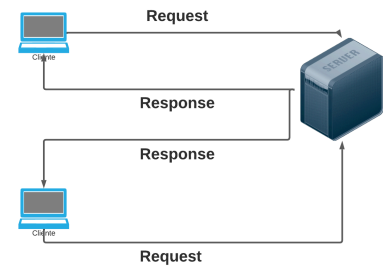


Web Full-Stack | Projeto 2

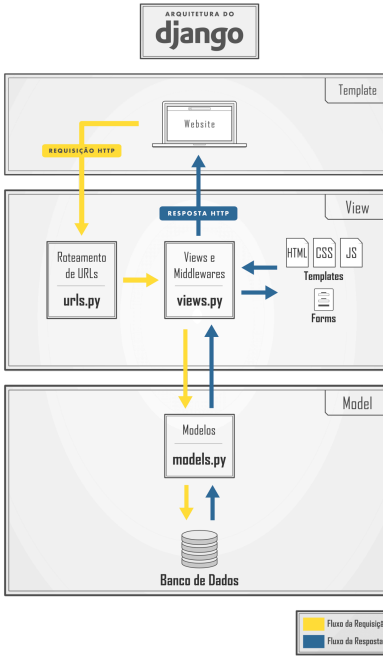
Acesse diretamente pelo Notion:

<https://grizzly-amaranthus-f6a.notion.site/Web-Full-Stack-Projeto-2-1936cf8ea89f80f6af81d7bcc3a5c274?pvs=4>

▼ Teoria



Fluxo de dados no Django:



▼ 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 usuário:

```
python manage.py startapp pacientes
```

Ative o auto-save

INSTALE O APP!

▼ Pacientes

Crie uma URL em core.urls para redirecionar para o APP criado:

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

Agora em pacientes.urls crie a url:

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

urlpatterns = [
    path('', views.pacientes, name='pacientes'),
]
```

Desenvolva a view responsável por exibir o HTML de pacientes:

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

Configure os arquivos estáticos e de media:

```
import os

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 um arquivo HTML para servir de base:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <script src="https://unpkg.com/@tailwindcss/browser@4"></script>
  <{% block 'head' %}{% endblock 'head' %}>
</head>
<body>
  <header class="bg-slate-50">
    <nav class="flex items-center justify-between p-6 lg:px-8 aria-label="Global">
      <div class="flex lg:flex-1">
        <a href="#" class="mr-1.5 p-1.5">
          
        </a>
      </div>
      <div class="flex lg:hidden">
        <button type="button" class="m-2.5 inline-flex items-center justify-center rounded-md p-2.5 text-gray-700">
          <span class="sr-only"> Open main menu </span>
          <svg class="size-6 fill-none viewBox="0 0 24 24" stroke-width="1.5" stroke="currentColor" aria-hidden="true" data-slot="icon">
            <path stroke-linecap="round" stroke-linejoin="round" d="M3.75 6.75h16.5m-18.5 5.25h16.5"/>
          </svg>
        </button>
      </div>
    </nav>
```

```
</button>
</div>
<div class="hidden lg:flex lg:gap-x-12">
  <a href="{% url 'pacientes' %}" class="text-sm/6 font-semibold text-gray-900">Pacientes</a>
</div>
<div class="hidden lg:flex lg:flex-1 lg:justify-end">

</div>
</nav>

</header>

{% block 'body' %}{% endblock 'body' %}
</body>
</html>
```

Agora crie o HTML de pacientes:

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

{% block 'body' %}
<div class="relative isolate">
  <svg class="pointer-events-none -z-50 absolute inset-x-0 top-0 h-[64rem] w-full stroke-gray-200 [mask-image:radial-gradient(32rem 32rem_at_center,white,transparent)]" aria-hidden="true">
    <defs>
      <pattern id="f932ae7-37de-4c0a-a8b0-a6e3b4d44b84" width="200" height="200" x="50%" y="-1" patternUnits="userSpaceOnUse">
        <path d="M-200 0h200v20h-20z M600 0h20v20h-20z M-400 600h20v20h-20z M200 600h20v20h-20z" stroke-width="0" />
      </pattern>
    </defs>
    <svg x="50%" y="-1" class="overflow-visible fill-gray-50">
      <path d="M-200 0h201v20h-20z M600 0h20v20h-20z M-400 600h20v20h-20z M200 600h20v20h-20z" stroke-width="0" />
    </svg>
    <rect width="100%" height="100%" stroke-width="0" fill="url(#f932ae7-37de-4c0a-a8b0-a6e3b4d44b84)" />
    </svg>
    <div class="pointer-events-none absolute left-1/2 right-0 top-0 -z-10 -ml-24 transform-gpu overflow-hidden blur-3xl lg:ml-24 xl:ml-48" aria-hidden="true">
      <div class="aspect-[801/1036] w-[50.0625rem] bg-gradient-to-tr from-[#ff80b5] to-[#9089fc] opacity-30" style="clip-path: polygon(63.1% 29.5%, 100% 17.1%, 76.6% 3%, 48.4% 0%, 44.6% 4.7%, 54.5% 25.3%, 59.8% 49%, 55.2% 57.8%, 44.4% 57.2%, 27.8% 47.9%, 35.1% 81.5%, 0% 97.7%, 39.2% 100%, 35.2% 81.4%, 97.2% 52.8%, 63.1% 29.5%)"></div>
    </div>

<div class="py-24 sm:py-32">
  <div class="mx-auto grid max-w-7xl gap-20 px-6 lg:px-8 xl:grid-cols-3">
    <div class="max-w-xl">

      <h2 class="text-pretty text-3xl font-semibold tracking-tight text-gray-900 sm:text-4xl">Cadastro pacientes</h2>

      <form action="" method="">
        <label for="nome" class="block mt-2 text-sm/6 font-medium text-gray-600">Nome</label>
        <div class="mt-2">
          <input type="text" name="nome" id="nome" autocomplete="given-name" class="block w-full rounded-md bg-white px-3 py-1.5 text-base text-gray-900 outline outline-1 -outline-offset-1 outline-gray-300 placeholder:text-gray-400 focus:outline focus:outline-2 focus:-outline-offset-2 focus:outline-indigo-600 sm:text-sm/6">
        </div>

        <label for="email" class="block mt-2 text-sm/6 font-medium text-gray-600">Email</label>
        <div class="mt-2">
          <input type="email" name="email" id="email" class="block w-full rounded-md bg-white px-3 py-1.5 text-base text-gray-900 outline outline-1 -outline-offset-1 outline-gray-300 placeholder:text-gray-400 focus:outline focus:outline-2 focus:-outline-offset-2 focus:outline-indigo-600 sm:text-sm/6">
        </div>

        <label for="telefone" class="block mt-2 text-sm/6 font-medium text-gray-600">Telefone</label>
        <div class="mt-2">
          <input type="text" name="telefone" id="telefone" class="block w-full rounded-md bg-white px-3 py-1.5 text-base text-gray-900 outline outline-1 -outline-offset-1 outline-gray-300 placeholder:text-gray-400 focus:outline focus:outline-2 focus:-outline-offset-2 focus:outline-indigo-600 sm:text-sm/6">
        </div>

        <div class="grid grid-cols-2 gap-2">
          <div>
            <label for="telefone" class="block mt-2 text-sm/6 font-medium text-gray-600">Queixa</label>
            <div class="mt-2">
              <select name="queixa" class="block w-full rounded-md bg-white px-3 py-2 text-base text-gray-900 outline outline-1 -outline-offset-1 outline-gray-300 placeholder:text-gray-400 focus:outline focus:outline-2 focus:-outline-offset-2 focus:outline-indigo-600 sm:text-sm/6">
                <option value="">TDAH</option>

              </select>
            </div>
          </div>
          <div>
            <label for="foto" class="block mt-2 text-sm/6 font-medium text-gray-600">Foto</label>
            <div class="mt-2">
              <input type="file" required name="foto" id="foto" class="block w-full rounded-md bg-white px-3 py-1.5 text-base text-gray-900 outline outline-1 -outline-offset-1 outline-gray-300 placeholder:text-gray-400 focus:outline focus:outline-2 focus:-outline-offset-2 focus:outline-indigo-600 sm:text-sm/6">
            </div>
          </div>
        </div>

        <div>
          <button type="submit" class="inline-flex justify-center rounded-md bg-indigo-600 mt-4 cursor-pointer px-3 py-2 w-full text-sm font-semibold text-white shadow-sm hover:bg-indigo-500 focus-visible:outline focus-visible:outline-2 focus-visible:outline-offset-2 focus-visible:outline-indigo-600">Cadastrar paciente</button>
        </div>
      </form>
    </div>
    <ul role="list" class="grid gap-x-8 gap-y-12 sm:grid-cols-2 sm:gap-y-16 xl:col-span-2">
      <li>
        <div class="flex items-center gap-x-6">
          <img class="size-16 rounded-full" src="" alt="">
          <div>
            <a href="#" class="text-base/7 font-semibold tracking-tight text-gray-900">Caio Sampaio</a>
            <p class="text-sm/6 font-semibold text-indigo-600">TDAH</p>
          </div>
        </div>
      </li>
    </ul>
  </div>
</div>
```

Vamos agora criar a tabela no banco de dados:

```
class Pacientes(models.Model):
    queixa_choices = (
        ('TDAH', 'TDAH'),
        ('D', 'Depressão'),
        ('A', 'Ansiedade'),
        ('TAG', 'Transtorno de ansiedade generalizada')
    )

    nome = models.CharField(max_length=255)
    email = models.EmailField(null=True, blank=True)
    telefone = models.CharField(max_length=255, null=True, blank=True)
    queixa = models.CharField(max_length=4, choices=queixa_choices, default='TDAH')
    foto = models.ImageField(upload_to='fotos')
    pagamento_em_dia = models.BooleanField(default=True)

    def __str__(self):
        return self.nome
```

Execute as migrações!

Liste as possíveis queixas a serem selecionadas:

```
{% for queixa in queixas %}
  <option value="{ {{queixa.0}} }">{{queixa.1}}</option>
{% endfor %}
```

Envie os dados do FORM para o BACK-END:

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

Altere a view para processar e salvar os dados:

```
def pacientes(request):
    if request.method == "GET":
        pacientes_list = Pacientes.objects.all()
        return render(request, 'pacientes.html', {'queixas': Pacientes.queixa_choices, 'pacientes': pacientes_list})
    else:
        nome = request.POST.get('nome')
        email = request.POST.get('email')
        telefone = request.POST.get('telefone')
        queixa = request.POST.get('queixa')
        foto = request.FILES.get('foto')

        if len(nome.strip()) == 0 or not foto:
            messages.add_message(request, constants.ERROR, 'O campo nome e foto são obrigatórios')
            return redirect('pacientes')

        paciente = Pacientes(
            nome=nome,
            email=email,
            telefone=telefone,
            queixa=queixa,
            foto=foto
        )
        paciente.save()

        messages.add_message(request, constants.SUCCESS, 'Paciente adicionado com sucesso')
        return redirect('pacientes')
```

Configure as mensagens:

```
from django.contrib.messages import constants

MESSAGE_TAGS = {
    constants.SUCCESS: 'bg-green-50 text-green-700',
    constants.ERROR: 'bg-red-50 text-red-700',
}
```

Exiba as mensagens no HTML

```
{% if messages %}
  {% for message in messages %}
    <div class="rounded-md {{message.tags}} p-4">
      <div class="flex">

        <div class="ml-3">
```

```
<form action="{% url 'atualizar_paciente' paciente.id %}" method="POST"> {% csrf_token %}
```

Por fim, marque o `option` como `selected`:

```
<option value="ativo" {% if paciente.pagamento_em_dia %}selected{% endif %}>Ativo</option>
<option value="inativo" {% if not paciente.pagamento_em_dia %}selected{% endif %}>Inativo</option>
```

▼ Consultas

Crie as tabelas no banco de dados para Tarefas e Consultas:

```
class Tarefas(models.Model):
    frequencia_choices = (
        ('D', 'Diário'),
        ('1S', '1 vez por semana'),
        ('2S', '2 vezes por semana'),
        ('3S', '3 vezes por semana'),
        ('N', 'Ao necessitar')
    )
    tarefa = models.CharField(max_length=255)
    instrucoes = models.TextField()
    frequencia = models.CharField(max_length=2, choices=frequencia_choices, default='D')

    def __str__(self):
        return self.tarefa

class Consultas(models.Model):
    humor = models.PositiveIntegerField()
    registro_geral = models.TextField()
    video = models.FileField(upload_to="video")
    tarefas = models.ManyToManyField(Tarefas)
    paciente = models.ForeignKey(Pacientes, on_delete=models.CASCADE)
    data = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return self.paciente.nome
```

Filtre todas as tarefas para exibir no formulário:

```
tarefas = Tarefas.objects.all()
```

Liste as tarefas no form:

```
{% for tarefa in tarefas %}
<option value="{{tarefa.id}}">{{tarefa}}</option>
{% endfor %}
```

Crie o else para o POST em paciente_view:

```
def paciente_view(request, id):
    paciente = Pacientes.objects.get(id=id)
    if request.method == "GET":
        tarefas = Tarefas.objects.all()
        return render(request, 'paciente.html', {'tarefas': tarefas, 'paciente': paciente})
    else:
        humor = request.POST.get('humor')
        registro_geral = request.POST.get('registro_geral')
        video = request.FILES.get('video')
        tarefas = request.POST.getlist('tarefas')

        consultas = Consultas(
            humor=int(humor),
            registro_geral=registro_geral,
            video=video,
            paciente=paciente
        )
        consultas.save()

        for i in tarefas:
            tarefa = Tarefas.objects.get(id=i)
            consultas.tarefas.add(tarefa)

        consultas.save()

        messages.add_message(request, constants.SUCCESS, 'Registro de consulta adicionado com sucesso.')
        return redirect(f'/pacientes/{id}')
```

Para listar as consultas busque no banco de dados:

```
consultas = Consultas.objects.filter(paciente=paciente)
```

Exiba no HTML:

```
{% for consulta in consultas %}
<li class="relative flex items-center space-x-4 py-4">
<div class="min-w-0 flex-auto">
<div class="flex items-center gap-x-3">
<div class="flex-none rounded-full p-1 {% if consulta.humor >= 3 %}bg-green-400/10 text-green-400{% else %}bg-red-400/10 text-red-400{% endif %}>
<div class="size-2 rounded-full bg-current"></div>
</div>
<h2 class="min-w-0 text-sm/6 font-sembold text-slate-800">
<a href="#" class="flex gap-x-2">
<span class="truncate">30/02/2024</span>
</a>
</h2>
</div>
<div class="mt-3 flex items-center gap-x-2.5 text-xs text-gray-400">
<p class="truncate">{{consulta.lnk_publico}}</p>
<p class="whitespace-nowrap">{{consulta.views}}</p>
</div>
</div>
<a href="#" class="flex gap-x-2">
<div class="flex-none rounded-full bg-red-400/10 px-2 py-1 text-xs font-medium text-red-400 ring-1 ring-inset ring-red-400/30">excluir</div>
<svg class="size-5 flex-none text-gray-400 viewBox: 0 0 20 20 fill: currentColor aria-hidden="true" data-slot="icon">
<path fill-rule="evenodd" d="M8.22 5.22a 7.5 7.5 0 1 1 0 10.66 4.25 4.25a 7.5 7.5 0 1 1 0 10.66 11.94 10 8.22 6.28a 7.5 7.5 0 1 0 10.66 2" clip-rule="evenodd" />
</svg>
</a>
</li>
{% endfor %}
```

Na etapa de excluir consulta, crie a URL:

```
path('excluir_consulta/<int:id>', views.excluir_consulta, name="excluir_consulta"),
```

Agora a view:

```
def excluir_consulta(request, id):
    consulta = Consultas.objects.get(id=id)
    consulta.delete()
    return redirect(f'/pacientes/{consulta.paciente.id}')
```

No HTML redirecione o botão de excluir:

```
{% url 'excluir_consulta' consulta.id %}
```

▼ Consulta pública

Crie uma URL para que o paciente possa ver sua consulta:

```
path('consulta_publica/<int:id>', views.consulta_publica, name="consulta_publica"),
```

A view:

```
def consulta_publica(request, id):
    consulta = Consultas.objects.get(id=id)
    if not consulta.paciente.pagamento_em_dia:
        raise Http404

    return render(request, 'consulta_publica.html', {'consulta': consulta})
```

O HTML:

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

{% block "body" %}

<div class="relative isolate">
<svg class="pointer-events-none -z-50 absolute inset-x-0 top-0 h-[64rem] w-full stroke-gray-200 [mask-image:radial-gradient(32rem 32rem at center,white,transparent)]" aria-hidden="true">
<defs>
<pattern id="f932ae7-37de-4c0a-a8b0-a6e3b4d44b84" width="200" height="200" x="50%" y="-1" patternUnits="userSpaceOnUse">
<path d="M.5 200V.5H200" fill="none" />
</pattern>
</defs>
<svg x="50%" y="-1" class="overflow-visible fill-gray-50">
<path d="M-200 0h201v201h-201zM600 0h201v201h-201zM1200 0h201v201h-201zM1800 0h201v201h-201zM2400 0h201v201h-201zM3000 0h201v201h-201zM3600 0h201v201h-201zM4200 0h201v201h-201zM4800 0h201v201h-201zM5400 0h201v201h-201zM6000 0h201v201h-201zM6600 0h201v201h-201zM7200 0h201v201h-201zM7800 0h201v201h-201zM8400 0h201v201h-201zM9000 0h201v201h-201zM9600 0h201v201h-201zM10200 0h201v201h-201zM10800 0h201v201h-201zM11400 0h201v201h-201zM12000 0h201v201h-201zM12600 0h201v201h-201zM13200 0h201v201h-201zM13800 0h201v201h-201zM14400 0h201v201h-201zM15000 0h201v201h-201zM15600 0h201v201h-201zM16200 0h201v201h-201zM16800 0h201v201h-201zM17400 0h201v201h-201zM18000 0h201v201h-201zM18600 0h201v201h-201zM19200 0h201v201h-201zM19800 0h201v201h-201zM20400 0h201v201h-201zM21000 0h201v201h-201zM21600 0h201v201h-201zM22200 0h201v201h-201zM22800 0h201v201h-201zM23400 0h201v201h-201zM24000 0h201v201h-201zM24600 0h201v201h-201zM25200 0h201v201h-201zM25800 0h201v201h-201zM26400 0h201v201h-201zM27000 0h201v201h-201zM27600 0h201v201h-201zM28200 0h201v201h-201zM28800 0h201v201h-201zM29400 0h201v201h-201zM30000 0h201v201h-201" stroke-width="0" />
</svg>
<rect width="100%" height="100%" stroke-width="0" fill="url(#f932ae7-37de-4c0a-a8b0-a6e3b4d44b84)" />
</svg>
<div class="pointer-events-none absolute left-1/2 right-0 top-0 -z-10 -ml-24 transform-gpu overflow-hidden blur-3xl lg:ml-48" aria-hidden="true">
<div class="aspect-[801/1036] w-[50.0625rem] bg-gradient-to-tr from-[#f8b05] to-[#f08080] opacity-30" style="clip-path: polygon(83.1% 29.5%, 100% 100%, 76.6% 3%, 48.4% 0%, 44.6% 4.7%, 54.5% 25.3%, 59.8% 49%, 55.2% 57.8%, 44.4% 57.2%, 27.8% 47.9%, 35.1% 81.5%, 0% 97.7%, 39.2% 100%, 35.2% 81.4%, 97.2% 52.8%, 63.1% 29.5%)"></div>
</div>

<div class="py-14 sm:py-14">
<div class="mx-auto max-w-7xl px-6 lg:px-8">
<div class="mx-auto max-w-2xl lg:mx-0">

<p class="mt-4 text-pretty text-4xl font-sembold tracking-tight text-gray-800 sm:text-4xl">{{consulta.paciente.nome}}</p>
</div>
</div>

<div class="bg-slate-50 h-screen border-t-1 border-slate-200">
<div class="max-w-7xl mx-auto px-6 lg:px-8 mt-6">

<video class="w-full" controls>
<source src="{{(consulta.video.url)}}" type="video/mp4">
</video>

<div class="mt-8">

<div class="mt-8 flow-root">
```

```
<div class="mx-4 my-2 overflow-x-auto sm:mx-6 lg:mx-8">
<div class="inline-block min-w-full py-2 align-middle sm:px-6 lg:px-8">
<table class="min-w-full divide-y divide-gray-300">
  <thead>
    <tr class="divide-x divide-gray-200">
      <th scope="col" class="py-3.5 pl-4 pr-4 text-left text-sm font-sembold text-gray-900 sm:pl-0">Tarefa</th>
      <th scope="col" class="px-4 py-3.5 text-left text-sm font-sembold text-gray-900">descrição</th>
      <th scope="col" class="px-4 py-3.5 text-left text-sm font-sembold text-gray-900">frequencia</th>
    </tr>
  </thead>

  <tbody class="divide-y divide-gray-200 bg-white">
    {% for tarefa in consulta.tarefas.all %}
      <tr class="divide-x divide-gray-200">
        <td class="whitespace-nowrap py-4 pl-4 pr-4 text-sm font-medium text-gray-900 sm:pl-0">{{tarefa.tarefa}}</td>
        <td class="whitespace-nowrap p-4 text-sm text-gray-500">{{tarefa.instrucoes}}</td>
        <td class="whitespace-nowrap p-4 text-sm text-gray-500">{{tarefa.get_frequencia_display}}</td>
      </tr>
    {% endfor %}
  </tbody>
</table>
</div>
</div>
</div>
</div>

</div>

</div>

{% endblock 'body' %}
```

Na model consulta crie uma property para exibir o link público de cada consulta:

```
def link_publico(self):
    return f"http://127.0.0.1:8000(reverse('consulta_publica', kwargs={'id': self.id}))"
```

▼ Visualização

Crie a model para contabilizar as visualizações:

```
class Visualizacoes(models.Model):
    consulta = models.ForeignKey(Consultas, on_delete=models.CASCADE)
    ip = models.GenericIPAddressField()
```

Execute as migrações!

Todas as vezes que a página for carregada, adicione no BD:

```
view = Visualizacoes(
    consulta=consulta,
    ip=request.META['REMOTE_ADDR']
)
view.save()
```

E para finalizar, crie a property:

```
def views(self):
    views = Visualizacoes.objects.filter(consulta=self)
    totais = views.count()
    unicas = views.values('ip').distinct().count()
    return f'({totais}) - {unicas}'
```

▼ Gráfico

Vamos desenvolver o gráfico de humor do paciente.

Primeiro, organize os dados:

```
tuple_grafico = [(str(i.data) for i in consultas), (str(i.humor) for i in consultas)]
```

E por fim o JavaScript:

```
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
<script>
const ctx = document.getElementById('myChart');

new Chart(ctx, {
  type: 'line',
  data: {
    labels: [(tuple_grafico[0][safe])],
    datasets: [{
      label: 'Humor do paciente',
      data: [(tuple_grafico[1][safe])],
      borderWidth: 1
    }]
  },
  options: {
    scales: {
      y: {
        beginAtZero: true
      }
    }
  }
});
</script>
```