

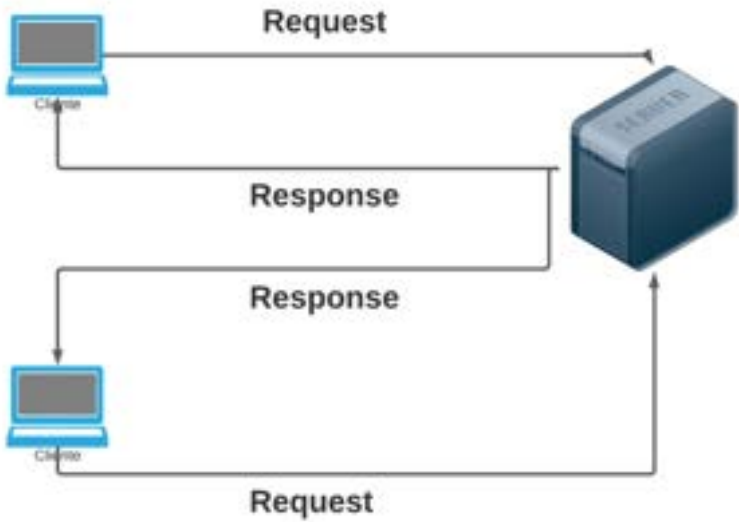


Web Full-Stack

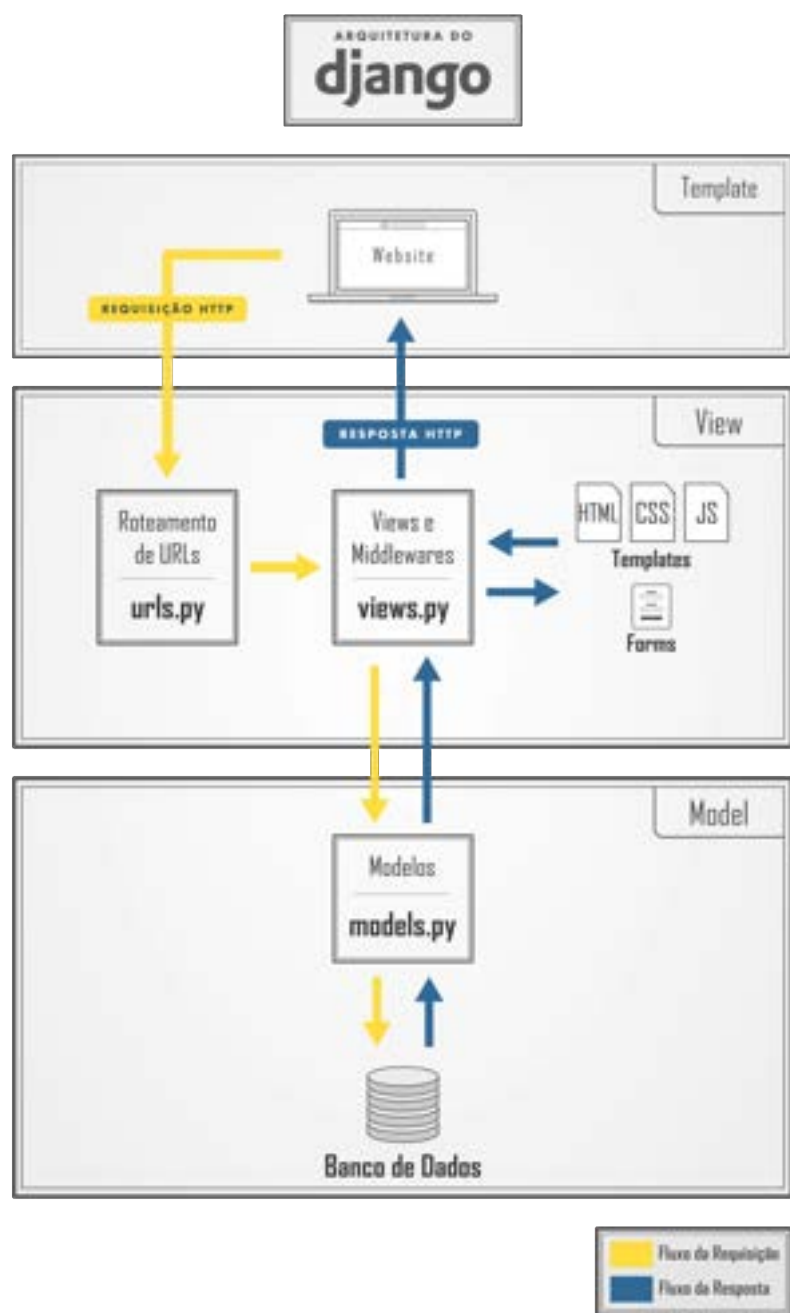
Acesse pelo Notion:

<https://www.notion.so/Web-Full-Stack-12f6cf8ea89f8009ab39cb5078877c20?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 usuario:

```
python manage.py startapp noivos
```

Ative o auto-save

INSTALE O APP!

▼ **Noivos**

Em URL's redirecione para o app criado:

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

Em noivos crie o arquivo `urls.py`:

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

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

Crie a view home:

```
def home(request):
    if request.method == "GET":
        return render(request, 'home.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 o arquivo `templates/base.html`

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <script src="https://cdn.tailwindcss.com"></script>
    <title>Eu.Noiva</title>
</head>
<body>
    {% block 'body' %} {% endblock %}

    <div class="absolute inset-x-0 top-10 -z-10 flex transform-gpu justify-center overflow-hidden blur-3xl"
        aria-hidden="true">
        <div class="aspect-[1108/632] w-[69.25rem] flex-none bg-gradient-to-r from-pink-400 to-purple-600 opacity-20"
            style="clip-path: polygon(43% 17%, 38% 52%, 84% 75%, 68% 21%, 41% 80%, 94% 35%, 17% 29%, 23% 73%, 92% 54%, 62% 83%, 29% 14%, 33% 74%, 3% 57%, 53%
        </div>
    </div>

</body>
</html>
```

Voltando ao `home.html`:

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

{% block 'body' %}
    <div class="max-w-7xl mx-auto mt-12">
        <div class="grid grid-cols-2 gap-4">
            <div class="bg-slate-200/20 p-6 rounded-md">

                <h2 class="text-balance text-xl font-semibold tracking-tight text-gray-900">Crie sua lista de presentes</h2>
                <br>
                <form action="" method="POST">
                    <label class="mt-4 block text-sm leading-6 text-gray-900">Nome do presente</label>
                    <input type="text" name="nome_presente" class="block w-full rounded-md border-0 px-3.5 py-2 text-gray-900 shadow-sm ring-1 ring-inset ring-g

                <div class="flex space-x-4">
                    <div class="w-full">
                        <label class="mt-4 block text-sm leading-6 text-gray-900">Foto de exemplo</label>
                        <input type="file" name="foto" class="block w-full rounded-md border-0 px-3.5 py-2 text-gray-900 shadow-sm ring-1 ring-inset ring-gr

                    </div>

                    <div class="w-full">
                        <label class="mt-4 block text-sm leading-6 text-gray-900">Preço estimado</label>
                        <input type="number" name="preco" class="block w-full rounded-md border-0 px-3.5 py-2 text-gray-900 shadow-sm ring-1 ring-inset ring

                    </div>

                </div>

                <div class="flex space-x-4">

                    <div class="w-full">
                        <label class="mt-4 block text-sm leading-6 text-gray-900">importancia</label>
```

```

        <input type="range" min='1' max="5" name="importancia" class="block w-full py-3 text-gray-900">
      </div>

      <div class="w-full">
        <input type="submit" class="rounded-md w-full mt-4 bg-indigo-500 px-3.5 py-2.5 text-sm font-semibold text-white shadow-sm hover:bg-inc
      </div>

    </div>

  </form>
</div>
<div class="mx-auto">

  <canvas id="myChart" class=""></canvas>

</div>

</div>

<div class="grid grid-cols-5 mt-4 gap-4">

  <div>

    <img class="aspect-[3/2] w-full rounded-2xl object-cover" src="" alt="">

    <h3 class="mt-2 text-lg font-semibold leading-8 text-gray-900">Air fryer</h3>

    <p class="mt-4 block text-sm leading-6 text-gray-900">Reservado por Caio Sampaio</p>

    <p class="text-base leading-7 text-gray-600"></p>

    <span class="inline-flex items-center rounded-md bg-green-400/10 px-2 py-1 text-xs font-medium text-green-400 ring-1 ring-inset ring-green-400">
      <path stroke-linecap="round" stroke-linejoin="round" d="M6 18 18 6M6 6 12 12" />
    </span>

  </div>

</div>

</div>
</div>

{% endblock %}
```

Para o gráfico adicione o JavaScript:

```

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

  new Chart(ctx, {
    type: 'pie',
    data: {
      labels: ['Não reservado', 'Reservado'],
      datasets: [{
        label: 'Presentes',
        data: [5, 6],
        borderWidth: 1
      }]
    },
    options: {
      scales: {
        y: {
          beginAtZero: true
        }
      }
    }
  });
</script>
```

Agora vamos criar um partials/navbar.html para barra de navegação

```

{% load static %}
<nav class="bg-slate-100">
  <div class="mx-auto max-w-7xl">
    <div class="relative flex h-16 items-center justify-between">

      <div class="flex flex-1 items-center justify-center sm:items-stretch sm:justify-start">
        <div class="flex flex-shrink-0 items-center">
          <img class="h-8 w-auto rounded-md" src="">
        </div>
        <div class="hidden sm:ml-6 sm:block">
          <div class="flex">
            <a href="{% url 'home' %}" class="rounded-md px-3 py-2 text-sm font-medium text-gray-900 hover:bg-gray-700 hover:text-white">Lista de presentes</a>
            <a href="" class="rounded-md px-3 py-2 text-sm font-medium text-gray-900 hover:bg-gray-700 hover:text-white">Convidados</a>
          </div>
        </div>
      </div>
    </div>
  </nav>
```

```
        </div>
      </div>
    </div>

    </div>
  </div>
```

```
</nav>
```

Inclua a navbar em base.html:

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

<https://drive.google.com/drive/folders/14T6RxyVd7ExrYUNRTQjEJJ5qYtcmfEE?usp=sharing>

Em templates/static/noivos/img adicione todas as imagens acima.

Adicione a logo na barra de navegação:

```
{% static 'noivos/img/logo.webp' %}
```

▼ Presentes

Crie uma tabela para os presente:

```
class Presentes(models.Model):
    nome_presente = models.CharField(max_length=100)
    foto = models.ImageField(upload_to='presentes')
    preco = models.DecimalField(max_digits=6, decimal_places=2)
    importancia = models.PositiveIntegerField()
    reservado = models.BooleanField(default=False)

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

Execute as migrações

Envie os dados do form para o back-end:

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

Processe os dados na view home:

```
def home(request):
    if request.method == "GET":

        return render(request, 'home.html')
    elif request.method == "POST":
        nome_presente = request.POST.get('nome_presente')
        foto = request.FILES.get('foto')
        preco = request.POST.get('preco')
        importancia = int(request.POST.get('importancia'))

        if importancia < 1 or importancia > 5:
            return redirect('home')

        presentes = Presentes(
            nome_presente = nome_presente,
            foto = foto,
            preco = preco,
            importancia=importancia,
        )

        presentes.save()

        return redirect('home')
```

Para listagem dos presentes busque todos eles nas views:

```
presentes = Presentes.objects.all()
return render(request, 'home.html', {'presentes': presentes})
```

Em home.html liste todos os presentes:

```
{% for presente in presentes %}
  <div>
    {% if presente.foto %}
    
    {% endif %}
    <h3 class="mt-2 text-lg font-semibold leading-8 text-gray-900">{{presente.nome_presente}}</h3>

    <p class="mt-4 block text-sm leading-6 text-gray-900">Reservado por Caio Sampaio</p>

    <p class="text-base leading-7 text-gray-600"></p>
    {% if presente.importancia <= 2 %}
    <span class="inline-flex items-center rounded-md bg-green-400/10 px-2 py-1 text-xs font-medium text-green-400 ring-1 ring-inset ring-green-400/30">Pou
      <path stroke-linecap="round" stroke-linejoin="round" d="M6 18 18 6 12 12" />
    </span>
  </div>
```

```

    </span>
    {% elif presente.importancia < 4 %}
    <span class="inline-flex items-center rounded-md bg-orange-400/10 px-2 py-1 text-xs font-medium text-orange-400 ring-1 ring-inset ring-orange-400/30">
      <path stroke-linecap="round" stroke-linejoin="round" d="M6 18 18 6M6 6 12 12" />
    </span>
    {% else %}
    <span class="inline-flex items-center rounded-md bg-red-400/10 px-2 py-1 text-xs font-medium text-red-400 ring-1 ring-inset ring-red-400/30">Muito imp
      <path stroke-linecap="round" stroke-linejoin="round" d="M6 18 18 6M6 6 12 12" />
    </span>
    {% endif %}

  </div>

{% endfor %}
```

Adicione a URL para os arquivos de media:

```

from django.conf import settings
from django.conf.urls.static import static

urlpatterns = [
    path('admin/', admin.site.urls),
    path('noivos/', include('noivos.urls')),
] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

▼ Lista de convidados

Crie a URL para lista de convidados:

```

path('lista_convidados', views.lista_convidados, name='lista_convidados'),
```

Agora a View lista_convidados:

```

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

Crie o HTML:

```

{% extends "base.html" %}
{% load static %}
{% block 'body' %}
  <div class="max-w-7xl mx-auto mt-12">
    <div class="grid gap-4 grid-cols-2">
      <div class="bg-slate-200/20 p-6 rounded-md">
        <form action="{% url 'lista_convidados' %}" method="POST">{% csrf_token %}
          <h2 class="text-balance text-xl font-semibold tracking-tight text-gray-900">Cradastre seus convidados</h2>
          <br>
          <label class="mt-4 block text-sm leading-6 text-gray-900">Nome do convidado</label>
          <input type="text" name="nome_convidado" class="block w-full rounded-md border-0 px-3.5 py-2 text-gray-900 shadow-sm ring-1 ring-inset ring-gr

        <div class="grid grid-cols-2 gap-2">
          <div>
            <label class="mt-4 block text-sm leading-6 text-gray-900">WhatsApp</label>
            <input type="text" name="whatsapp" class="block w-full rounded-md border-0 px-3.5 py-2 text-gray-900 shadow-sm ring-1 ring-inset ring-

          </div>
          <div>
            <label class="mt-4 block text-sm leading-6 text-gray-900">Máximo de acompanhantes</label>
            <input type="number" name="maximo_acompanhantes" class="block w-full rounded-md border-0 px-3.5 py-2 text-gray-900 shadow-sm ring-1 ri

          </div>
          <input type="submit" class="rounded-md mt-4 bg-indigo-500 px-3.5 py-2.5 text-sm font-semibold text-white shadow-sm hover:bg-indigo-400 foc

        </div>
      </form>
    </div>

    <div>
      <ul role="list" class="divide-y divide-gray-100">
        {% for convidado in convidados %}
          <li class="flex items-center justify-between gap-x-6 py-5">
            <div class="flex min-w-0 gap-x-4">
              
              <div class="min-w-0 flex-auto">
                <p class="text-sm font-semibold leading-6 text-gray-900">{{convidado.nome_convidado}}</p>
                <p class="mt-1 truncate text-xs leading-5 text-gray-500">{{convidado.whatsapp}}</p>
                <p class="mt-1 text-xs leading-5 text-gray-500">{{convidado.link_convite}}</p>
              </div>
            </div>
            <span class="inline-flex items-center rounded-md px-2 py-1 text-xs font-medium ring-1 ring-inset {% if convidado.status == 'AC' %}text

          </li>
        {% endfor %}

      </ul>
    </div>

  </div>
```

```
</div>

{% endblock %}
```

Atualize a navbar:

```
<a href="{% url 'lista_convitados' %}" class="rounded-md px-3 py-2 text-sm font-medium text-gray-900 hover:bg-gray-700 hover:text-white">Convitados</a>
```

Crie a model para os convidados:

```
class Convitados(models.Model):
    status_choices = (
        ('AC', 'Aguardando confirmação'),
        ('C', 'Confirmado'),
        ('R', 'Recusado')
    )

    nome_convitado = models.CharField(max_length=100)
    whatsapp = models.CharField(max_length=25, null=True, blank=True)
    maximo_acompanhantes = models.PositiveIntegerField(default=0)
    token = models.CharField(max_length=25)
    status = models.CharField(max_length=2, choices=status_choices, default='AC')

    def save(self, *args, **kwargs):
        if not self.token:
            self.token = secrets.token_urlsafe(16)
        super(Convitados, self).save(*args, **kwargs)
```

Envie os dados do formulário para o Back-end

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

Processe os dados:

```
def lista_convitados(request):
    if request.method == 'GET':
        convidados = Convitados.objects.all()
        return render(request, 'lista_convitados.html', {'convitados': convidados})
    elif request.method == 'POST':
        nome_convitado = request.POST.get('nome_convitado')
        whatsapp = request.POST.get('whatsapp')
        maximo_acompanhantes = int(request.POST.get('maximo_acompanhantes'))

        convidados = Convitados(
            nome_convitado=nome_convitado,
            whatsapp=whatsapp,
            maximo_acompanhantes=maximo_acompanhantes
        )

        convidados.save()

        return redirect('lista_convitados')
```

▼ Convitado

Crie um app para os convidados:

```
python3 manage.py startapp convidados
```

Instale o APP

Crie a URL para convidados:

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

Agora a URL de convidados:

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

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

Crie a views convidados:

```
def convitados(request):
    token = request.GET.get('token')
    convidado = Convitados.objects.get(token=token)
    presentes = Presentes.objects.filter(reservado=False).order_by('-importancia')
    return render(request, 'convitados.html', {'convidado': convidado, 'presentes': presentes, 'token': token})
```

Exiba a URL de cada convidado:

```
@property
def link_convite(self):
```

```
return f'http://127.0.0.1:8000{reverse("convidados"))}?token={self.token}'
```

Crie o convidados.html

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

{% block 'body' %}
  <div class="max-w-7xl mx-auto mt-12">
    <div class="mx-auto max-w-2xl text-center">

      <h2 class="text-balance text-4xl font-semibold tracking-tight text-gray-900 sm:text-5xl">Olá, {{convidado.nome_convidado}}</h2>

      <p class="mt-2 text-lg leading-8 text-gray-600">Confirme sua presença no casamento: <a href="" class="text-green-400">confirmar</a>, <a href="" cl

    </div>
    <br>
    <hr>

    <div class="grid grid-cols-2 gap-4">
      <div>
        <p class="mt-2 text-lg leading-8 text-gray-600">Escolha o presente para os noivos</p>
        <div class="grid grid-cols-3 mt-4 gap-4">

          {% for presente in presentes %}
            <div>
              {% if presente.foto %}
                
              {% endif %}
              <h3 class="mt-2 text-lg font-semibold leading-8 text-gray-900">{{presente.nome_presente}}</h3>
              <p class="text-base leading-7 text-gray-600"></p>
              {% if presente.importancia <= 2 %}
                <span class="inline-flex items-center rounded-md bg-green-400/10 px-2 py-1 text-xs font-medium text-green-400 ring-1 ring-inset ri
                  <path stroke-linecap="round" stroke-linejoin="round" d="M6 18 18 6M6 6l12 12" />
                </svg>
              </span>
              {% elif presente.importancia < 4 %}
                <span class="inline-flex items-center rounded-md bg-orange-400/10 px-2 py-1 text-xs font-medium text-orange-400 ring-1 ring-inset
                  <path stroke-linecap="round" stroke-linejoin="round" d="M6 18 18 6M6 6l12 12" />
                </svg>
              </span>
              {% else %}
                <span class="inline-flex items-center rounded-md bg-red-400/10 px-2 py-1 text-xs font-medium text-red-400 ring-1 ring-inset ring-r
                  <path stroke-linecap="round" stroke-linejoin="round" d="M6 18 18 6M6 6l12 12" />
                </svg>
              </span>
              {% endif %}

              <a href="" class="rounded-md block w-full mt-4 bg-indigo-500 px-3.5 py-2.5 text-sm font-semibold text-white shadow-sm hover:bg-inc

            </div>
          {% endfor %}

        </div>
      </div>
    </div>

    <div class="bg-slate-200/20 p-6 rounded-md mt-4 drop-shadow-lg ring-1 ring-gray-600/20">

      <h2 class="text-balance text-xl font-semibold tracking-tight text-gray-900 text-center">Adicione seus acompanhantes</h2>
      <form action="" method="POST">
        <label class="mt-4 block text-sm leading-6 text-gray-900">Nome</label>
        <input type="text" name="nome" class="block w-full rounded-md border-0 px-3.5 py-2 text-gray-900 shadow-sm ring-1 ring-inset ring-gray-300
        <input type="submit" class="rounded-md mt-4 bg-indigo-500 px-3.5 py-2.5 text-sm font-semibold text-white shadow-sm hover:bg-indigo-400 foc

      </form>
      <br>
      <br>
      <hr>
      <br>
      <ul>

        <li class="bg-slate-100 p-4">
          Caio Sampaio

        </li>
      </ul>
    </div>
  </div>
{% endblock %}
```

Crie a URL para confirmar presença:


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

Crie a view para responder_presenca:

```
def responder_presenca(request):
    resposta = request.GET.get('resposta')
    token = request.GET.get('token')
    convidado = Convidados.objects.get(token=token)
    if resposta not in ['C', 'R']:
        messages.add_message(request, constants.ERROR, 'Você deve confirmar ou recusar')
        return redirect(f'{reverse('convidados')}?token={token}')

    convidado.status = resposta
    convidado.save()

    return redirect(f'{reverse('convidados')}?token={token}')
```

No HTML adicione:

```
{% if convidado.status == 'AC' %}
    <p class="mt-2 text-lg leading-8 text-gray-600">Confirme sua presença no casamento: <a href="{% url 'responder_presenca' %}?resposta=C&token={{token}}" class="font-semibold text-indigo-500">Confirme
{% endif %}
```

▼ Reservar presente

Crie uma URL para reservar o presente:

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

Adicione o campo na model presente:

```
reservado_por = models.ForeignKey(Convidados, null=True, blank=True, on_delete=models.DO_NOTHING)
```

Crie a View reservar presente:

```
def reservar_presente(request, id):
    token = request.GET.get('token')

    convidado = Convidados.objects.get(token=token)
    presente = Presentes.objects.get(id=id)

    presente.reservado=True
    presente.reservado_por = convidado
    presente.save()
    return redirect(f'{reverse('convidados')}?token={token}')
```

Reserve o presente:

```
<a href="{% url 'reservar_presente' presente.id %}?token={{token}}" class="rounded-md block w-full mt-4 bg-indigo-500 px-3.5 py-2.5 text-sm font-semibold text-white">Reservar
```

Calcule os presentes reservados e os não reservados:

```
nao_reservado = Presentes.objects.filter(reservado=False).count()
reservado = Presentes.objects.filter(reservado=True).count()
```

E altere no JS

```
data: {{data|safe}},
```