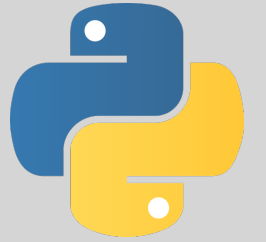




Django



Aplicativo visitantes

- 1 - Criar o aplicativo **visitantes**
- 2 - Registrar o novo aplicativo no projeto

Django



```
class Visitante(models.Model):
    nome_completo = models.CharField(
        verbose_name="Nome Completo",
        max_length=150,
    )

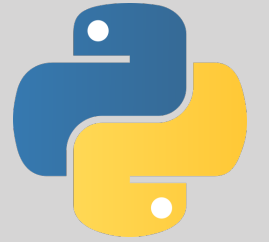
    cpf = models.CharField(
        verbose_name="CPF",
        max_length=11,
    )

    data_nascimento = models.DateField(
        verbose_name="Data de Nascimento",
        auto_now_add=False,
        auto_now=False,
    )

    numero_casa = models.PositiveSmallIntegerField(
        verbose_name="Número da casa a ser visitada",
    )

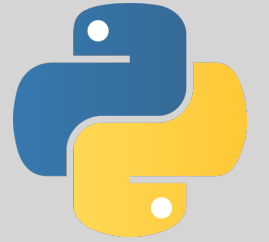
    placa_veiculo = models.CharField(
        verbose_name="Placa do Veículo",
        max_length=7,
        blank=True,
        null=True,
    )
```

Django



```
horario_chegada = models.DateTimeField(  
    verbose_name="Horário de Chegada na Portaria",  
    auto_now_add=True,  
)  
  
horario_saida = models.DateTimeField(  
    verbose_name="Horário de saída do condomínio",  
    auto_now=False,  
    blank=True,  
    null=True,  
)  
  
horario_autorizacao = models.DateTimeField(  
    verbose_name="Horário de Autorização de Entrada",  
    auto_now=False,  
    blank=True,  
    null=True,  
)  
  
morador_responsavel = models.CharField(  
    verbose_name="Nome do morador responsável por autorizar a entrada",  
    max_length=50,  
    blank=True,  
)
```

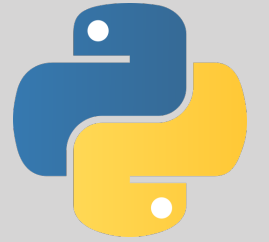
Django



Adicionando uma foreign key

```
registrado_por = models.ForeignKey(  
    "porteiros.Porteiro",  
    verbose_name="Porteiro responsável pelo registro",  
    on_delete=models.PROTECT,  
)  
  
class Meta:  
    verbose_name="Visitante",  
    verbose_name_plural="Visitantes",  
    db_table="visitante"  
  
def __str__(self):  
    return self.nome_completo
```

Django



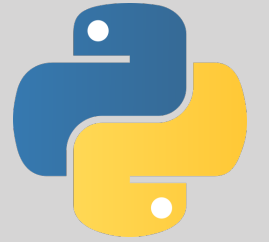
Registrando o aplicativo para o Django

Abrir admin.py

Importar a classe: `from visitantes.models import Visitante`

Adicionar: `admin.site.register(Visitante)`

Django

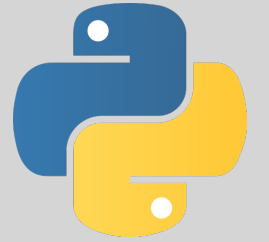


Validando o código escrito e aplicando as alterações no banco de dados

```
python manage.py makemigrations visitantes
```

```
python manage.py migrate
```

Django

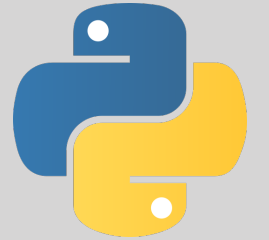


```
python manage.py runserver
```

```
127.0.0.1:8000/admin
```

Adicione um visitante

Django



Adicionando o valor de variáveis em um template

Para exibir valores em um template vamos usar dicionários. Abra o arquivo `views.py` do aplicativo `usuarios`

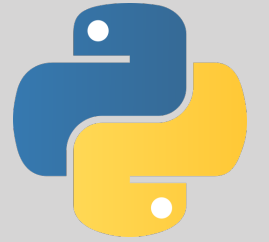
Na função `index` adicione uma variável dicionário

```
context = {  
    "nome_pagina": "Início da Dashboard",  
}
```

Essa variável deve ser passar como parâmetro na função `render`

```
return render(request, "index.html", context)
```

Django



Adicionando o valor de variáveis em um template

Abrir o arquivo index.html

No local onde o valor da variável deve ser impresso

```
{{ nome_pagina }}
```

Django



```
python manage.py runserver
```

```
127.0.0.1:8000
```