

# SI-tecnologias :) Ltda

Programação - 1  
Projeto de sistemas de informação

# Muita gente precisa de tecnologia.

## Ela está em vários lugares.



# Está no prontuário do médico



# No sistema da padaria



# Na agenda eletrônica dos estudantes



# No jogo da criança e do adulto



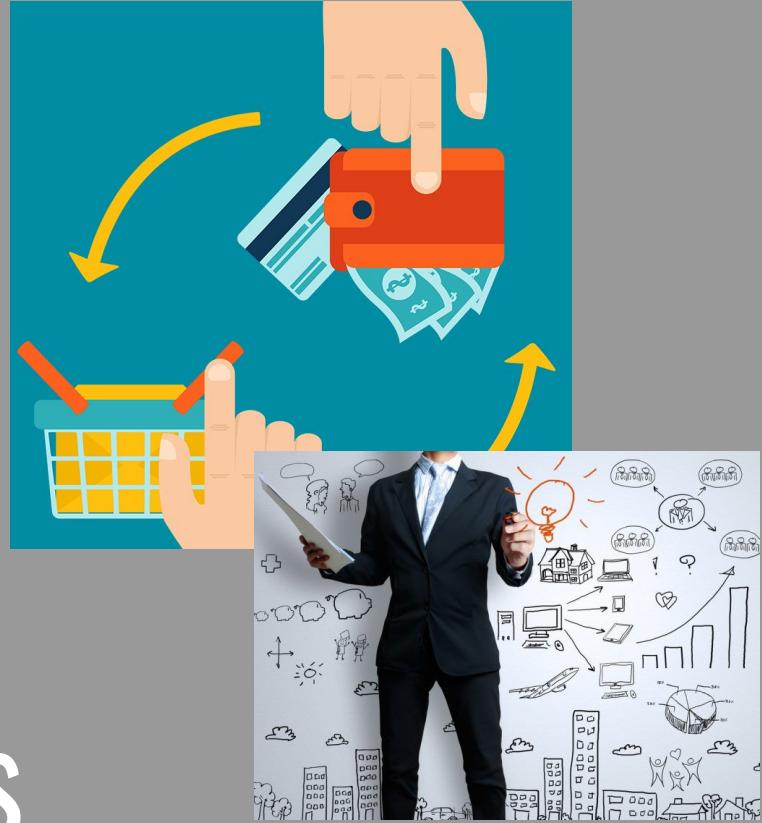
# No sistema de acessibilidade para pessoas especiais



# No restaurante

# Na lanchonete

# Em lojas de vendas



O que voce pode **criar**  
que seja

**útil/interessante/desafiador**

Com o que você **aprendeu** na  
**disciplina?**  
está aprendendo

# O projeto de SI é um projeto aberto



*Mas que tem alguns requisitos mínimos*

**O que você vai resolver?**

# Começando

---

Primeiramente você precisa encontrar um problema.

Em seguida, construir uma ferramenta computacional.

Seu programa deve ter algumas funcionalidades básicas. O resto você vai usar sua criatividade.



HikingArtist.com

*Isso é um nome bonito para "programa"*

Pode ser:  
**Sistema para:**  
Hospitais, hotéis, ONGs,  
construtoras, universidades,  
polícias, jogos, etc...



# Você precisa fazer um sistema de informação:

- Armazenamento
- Atualização
- Remoção
- Busca



## Por exemplo:

Se você estiver desenvolvendo um programa para um hotel, o que você pode querer salvar é uma reserva:

- Uma reserva pode ter as seguintes informações:
  - CPF do contratante
  - nome do contratante
  - data de início
  - data de fim
  - apartamento alocado

Se for um sistema de biblioteca, você pode querer armazenar os livros:

- Nome
- Autor
- Editora
- Ano de lançamento





Na base de dados...  
Você vai precisar manipular informações que vão ser:  
Salvas, Atualizadas, Removidas, Buscadas

# Vamos detalhar os requisitos

# Requisitos Mínimos:

1- Seu programa deverá realizar cadastros. Seja de livros, reservas, vendas, etc. Cada elemento cadastrado deve ter:

= 1 atributo que o identifique unicamente (atributo identificador) + pelo menos 1 atributo qualquer + 1 campo de informação da criação daquele elemento (data/hora) + 1 campo de atualização (data/hora da última modificação).

Por exemplo: Se seu programa armazenar as vendas de sanduiches de Jean: Você pode usar o CPF+DATADECOMPRA como número inteiro para identificar a venda, um campo de CPF para quem foi o comprador, um campo para salvar onde foi essa venda (qual o Centro da UFPE), um campo para armazenar quando foi a compra e um campo para armazenar a última atualização da venda.

2- Seu programa deve ser capaz de remover algum elemento, usando algum atributo do elemento.

3- Seu programa deve ser capaz de atualizar os campos dos elementos armazenados

4- Seu programa deve ser capaz de fazer buscas nos elementos armazenados.

5- Toda vez que o programa encerrar, ele deve atualizar os arquivos com os elementos.

# Requisitos Mínimos:

- 6- Você deve escolher pelo menos 1 requisito PLUS. Se você escolher mais requisitos PLUS, você recebe uma pontuação extra apenas nos requisitos extras ao projeto. No final, seu projeto pode valer por exemplo 11, 12, 13... esse ponto vai ser considerado extra e vai entrar na pontuação da 2a unidade.
- 7- Você deve armazenar as informações no seu programa usando dicionários de tuplas
- 8- Qualquer operação realizada no sistema deve ser armazenada em um arquivo de LOG (quem executou, quando executou, o que executou).
- 9- Você deve ter uma opção de ordenar as informações armazenadas.
- 10- Ter login/logout. Para acessar o sistema, é preciso ser um usuário cadastrado no sistema. O programa deve ter pelo menos dois níveis de acesso às suas funcionalidades do programa. Cada usuário vai possuir um nível de acesso. Por exemplo: gerente: pode cadastrar, remover, buscar, atualizar; estagiário: pode apenas buscar.
- 11 - Gerar relatório: imprima todos os objetos de seu programa em um arquivo num formato legível e de forma organizada (use tabulação, caracteres especiais e explique os dados armazenados).

# Requisitos PLUS:

1- Usar criptografia (*explicado mais na frente*) [vale +0.5]

2- Usar Django [vale +1.0]

3- Usar Interface gráfica (QT, TKinter, pygame) [vale +1.0]

4- Usar planilhas (arquivo csv) para armazenar informações [vale +0.5]

5- Usar biblioteca do Telegram [vale +1.0]

6- Usar alguma biblioteca de inteligência artificial [vale +1.5] (por exemplo, Rede Neural  
[https://scikit-learn.org/stable/modules/neural\\_networks\\_supervised.html](https://scikit-learn.org/stable/modules/neural_networks_supervised.html))

7- Usar socket (comunicação em rede) [vale +1.5]

8- Usar a biblioteca OPENCV [vale +1.5]

**ATENÇÃO:** Você deve escolher 1 requisito PLUS obrigatoriamente.

A partir do segundo requisito, a pontuação extra é válida.

**ATENÇÃO 2:** O uso desses recursos deve ser feito de forma correta. Não é só usar. Ele deve estar em alguma parte do programa que faça sentido. Use a criatividade com segurança. :)

# Funcionalidades do sistema

---

## **Para fazer o login**

Use um arquivo usuarios.txt onde você vai armazenar o login, senha e o nível de acesso dos usuários.

## **Para armazenar os elementos do seu sistema**

Use um arquivo elementos.txt onde você vai armazenar os objetos do seu sistema

**Permita que seja possível alterar o nível de acesso de um usuário dentro do programa.**

# Observações gerais

---

- Todas as funções de Python implementadas no sistema devem estar **documentadas**, utilizando docstring de descrição para funcionar no help() do Python.
- Ao inicializar, o sistema deve carregar os dados existentes nos arquivos **"elementos.txt"** e **"usuarios.txt"** e armazenar em estruturas de dados do tipo dicionário de tuplas para a manipulação durante a execução do programa. Cada usuário e cada elemento deverá ser tratado como uma tupla.

*Por exemplo, se você tiver 2 clientes de um hotel cadastrados no sistema, um dicionário terá que armazenar os dois clientes. Mas cada cliente é representado por uma tupla.*

{111: ("Fernando", 27, "Recife", "Rua dos Reitores"), 777:("Vitor", 22, "Recife", "Avenida Boa Viagem")}, considerando que 111 e 777 são os atributos identificadores dos dois clientes "Fernando" e "Victor" respectivamente.

# Observações gerais

---

- Modularize seu sistema. Escreva uma função para cada funcionalidade existentes. Não repita partes do código que fazem a mesma coisa: crie funções e as chame quando precisar.
- Você não poderá importar nenhuma biblioteca de Python sem orientação do professor ou da monitoria. Para realizar esse projeto, pode usar as funções nativas de Python.

# Para criptografia

---

**Se você quiser implementar a criptografia:**

O texto armazenado no arquivo `elementos.txt` deverá ser criptografado através de uma técnica bem conhecida chamada **RSA**: haverá um arquivo chamado “`chavePublica.txt`” que conterá dois inteiros **e** e **n**. Quando precisar armazenar uma string, você fará, para cada caracter **x** dessa string, o cálculo do seu valor criptografado **y =  $\text{ord}(x)^e \bmod n$** .

Para recuperar a informação, você precisará acessar o arquivo “`chavePrivada.txt`” e acessar o conteúdo de dois inteiros **d** e **n** e fazer para cada valor criptografado **y**, o cálculo do seu valor de origem através da operação **x=  $\text{char}(y^d \bmod n)$** . Para o teste de vocês, considere um exemplo de arquivo “`chavePublica.txt`” contendo **71 1073** (**e** e **n** respectivamente) e “`chavePrivada.txt`” contendo **1079 1073** (**d** e **n** respectivamente).

# Detalhamento das funcionalidades

---

Exemplos de funcionamento:

Se você precisa armazenar: "fernandoneto" e "senha 123" que identificam o nome de usuário e senha de um dado usuário, o conteúdo a ser armazenado no arquivo deve ser os números inteiros criptografados:

855, 566, 321, 110, 251, 110, 158, 111, 110, 566, 348, 111

28, 566, 110, 650, 251, 577, 774, 501, 341

Perceba que aqui o conteúdo de usuário e senha estão em linhas diferentes.

Você pode colocá-los na mesma linha desde que padronize alguma forma de separar as informações. Fique à vontade para padronizar o armazenamento de seus dados da sua forma.

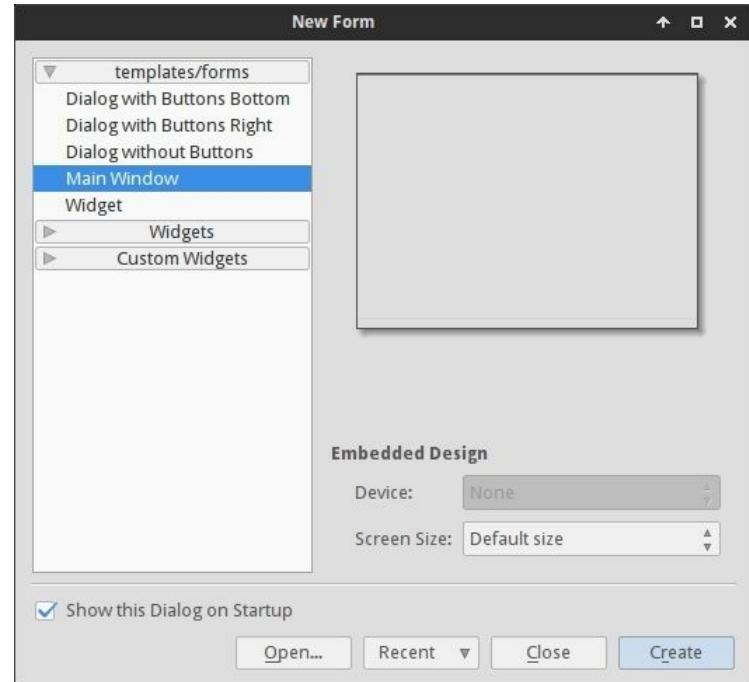
O que você deve obedecer é apenas a criptografia simples das informações.

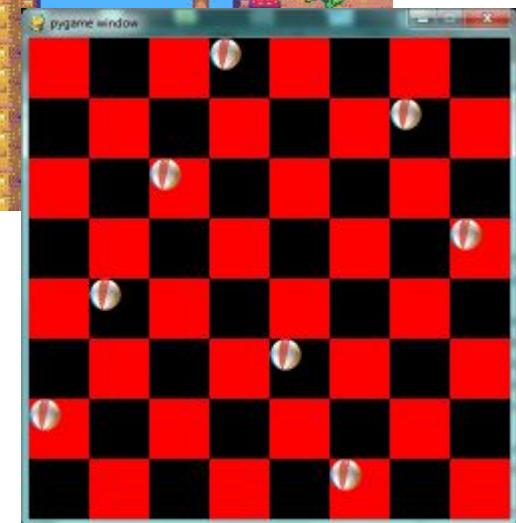
# Interface gráfica

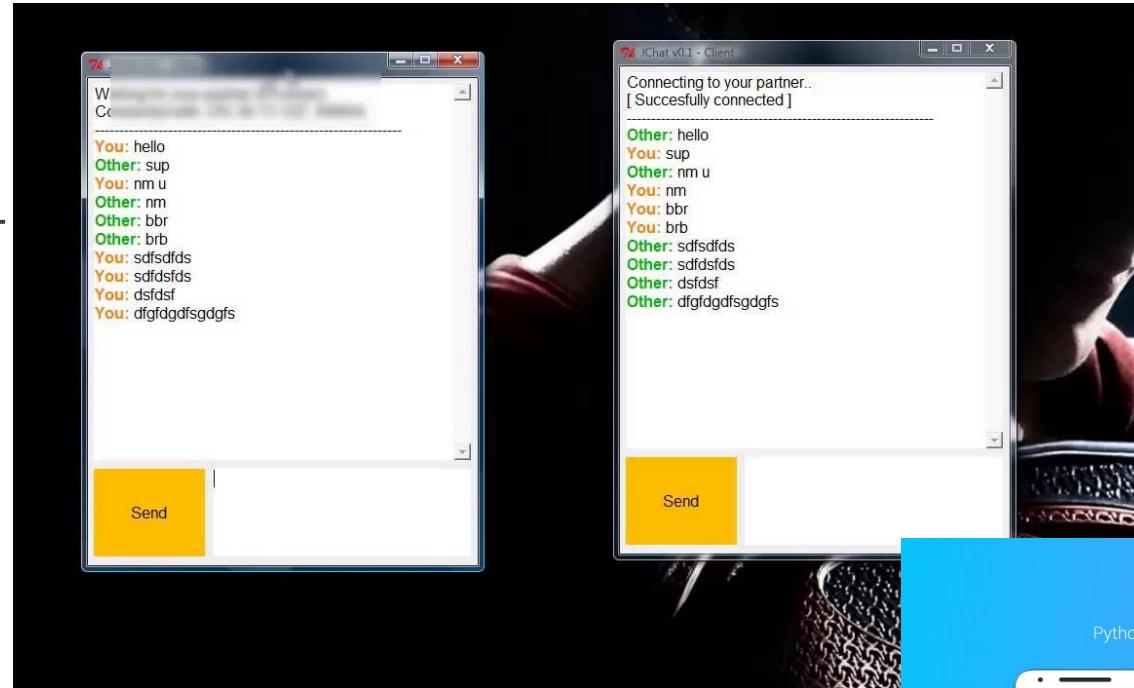
---

Existem vários sistemas de interface compatíveis com o Python

<https://wiki.python.org.br/ComparacaoDeGUIs>





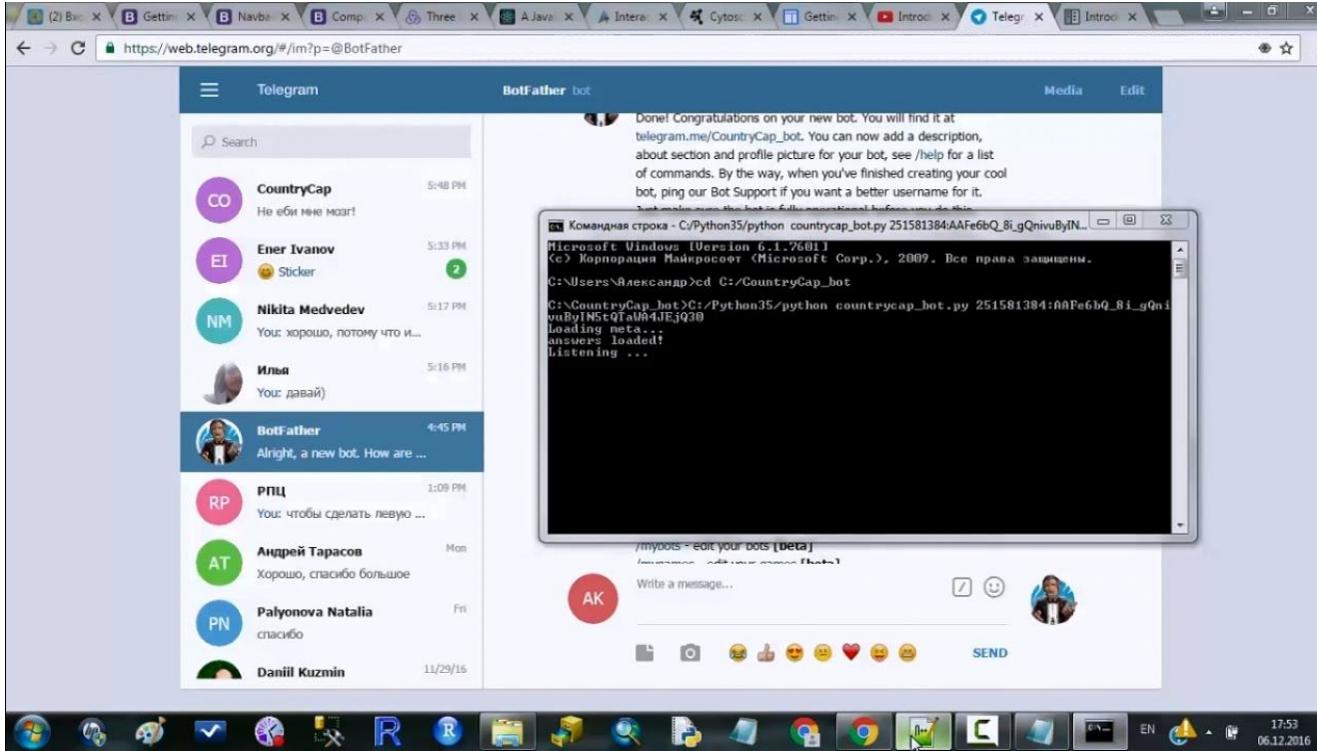


**PyMessenger**  
Python API for Facebook Messenger Platform

The advertisement features three mobile phone screens displaying the PyMessenger interface. The phones are arranged horizontally against a blue background.

- Phone 1:** Shows a Facebook Messenger conversation with a bot named "enginebai.ai". The screen displays a message from the bot about "Swap Points" and a link to "Py".
- Phone 2:** Shows the PyMessenger management interface for the "enginebai.ai" bot. It includes sections for "MANAGE", "Py", and "Typically replies instantly".
- Phone 3:** Shows another view of the PyMessenger management interface, possibly for a different bot or a different section of the platform.

# Telegram Bot



# Critérios gerais de avaliação

---

- Modularização do código: funções e arquivos de funções
- Corretude das funcionalidades dos requisitos mínimos.
- Utilização de boas práticas de programação.
- Aparência do sistema (em execução).

# Entregas

---

## Encontro com o professor:

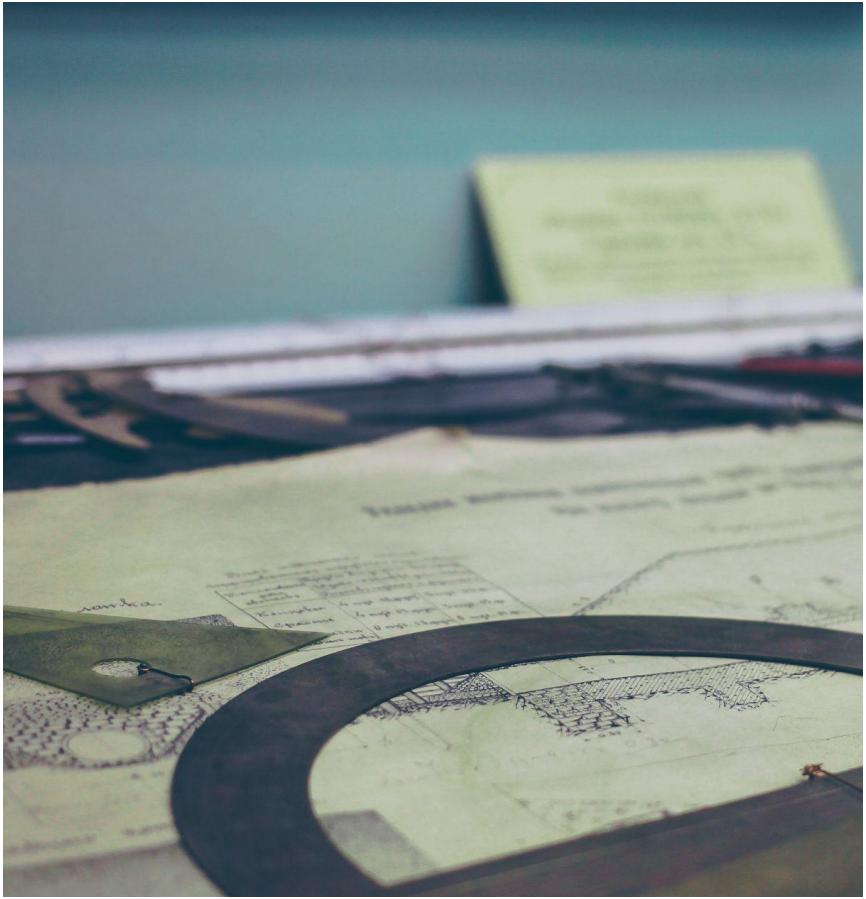
- Entrega 0 (discutir sobre o tema. Traga a idéia para o professor validar)

## - Vão acontecer no GIT.HUB:

- 1a entrega (10% da nota): ter implementado 1 requisito básico.
- 2a entrega (20% da nota): ter implementado 2 requisitos básicos
- 3a entrega (70% da nota) e última entrega

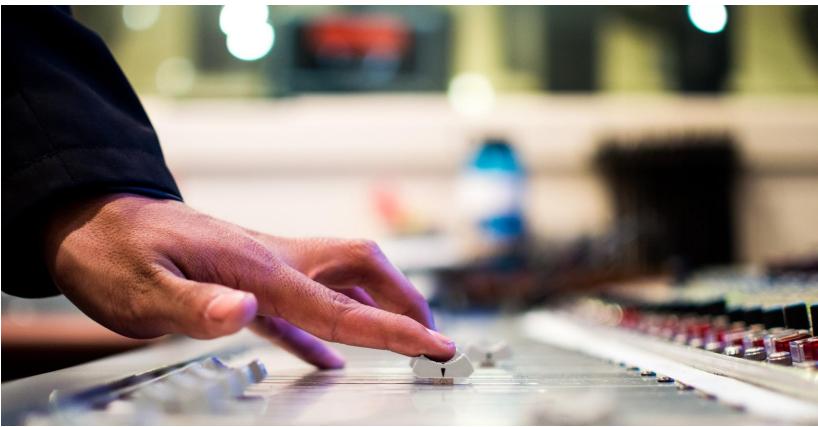
*Consultem as datas no cronograma oficial da disciplina no classroom.*





## Divirta-se

- 1a entrega (10% da nota): ter a idéia do projeto estruturada no papel, ter implementado 1 requisito básico.
- 2a entrega (20% da nota): ter implementado 2 requisitos básicos
- 3a (70% da nota) e ultima entrega  
Consultem as datas no cronograma oficial da disciplina no classroom.



---

## Divirta-se

O projeto vai lhe dar experiência para resolver alguns problemas práticos.

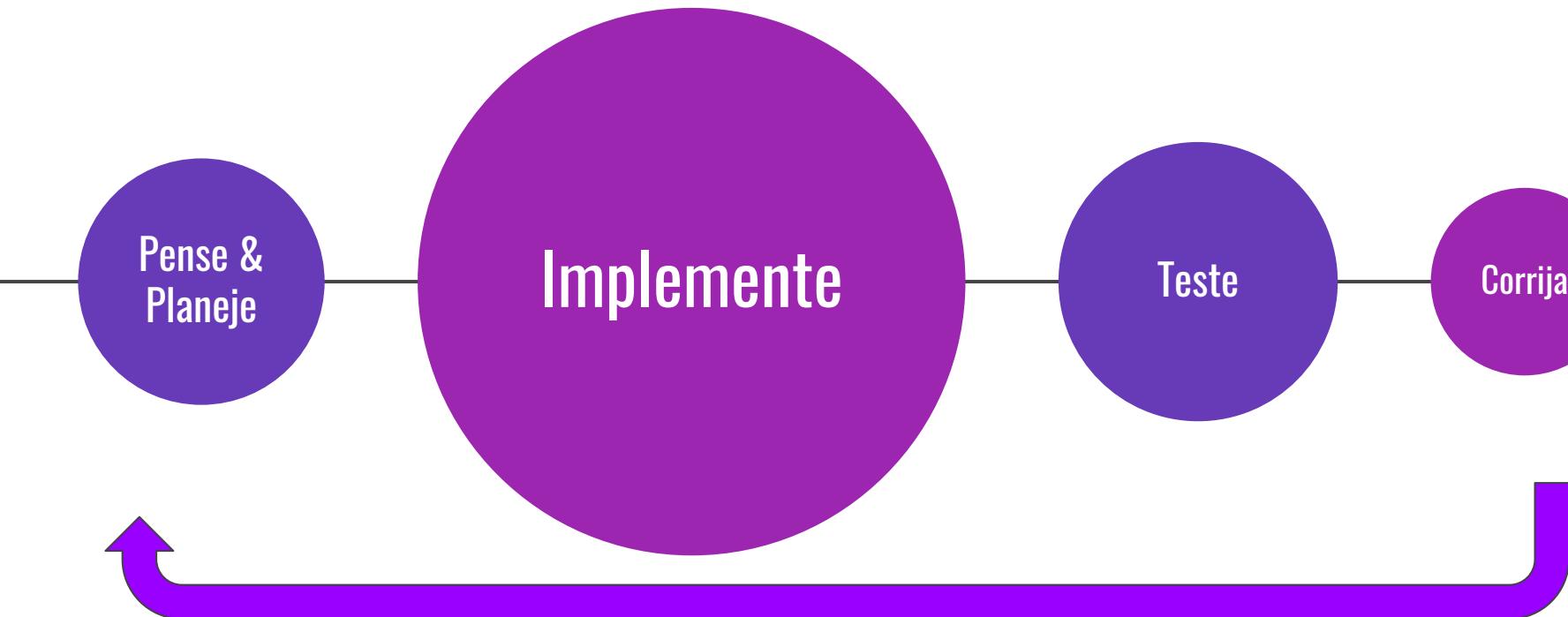


---

## Comece hoje

Se você começar hoje, a probabilidade de ele não atrasar ou ficar pela metade na entrega será pequena.

# Fluxo de execução





# Bom projeto

Ah sim...

>> Projeto individual

>> Fale com o professor/monitores/monitoras

>> Bom coding :)

— — —

Pense que esse projeto  
pode estar no seu  
portfólio

---

---

Experiências acadêmicas  
que contam como  
experiências profissionais.

