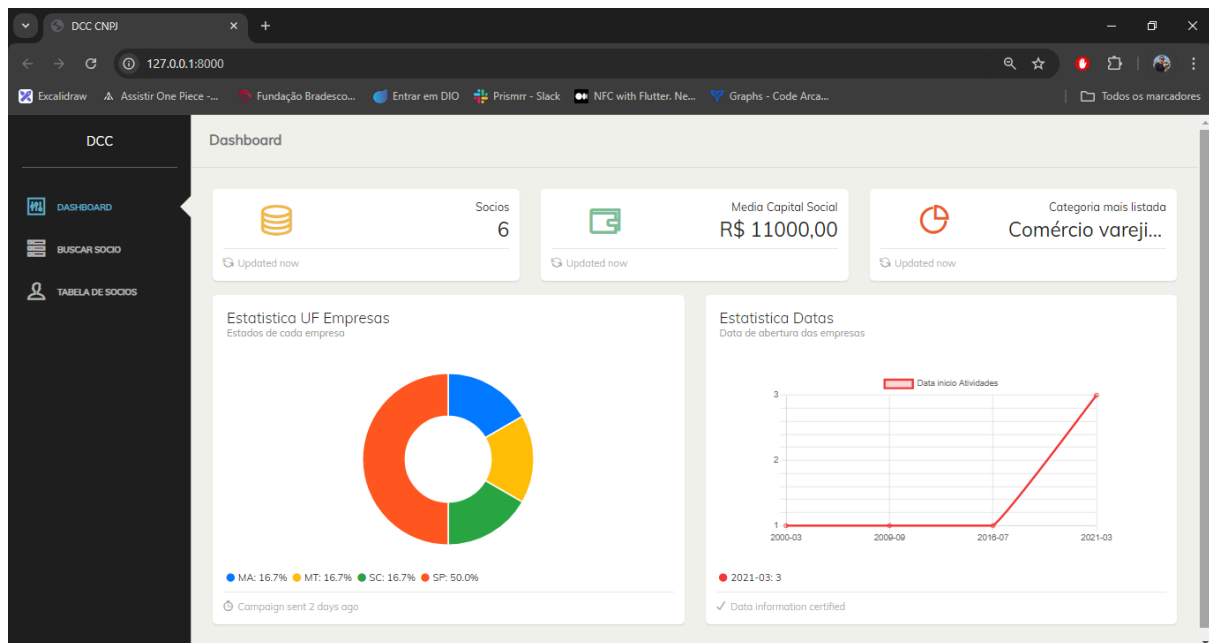
Inicie o servidor Django:

```
python manage.py runserver
```

Acesse a aplicação em: <http://127.0.0.1:8000>

Telas do sistema Web

Dashboard



São apresentados os principais dados que estão presentes na tabela “cnpj_busca” como quantidade de Sócios, Média de capital social e o cnae mais listado na tabela. Em baixo são apresentados gráficos das UF’s das empresas dos sócios e ao lado o gráfico com as datas iniciais das empresas.

Busca

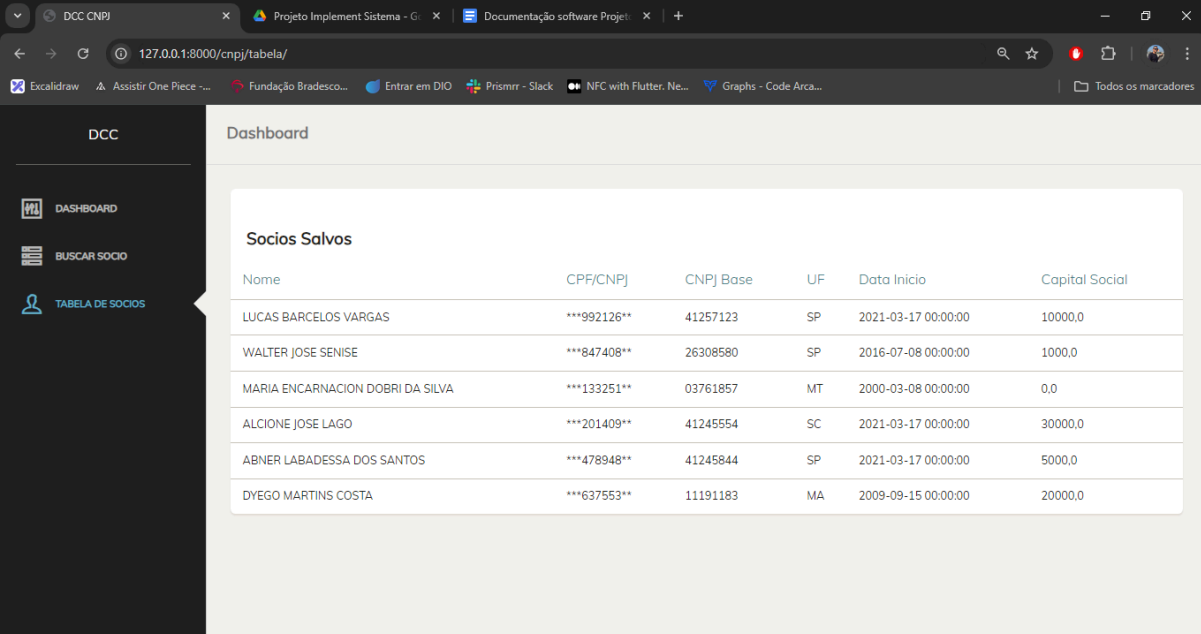
The screenshot shows the 'DCC CNPJ' search interface. The left sidebar contains links for 'DASHBOARD', 'BUSCAR SOCIO', and 'TABELA DE SOCIOS'. The main area features a search form with fields for 'Nome' (containing 'MOACIR JOSE MORAIS') and 'CPF' (containing '***335319**'). A 'Buscar' button is located below the form. Below the search form is a table with the following columns: 'CNPJ base', 'Estabelecimento', 'Motivo', 'CNAE Principal', 'Data de Início', 'UF', and 'Valor Capital'. The table contains one row of data: '22928372', 'ASSOCIACAO DOS AGRICULTORES JARACATIA DA COMUNIDADE DE SANTA INES - NOVA PRATA DO IGUAÇU - PR', 'SEM MOTIVO', 'Atividades de associações de defesa de direitos sociais', '2015-07-21 00:00:00', 'PR', and '0.0'. A 'Salvar no banco de dados' button is located below the table.

CNPJ base	Estabelecimento	Motivo	CNAE Principal	Data de Início	UF	Valor Capital
22928372	ASSOCIACAO DOS AGRICULTORES JARACATIA DA COMUNIDADE DE SANTA INES - NOVA PRATA DO IGUAÇU - PR	SEM MOTIVO	Atividades de associações de defesa de direitos sociais	2015-07-21 00:00:00	PR	0.0

A busca pelos sócios no banco consiste em pesquisar pelo nome do sócio com letras maiúsculas e o seu cpf sendo ele padronizado com *** nos 3 primeiros dígitos e ** nos 2 últimos dígitos por conta da LGPD, se houver algum cnpj onde esses dados têm relação irá retornar os dados da empresa e a opção de salvar no banco

de dados, ao clicar em salvar será salvo no banco e retornará uma mensagem dizendo que foi salva, se essa empresa já estiver listada na nossa tabela ele irá informar que já existe aquela busca no banco e desativará o botão. Se não encontrar nenhum dado ele retorna a mensagem que não foi encontrado nenhum dado.

Tabela



The screenshot shows a web browser window with the URL `127.0.0.1:8000/cnpj/tabela/`. The application has a dark sidebar on the left with the following menu items: DCC, DASHBOARD, BUSCAR SOCIO, and TABELA DE SOCIOS (highlighted). The main content area is titled 'Dashboard' and contains a table labeled 'Socios Salvos'. The table has six columns: Nome, CPF/CNPJ, CNPJ Base, UF, Data Inicio, and Capital Social. It lists six partners with their respective details.

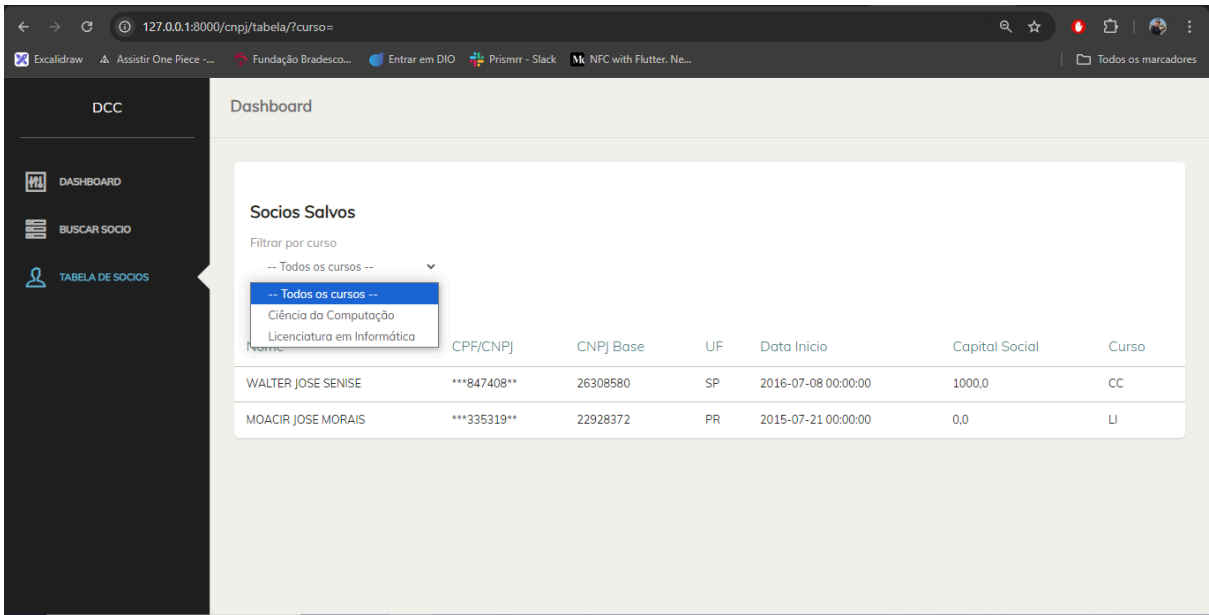
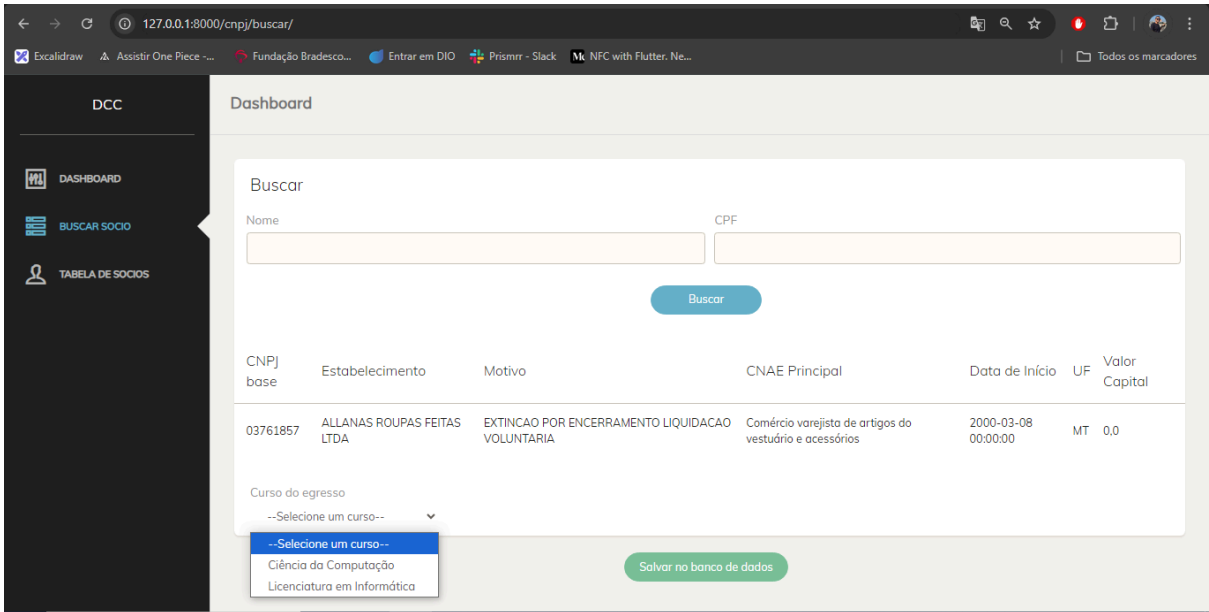
Nome	CPF/CNPJ	CNPJ Base	UF	Data Inicio	Capital Social
LUCAS BARCELOS VARGAS	***992126**	41257123	SP	2021-03-17 00:00:00	10000,0
WALTER JOSE SENISE	***847408**	26308580	SP	2016-07-08 00:00:00	1000,0
MARIA ENCARNACION DOBRI DA SILVA	***133251**	03761857	MT	2000-03-08 00:00:00	0,0
ALCIONE JOSE LAGO	***201409**	41245554	SC	2021-03-17 00:00:00	30000,0
ABNER LABADESSA DOS SANTOS	***478948**	41245844	SP	2021-03-17 00:00:00	5000,0
DYEGO MARTINS COSTA	***637553**	11191183	MA	2009-09-15 00:00:00	20000,0

Traz informações dos sócios salvos na pesquisa como Nome, CPF, CNPJ Data Inicio e Capital social. Podendo ser editados e alterados pelo admin do django, acessando a url: <http://127.0.0.1:8000/admin>

Melhoria/Aperfeiçoamento

Foi solicitado pelo cliente que houvesse um campo que informasse qual curso o aluno egressos pesquisado pertence. Foi implementado então esse novo campo na aplicação e também no banco de dados de modo que ele pudesse ser filtrado na busca.

Resultado



Referencia:

<https://pt.linkedin.com/pulse/base-cnpj-receita-federal-python-ne-y-moresco>