

*\*Essa é uma versão dos Guias de Estudo desenvolvida para auxiliar o processo de impressão e para facilitar a leitura dos guias por programas que fornecem a leitura automatizada para suporte a todos os alunos que necessitem. Dessa forma, não apresentaremos ilustrações nesse arquivo. **#ParaTodosLerem**.*

## Módulo #7

### Guias de estudo

### Integrações

*Quando desenvolvermos um programa, precisamos pensar que o mesmo não é composto apenas pelos algoritmos.*

*Observe que os códigos que desenvolvemos em Python, ao finalizar suas execuções, não armazenam os dados gerados para serem utilizados mais tarde, em uma outra execução.*

*Imagine uma aplicação como **WhatsApp**, os dados que foram gerados em algum momento e que não foram excluídos estão disponíveis para uso e visualização.*

*Para que isso se torne possível é preciso utilizar um outro recurso, que são os **bancos de dados**. Ou seja, nos programas implementados os usuário podem inserir dados e estes serão armazenados nos bancos e, depois, o banco de dados fornece as informações para que os usuários possam manipular no programa.*

*Então, é preciso pensar em uma integração entre implementação e banco de dados para que isso funcione.*

*Além disso, é preciso pensar em uma segunda integração. Você já utilizou alguma aplicação pronta em uma IDE que você executava o código?*

*Obviamente não! As aplicações que utilizamos possuem uma ou mais telas para que possamos interagir com o sistema. Vemos isso em sistemas bancários, aplicativos no geral, sistemas de gerenciamento de empresas e outros.*

*Para adicionar estas "telas iterativas" ou seja, o **frontend** dos nossos programas, é preciso fazer uma integração entre **frontend** e o **backend** (nossas implementações em Python).*

A boa notícia é que há muitos frameworks no mercado que nos auxiliam com estas integrações. Alguns deles são baseados em **MVC (Modelo, Visualização e Controle)** ou **MVT (Modelo, Visualização e Template)**.

Vamos ver um exemplo?

O **Django**, que vocês até aqui já conhecem, é baseado em **MVT** e estas camadas correspondem a:

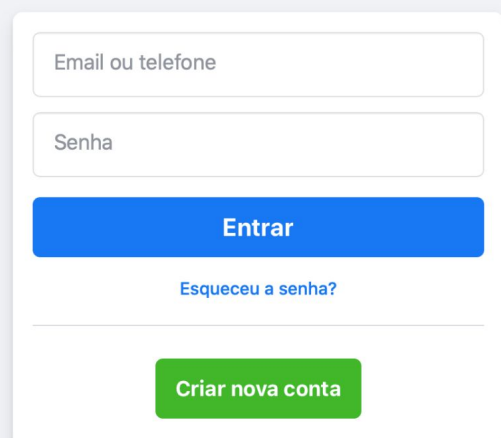
- **Camada Modelo:** Faz a integração entre bancos de dados e programa;
- **Camada Template:** Onde se encontra a lógica de negócio. É aqui que determinamos aquilo que o programa vai executar. Por exemplo, se o usuário clicar no botão deletar, aqui temos a implementação do que vai acontecer quando este botão for acionado.
- **Camada Visualização:** Onde ficam as telas, ou seja, o **frontend**.

Muitos programas que implementamos são baseados em formulários. Estes formulários podem ser compostos por campos que o usuário preenche e cadastra seus dados, esses campos podem ser formulários com visualização de dados já incluídos em que o usuário pode visualizar, alterar ou excluir. Vamos ver um exemplo.

Na página de login do **Facebook** há um formulário para o usuário entrar na plataforma. Este formulário é composto pelos campos de login (email ou telefone) e senha. Além disso, há os botões de Entrar, Esqueceu sua senha ou criar nova conta. Cada um destes botões direciona o usuário a uma funcionalidade diferente do site.

# facebook

O Facebook ajuda você a se conectar e compartilhar com as pessoas que fazem parte da sua vida.

A mockup of the Facebook login interface. It features two input fields: 'Email ou telefone' and 'Senha'. Below these is a blue 'Entrar' button. Underneath the button is a link that says 'Esqueceu a senha?'. At the bottom of the form is a green button labeled 'Criar nova conta'.

Criar uma Página para uma celebridade, uma marca ou uma empresa.

Direitos de imagem reservados à Facebook.com

*Se você for desenvolver algo parecido em **Django** a camada de visualização ficará responsável por determinar qual template buscar a cada iteração do usuário.*

*Por exemplo, se o usuário clicou em Criar nova conta, a camada de visualização faz uma requisição à camada de template que vai retornar o formulário de cadastro.*

*Quando o usuário faz o cadastro, a camada de visualização faz uma requisição a camada Modelo para que os dados sejam cadastrados no banco de dados, esta camada retorna uma mensagem de "salvo com sucesso" ou "usuário já cadastrado" ou algo do tipo. E a camada de visualização interpreta esta mensagem e exibe para o usuário.*

### **Lembre-se**

*Ao criar uma aplicação usando framework **Django** ou similar, organize corretamente toda a implementação em camadas **MVC** ou **MVT** para que seus códigos fiquem organizados. Isso facilitará manutenções futuras nos códigos e deixará sua aplicação mais segura.*