

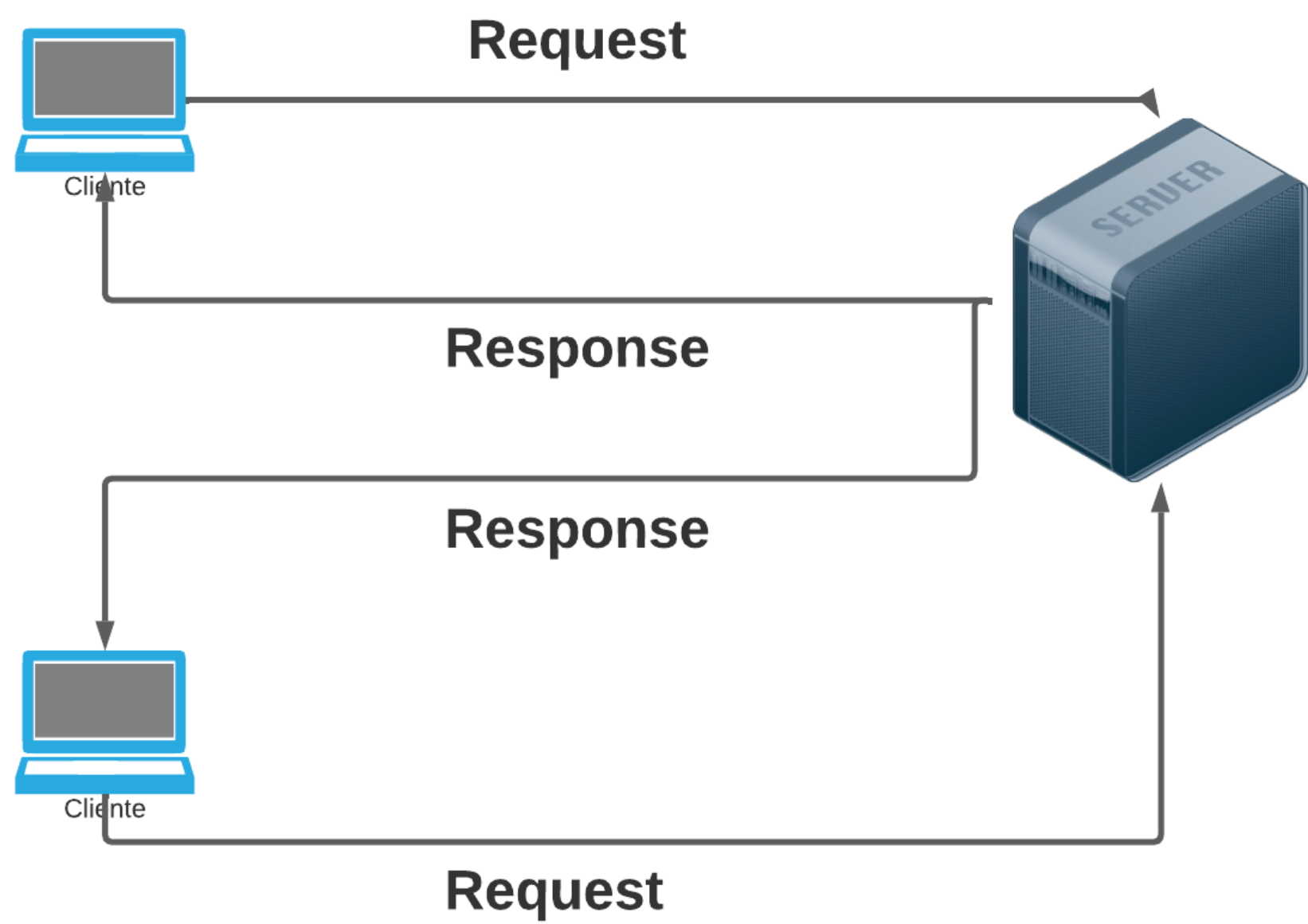


Aula 1

Acesse diretamente pelo Notion:

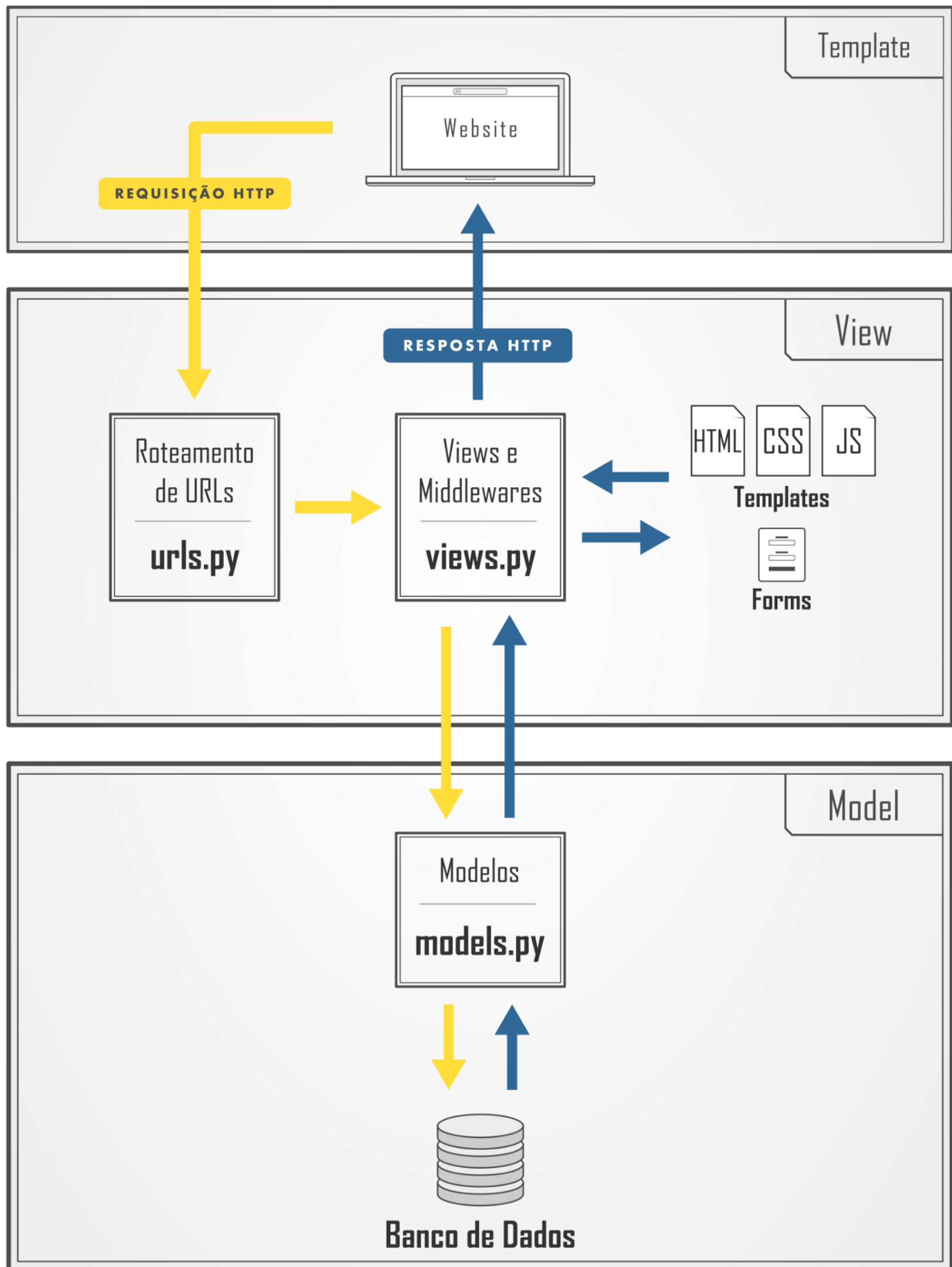
```
https://grizzly-amaranthus-f6a.notion.site/Aula-1-05f0cce37def4684a07b3e7593ac6fa8?pvs=4
```

▼ Conceitos



Fluxo de dados no Django:

ARQUITETURA DO django



▼ O projeto

```
https://www.figma.com/design/PeBjtQFxrjhN4hJxYcVOYR/Pythonando-PSW-11?node-id=0-1&t=qMs0d8R6MvmDfE1J-1
```

▼ Configurações iniciais

Primeiro devemos criar o ambiente virtual:

```
# Criar
# Linux
python3 -m venv venv
# Windows
python -m venv venv
```

Após a criação do venv vamos ativa-lo:

```
#Ativar
# Linux
source venv/bin/activate
# Windows
venv\Scripts\Activate

# Caso algum comando retorne um erro de permissão execute o código e tente novamente:

Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy RemoteSigned
```

Agora vamos fazer a instalação do Django e as demais bibliotecas:

```
pip install django
pip install pillow
```

Vamos criar o nosso projeto Django:

```
django-admin startproject core .
```

Rode o servidor para testar:

```
python manage.py runserver
```

Crie o app usuario:

```
python manage.py startapp usuarios
```

Ative o auto-save

INSTALE O APP!

▼ Cadastro

Crie uma URL para o app usuario:

```
path('usuarios/', include('usuarios.urls')),
```

No app usuario crie uma URL para cadastro:

```
from django.urls import path
from . import views

urlpatterns = [
    path('cadastro/', views.cadastro, name="cadastro"),
]
```

Em usuarios/views.py crie a view cadastro:

```
def cadastro(request):
    if request.method == "GET":
        return render(request, 'cadastro.html')
```

Configure onde o Django deve buscar por arquivos de templates:

```
os.path.join(BASE_DIR, 'templates')
```

Em templates crie o base.html

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>{% block 'title' %}{% endblock %}</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel=
    {% block 'head' %}{% endblock 'head' %}
  </head>
  <body>
    {% block 'body' %}{% endblock 'body' %}
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"
  </body>
</html>
```

Crie o cadastro.html

```
{% extends "base.html" %}

{% block 'head' %}

{% endblock 'head' %}
```

```

{% block 'body' %}

<div class="container-fluid">
  <div class="row">
    <div class="col-md-2 d-flex justify-content-center bg-img ">
    </div>
    <div class="col-md-10 d-flex justify-content-center bg-color-dark form-auth">
      <div class="container-form">
        <h1 class="text-logo">START.SE</h1>
        <h1>Cadastro-se</h1>
        <br>
        <form action="#" >
          <label>Username</label>
          <input type="text" class="form-control input-pers" name="username" placeh
          <br>
          <label>Senha</label>
          <input type="password" class="form-control input-pers" name="senha" place
          <br>
          <label>Confirmar senha</label>
          <input type="password" class="form-control input-pers" name="confirmar_se
          <div class="centralizar">
            <input type="submit" class="btn btn-indigo" value="Cadastrar">
            <a href="#" class="texto">Já possui uma conta</a>
          </div>
        </form>
      </div>
    </div>
  </div>
</div>

{% endblock 'body' %}

```

Configure os arquivos estáticos:

```

STATIC_URL = '/static/'
STATICFILES_DIRS = (os.path.join(BASE_DIR, 'templates/static'),)
STATIC_ROOT = os.path.join('static')

MEDIA_ROOT = os.path.join(BASE_DIR, 'media')
MEDIA_URL = '/media/'

```

Crie o geral/css/base.css

```

:root{

  --main-color: #161424;
  --dark-color: #0D0C16;
  --contrast-color: #4F46E5;

}

body{

```

```

    color: white !important;
    background-color: var(--main-color) !important;
}

.bg-color-dark{

    background-color: var(--main-color);

}

.p-bold{

    font-weight: bold;

}

.color-dark{
    color: var(--dark-color);
}

.dark-color{
    background-color: var(--dark-color);
    color: white !important;
}

.centralizar{
    text-align: center;
}

::-webkit-input-placeholder {
    color: rgb(196, 196, 196) !important;
}

:-moz-placeholder { /* Firefox 18- */
    color: rgb(196, 196, 196) !important;
}

::-moz-placeholder { /* Firefox 19+ */
    color: rgb(196, 196, 196) !important;
}

:-ms-input-placeholder {
    color: rgb(196, 196, 196) !important;
}

```

Importe o base.css em base.html:

```
<link rel="stylesheet" href="{% static 'geral/css/base.css' %}">
```

Agora em usuarios/css crie o usuarios.css

```

.bg-img{
    background-image: url('');
    height: 100vh;
    background-size: cover;
}

```

```

}

.form-auth{
  padding: 50px;
}

.text-logo{
  color: rgb(161, 161, 161);
  font-weight: 100;
}

.container-form{
  width: 35%;
  margin-top: 4em;
}

.input-pers{
  margin-top: 5px;
  background-color: transparent !important;
  font-size: 18px;
  color: white !important;
}

.btn-indigo{
  background-color: var(--contrast-color);
  color: white;
  width: 100%;
  margin-top: 20px;
  font-size: 20px;
}

.btn-indigo-normal{

  background-color: var(--contrast-color);
  color: white;

}

.btn-indigo:hover{
  background-color: #332cb6;
}

.centralizar{
  text-align: center;
}

.texto{
  color: white;
  text-decoration: none;
}

```

Importe o CSS criado acima:

```
<link rel="stylesheet" href="{% static 'usuarios/css/usuarios.css' %}">
```

Adicione as imagens em usuarios/img

https://drive.google.com/drive/folders/1iYSK7t-769dRV2gAZ_TUyBmM60GMXbW-?usp=drive_link

```
https://drive.google.com/drive/folders/1iYSK7t-769dRV2gAZ_TUyBmM60GMXbW-?usp=drive_link
```

Adicione o caminho para imagem no CSS:

```
background-image: url('/static/usuarios/img/bg.png');
```

Execute as migrações:

```
python manage.py makemigrations
python manage.py migrate
```

Altere o FORM para enviar os dados ao back-end:

```
<form action="{% url 'cadastro' %}" method="POST">{% csrf_token %}
```

Na view cadastro:

```
def cadastro(request):
    if request.method == "GET":
        return render(request, 'cadastro.html')
    elif request.method == "POST":
        username = request.POST.get('username')
        senha = request.POST.get('senha')
        confirmar_senha = request.POST.get('confirmar_senha')

        if not senha == confirmar_senha:
            print(1)
            return redirect('/usuarios/cadastro')

        if len(senha) < 6:
            print(2)
            return redirect('/usuarios/cadastro')

        users = User.objects.filter(username=username)

        if users.exists():
            print(3)
            return redirect('/usuarios/cadastro')

        user = User.objects.create_user(
            username=username,
            password=senha
        )
```



```
return redirect('/usuarios/logar')
```

Configure as mensagens:

```
from django.contrib.messages import constants

MESSAGE_TAGS = {
    constants.DEBUG: 'alert-primary',
    constants.ERROR: 'alert-danger',
    constants.SUCCESS: 'alert-success',
    constants.INFO: 'alert-info',
    constants.WARNING: 'alert-warning',
}
```

Adicione as mensagens em trechos específicos de código:

```
messages.add_message(request, constants.ERROR, 'A senha deve possuir pelo menos 6 caracteres')
```

Renderize as mensagens no HTML:

```
{% if messages %}
    <br>
    {% for message in messages %}
        <section class="alert {{message.tags}}">
            {{message}}
        </section>
    {% endfor %}
{% endif %}
```

▼ Login

Crie uma URL para a view login:

```
path('logar/', views.logar, name='logar'),
```

Crie a view login_view:

```
def logar(request):
    if request.method == "GET":
        return render(request, 'logar.html')
```

Agora crie o logar.html:

```
{% extends "base.html" %}
{% load static %}
{% block 'head' %}

<link rel="stylesheet" href="{% static 'usuarios/css/usuarios.css' %}">

{% endblock 'head' %}
```

```
{% block 'body' %}

<div class="container-fluid">
  <div class="row">
    <div class="col-md-2 d-flex justify-content-center bg-img ">
    </div>
    <div class="col-md-10 d-flex justify-content-center bg-color-dark form-auth">
      <div class="container-form">
        <h1 class="text-logo">START.SE</h1>
        <h1>Logar</h1>
        {% if messages %}
          {% for message in messages %}
            <section class="alert {{message.tags}}">
              {{message}}
            </section>
          {% endfor %}
        {% endif %}
        <br>
        <form action="#" method="">
          <label>Username</label>
          <input type="text" class="form-control input-pers" name="username" placeh
          <br>
          <label>Senha</label>
          <input type="password" class="form-control input-pers" name="senha" place
          <div class="centralizar">
            <input type="submit" class="btn btn-indigo" value="Logar">
            <a href="" class="texto">Não possui uma conta</a>
          </div>
        </form>
      </div>
    </div>
  </div>
</div>

{% endblock 'body' %}
```

Redirecione entre as páginas!

Envie os dados do formulário para a view:

```
<form action="{% url 'logar' %}" method="POST">{% csrf_token %}
```

Crie a lógica para view:

```
def logar(request):
    if request.method == "GET":
        return render(request, 'logar.html')
    elif request.method == "POST":
        username = request.POST.get('username')
        senha = request.POST.get('senha')
```

```

user = auth.authenticate(request, username=username, password=senha)

if user:
    auth.login(request, user)
    return redirect('/home') # Vai dar erro

messages.add_message(request, constants.ERROR, 'Usuario ou senha inválidos')
return redirect('/usuarios/logar')

```

▼ Cadastro de empresas

Crie um app para empresarios:

```
python3 manage.py startapp empresarios
```

Cadastre o APP!!

Em `models.py` crie a tabela no banco de dados:

```

class Empresas(models.Model):
    tempo_existencia_choices = (
        ('-6', 'Menos de 6 meses'),
        ('+6', 'Mais de 6 meses'),
        ('+1', 'Mais de 1 ano'),
        ('+5', 'Mais de 5 anos')
    )

    estagio_choices = (
        ('I', 'Tenho apenas uma ideia'),
        ('MVP', 'Possuo um MVP'),
        ('MVPP', 'Possuo um MVP com clientes pagantes'),
        ('E', 'Empresa pronta para escalar'),
    )

    area_choices = (
        ('ED', 'Ed-tech'),
        ('FT', 'Fintech'),
        ('AT', 'Agrotech'),
    )

    user = models.ForeignKey(User, on_delete=models.DO_NOTHING)
    nome = models.CharField(max_length=50)
    cnpj = models.CharField(max_length=30)
    site = models.URLField()
    tempo_existencia = models.CharField(max_length=2, choices=tempo_existencia_choices, default='')
    descricao = models.TextField()
    data_final_captacao = models.DateField()
    percentual_equity = models.IntegerField() # Percentual esperado
    estagio = models.CharField(max_length=4, choices=estagio_choices, default='I')
    area = models.CharField(max_length=3, choices=area_choices)
    publico_alvo = models.CharField(max_length=3)
    valor = models.DecimalField(max_digits=9, decimal_places=2) # Valor total a ser vendido
    pitch = models.FileField(upload_to='pitches')
    logo = models.FileField(upload_to='logo')

    def __str__(self):
        return f'{self.user.username} | {self.nome}'

```

Adicione no admin:

```
from .models import Empresas
# Register your models here.

admin.site.register(Empresas)
```

Crie um super usuário:

```
python3 manage.py createsuperuser
```

No Core do projeto crie uma URL que redirecione para o app empresarios:

```
path('empresarios/', include('empresarios.urls')),
```

Crie o `urls.py` de empresarios:

```
from django.urls import path
from . import views

urlpatterns = [
    path('cadastrar_empresa/', views.cadastrar_empresa, name="cadastrar_empresa"),
]
```

crie a view `cadastrar_empresa`:

```
def cadastrar_empresa(request):
    if request.method == "GET":
        return render(request, 'cadastrar_empresa.html')
```

Crie o HTML `cadastar_empresa`:

```
{% extends "base.html" %}
{% load static %}
{% block 'head' %}

    <link rel="stylesheet" href="{% static 'usuarios/css/usuarios.css' %}">

{% endblock 'head' %}

{% block 'body' %}

    <div class="container">
        <br>
        <br>
        {% if messages %}
            {% for message in messages %}
                <section class="alert {{message.tags}}">
                    {{message}}

                </section>
            {% endfor %}
        {% endif %}
    </div>
```

```

<div class="row">
  <div class="col-md">
    <form action="" method="">
      <label>Nome</label>
      <input type="text" name="nome" placeholder="Digite o nome da sua empresa .
      <br>
      <label>Cnpj</label>
      <input type="text" name="cnpj" placeholder="Digite o seu CNPJ ..." class="f
      <br>
      <label>Site</label>
      <input type="text" name="site" placeholder="Digite o seu site ..." class="f
      <br>
      <label>A quanto tempo a startup existe ?</label>
      <select name="tempo_existencia" class="form-select">
        <option value="">T 1</option>
      </select>
      <br>
      <label>Descreva a sua empresa</label>
      <textarea name="descricao" class="form-control"></textarea>
      <br>
      <label>Data final para captação</label>
      <input type="date" name="data_final" class="form-control">
      <br>
      <label>Qual a porcentagem (%) de equity oferecida em contrapartida ao valor
      <input type="number" name="percentual_equity" class="form-control">
      <input type="submit" class="btn btn-lg btn-indigo" value="cadastrar">

    </div>
    <div class="col-md">
      <br>
      <label>Qual o estágio da empresa ?</label>
      <br>
      <br>
      <input type="checkbox" name="estagio" value="I"><span>&nbsp;&nbsp;  Tenho aper
      <br>
      <input type="checkbox" name="estagio" value="MVP"><span>&nbsp;&nbsp;  Possui u
      <br>
      <input type="checkbox" name="estagio" value="MVPP"><span>&nbsp;&nbsp;  Possui
      <br>
      <input type="checkbox" name="estagio" value="E"><span>&nbsp;&nbsp;  Empresa p
      <br>
      <br>
      <label>Qual área sua startup está inseria ?</label>
      <select name="area" class="form-select">

        <option value="">Area 1</option>

      </select>
      <br>
      <label>Qual seu público alvo</label>
      <select name="publico_alvo" class="form-select">
        <option value="BTC">BTC</option>
        <option value="B2B">B2B</option>
      </select>
      <br>
      <label>Qual valor deseja captar ?</label>
      <input type="number" name="valor" class="form-control">
      <br>

```

```

        <label>Envie o pitch de apresentação</label>
        <input type="file" name="pitch" class="form-control">
        <br>
        <label>Envie a logo da empresa</label>
        <input type="file" name="logo" class="form-control">
        </form>
    </div>
</div>
</div>
{% endblock 'body' %}

```

Permita que somente usuários logados possa acessar essa página:

```

if not request.user.is_authenticated:
    return redirect('/usuarios/logar')

```

Envie os choices no context do render:

```

return render(request, 'cadastrar_empresa.html', {'tempo_existencia': Empresas.tempo_existe

```

Exiba os options dinamicamente:

```

{% for te in tempo_existencia %}
    <option value="{{te.0}}">{{te.1}}</option>
{% endfor %}

```

```

{% for area in areas %}
    <option value="{{area.0}}">{{area.1}}</option>
{% endfor %}

```

Envie os dados do formulário:

```

<form action="{% url 'cadastrar_empresa' %}" method="POST" enctype='multipart/form-data'>{%

```

Na view cadastrar empresa:

```

def cadastrar_empresa(request):
    if request.method == "GET":
        return render(request, 'cadastrar_empresa.html', {'tempo_existencia': Empresas.tempo_existe
    elif request.method == "POST":
        nome = request.POST.get('nome')
        cnpj = request.POST.get('cnpj')
        site = request.POST.get('site')
        tempo_existencia = request.POST.get('tempo_existencia')
        descricao = request.POST.get('descricao')
        data_final = request.POST.get('data_final')
        percentual_equity = request.POST.get('percentual_equity')
        estagio = request.POST.get('estagio')
        area = request.POST.get('area')
        publico_alvo = request.POST.get('publico_alvo')
        valor = request.POST.get('valor')
        pitch = request.FILES.get('pitch')

```

```

logo = request.FILES.get('logo')

try:
    empresa = Empresas(
        user=request.user,
        nome=nome,
        cnpj=cnpj,
        site=site,
        tempo_existencia=tempo_existencia,
        descricao=descricao,
        data_final_captacao=data_final,
        percentual_equity=percentual_equity,
        estagio=estagio,
        area=area,
        publico_alvo=publico_alvo,
        valor=valor,
        pitch=pitch,
        logo=logo
    )
    empresa.save()
except:
    messages.add_message(request, constants.ERROR, 'Erro interno do sistema')
    return redirect('/empresarios/cadastrar_empresa')

messages.add_message(request, constants.SUCCESS, 'Empresa criada com sucesso')
return redirect('/empresarios/cadastrar_empresa')

```

▼ Barra de navegação

Crie o arquivo parcial da barra de navegação:

```

<nav class="navbar navbar-expand-lg navbar-dark dark-color">
  <div class="container-fluid">
    <div class="collapse navbar-collapse justify-content-center" id="navbarNav">
      <ul class="navbar-nav">
        <li class="nav-item">
          <a class="nav-link active" aria-current="page" href="">Marketplace</a>
        </li>
        <li class="nav-item">
          <a class="nav-link active" href="{% url 'cadastrar_empresa' %}">Quero captar<
        </li>
      </ul>
    </div>
  </div>
</nav>

```

Inclua a barra de navegação na aba de cadastrar empresas:

```
{% include "partials/barra_navegacao.html" %}
```