

# SI-tecnologias :) Ltda

## Projeto de sistemas de informação

# Contrato

---

A turma de Sistemas de Informação 2017.2 do Centro de Informática foi contratada para prestar serviços a diversas empresas de Recife.

Nosso objetivo é construir aplicações que resolvam problemas corriqueiros (cadastro, busca, armazenamento) de diversas empresas.

**O que sua equipe vai resolver?**

# Contrato

---

Estão disponíveis diversas opções de sistemas. Você precisa escolher uma área de aplicação dentre diversas instituições possíveis:

## **Sistemas para:**

Hospitais, hotéis, ONGs, construtoras, universidades, polícias, etc...

# Sistema de informação

— — —

Você vai desenvolver um sistema que gerencie (armazene e manipulação) informação.

Se você estiver desenvolvendo um programa para um hotel, a informação (elemento) que pode ser armazenada é uma **reserva**. Toda reserva tem nome do contratante, CPF do contratante, data de inicio, data de fim, apartamento alocado como atributos.



# Objetivo

— — —

Neste projeto, você deverá desenvolver um sistema que tem alguns **requisitos mínimos**:

- 1- Deverá armazenar em arquivos as informações que serão detalhadas nas próximas seções deste documento;
- 2- Utilizar dicionários para diminuir o tempo de busca por informações do programa durante sua execução;
- 3- Representar em Tuplas os elementos que tiverem dados heterogêneos;
- 4- As funções do código do sistema devem ser escritas modularizadas e documentadas;
- 3- A equipe que apresentar interface gráfica no programa terá bonificação de +1 na nota.

# Funcionalidades do sistema

---

Você precisará construir um sistema que tenha as seguintes funcionalidades gerais:

- 1- Login/Logout
- 2- Cadastro e remoção de usuário
- 3- Cadastro de elementos
- 4- Remoção de elementos
- 5- Busca de elementos
- 6- Atualização de elementos
- 7- Ordenação de elementos
- 8- Log das ações executadas no sistema
- 9- Alguns arquivos deverão estar criptografados.
- 10 - Imprimir os elementos existentes em um arquivo sem criptografia.

# Detalhamento das funcionalidades

---

## 1- Login/Logout

> o sistema deverá ter a opção de login pelo usuário e senha. O programa deverá ter como padrão de criação um usuário cadastrado no nível de superusuário que tem como login "adm" e senha "adm".

O arquivo que armazena os dados dos usuários deve se chamar "usuarios.txt".

Ele será criptografado usando uma função simples: o arquivo conterá números, que são os valores inteiros ASCII das letras correspondentes mais o valor de 37.

## 2- Cadastro de novos usuários

O programa deve permitir o cadastro de novos usuários também. Ao cadastrar um usuário, o nível de acesso ao sistema deverá ser inicialmente de estagiário.

## 3- Alterar nível de acesso ao sistema

Um usuário do tipo super usuário poderá alterar o nível de acesso de outros usuários do sistema, podendo tornar também outro usuário como super usuário.

(super usuário: acesso completo às funcionalidades do sistema assim como modifica o nível de acesso dos usuários; usuário gerente: acesso completo; usuário tecnico: pode cadastrar elementos e buscar; usuario estagiário: pode buscar elementos.)



# Detalhamento das funcionalidades

---

## 3- Cadastro de elementos

A depender da funcionalidade do sistema, o sistema deve cadastrar os elementos contendo pelo menos 4 atributos que caracterize esse elemento mais um atributo de identificador único (se for uma pessoa, CPF; um livro ISBN; etc).

Os elementos deverão ser salvos em um arquivo chamado "elementos.txt". Seu conteúdo também deve seguir a mesma criptografia simples detalhada acima na sessão login/logout.

## 4- Remoção de elementos

O usuário poderá remover elementos fazendo a busca pelo seu identificador único ou por pelo menos 2 de seus atributos.

Se 2 ou mais elementos possuírem a mesma descrição de algum atributo que usuário esteja fazendo a busca, devem ser mostrados todos esses elementos, para que então o usuário escolha qual deles vai ser removido.

## 5- Busca de elementos

O elemento pode ser buscado por seu atributo identificador ou por pelo menos 2 de seus atributos.

# Detalhamento das funcionalidades

---

## **6- Atualização de elementos**

Os atributos de um elemento podem ser atualizados. Primeiramente, é necessário uma busca (item 5) por um elemento para depois sugerir de alguma forma ao usuário a atualização de seus valores de atributo.

## **7- Ordenação de elementos**

O programa deve imprimir em um arquivo todos os elementos cadastrados na ordem usando o identificador único como parâmetro de ordenação, usando algum algoritmo de ordenação existente implementado pela equipe.

## **8- Log de ações (Armazenamento e busca por data ou por usuário)**

Para todas as operações (descritas nos itens anteriores), será salvo no arquivo "log.txt" o login do usuário, a data de execução e a operação executada.

Para este arquivo não precisa haver criptografia.

O usuário poderá fazer a busca nos logs existentes pela data de execução da ação ou por usuário.

# Detalhamento das funcionalidades

— — —

## 9- Arquivos com criptografia.

A criptografia simples é para o arquivo "usuarios.txt" e "elementos.txt". Os arquivos que armazenam informações do programa

só precisam ser alterados quando o usuário pedir para sair do programa. Ou seja, se durante a execução do sistema, ele inserir 2 elementos, atualizar 3 elementos e remover 7 elementos,

essas alterações só precisam estar no arquivo "elementos.txt" quando o usuário encerrar o programa.

# Detalhamento das funcionalidades

— — —

Exemplos de funcionamento:

Se você precisa armazenar: "fernandoneto" e "senha 123" que identificam o nome de usuario e senha de um dado usuário, o conteúdo a ser armazenado no arquivo de "usuarios.txt" deve ser os números inteiros ASCII correspondentes a cada letra somados a esses números o valor 37.

Para esse exemplo, o arquivo ficaria:

```
139 138 151 147 134 147 137 148 147 138 153 148
```

```
152 138 147 141 134 69 86 87 88
```

Perceba que aqui o conteúdo de usuario 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.

# Detalhamento das funcionalidades

---

## **10 - Imprimir os elementos existentes em um arquivo sem criptografia.**

Seu programa deve conseguir imprimir todos os arquivos cadastrados no sistema em um arquivo com apresentação legível e organizada. O arquivo deve se chamar "impressaoelementos.txt".

## **11 - Sair**

Ao sair, o programa deve atualizar os usuários e elementos que foram cadastrados, atualizados ou removidos durante a execução do programa nos arquivos correspondentes.

# 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 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.

Por exemplo: o dicionário que armazena clientes ficaria assim: {1: ("Fernando", 27, "Recife", "Rua dos Reitores"), 2: ("Vitor", 22, "Recife", "Avenida Boa Viagem")}, considerando que 1 e 2 são os atributos identificadores dos dois clientes "Fernando" e "Victor" respectivamente.

# Observações gerais

---

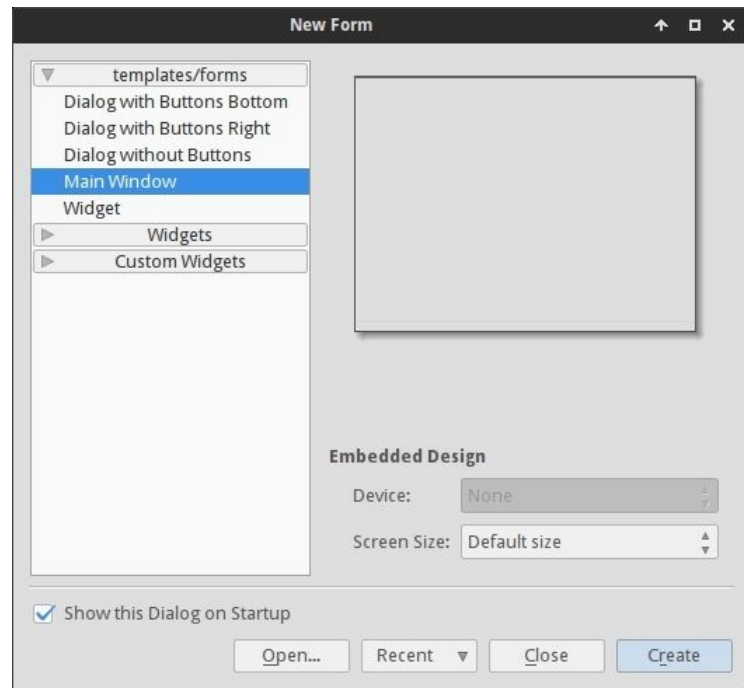
- A interface gráfica é opcional mas terá uma bonificação de +1 na nota do projeto.
- Modularize seu sistema. Escreva uma função para cada funcionalidade existentes. Não repita partes do código que fazem a mesma coisa: crie funções e as chame quando precisar.
- Você não poderá importar nenhuma biblioteca de Python. Você pode usar as funções que o python por padrão oferece ao programador.

# Interface gráfica

— — —

Existem vários sistemas de interface compatíveis com o Python

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





# Critérios gerais de avaliação

— — —

- Modularização do código em funções e em arquivos de funções
- Funcionamento correto das funcionalidades
- Utilização de boas práticas de programação
- Aparência do sistema (em execução)

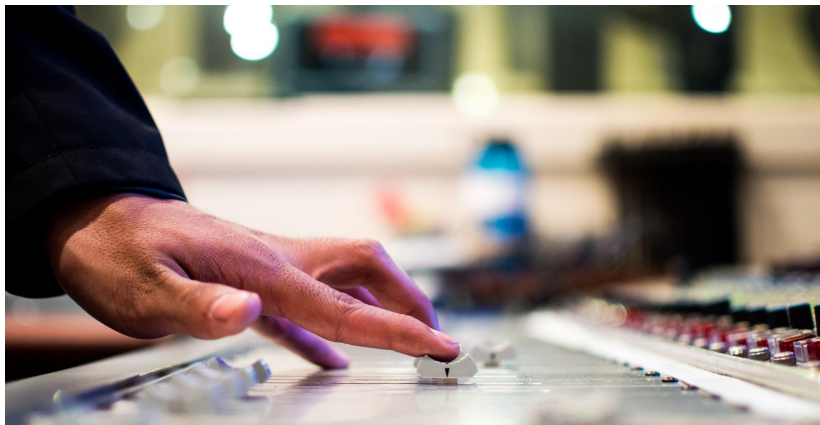


## **Cronograma**

30/10 - Acompanhamento do projeto (leitura e escrita em arquivos)

06/11 - Acompanhamento do projeto (adicionar, remover e buscar elementos)

13/11- Apresentação final do projeto do Projeto



---

## Divirta-se

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

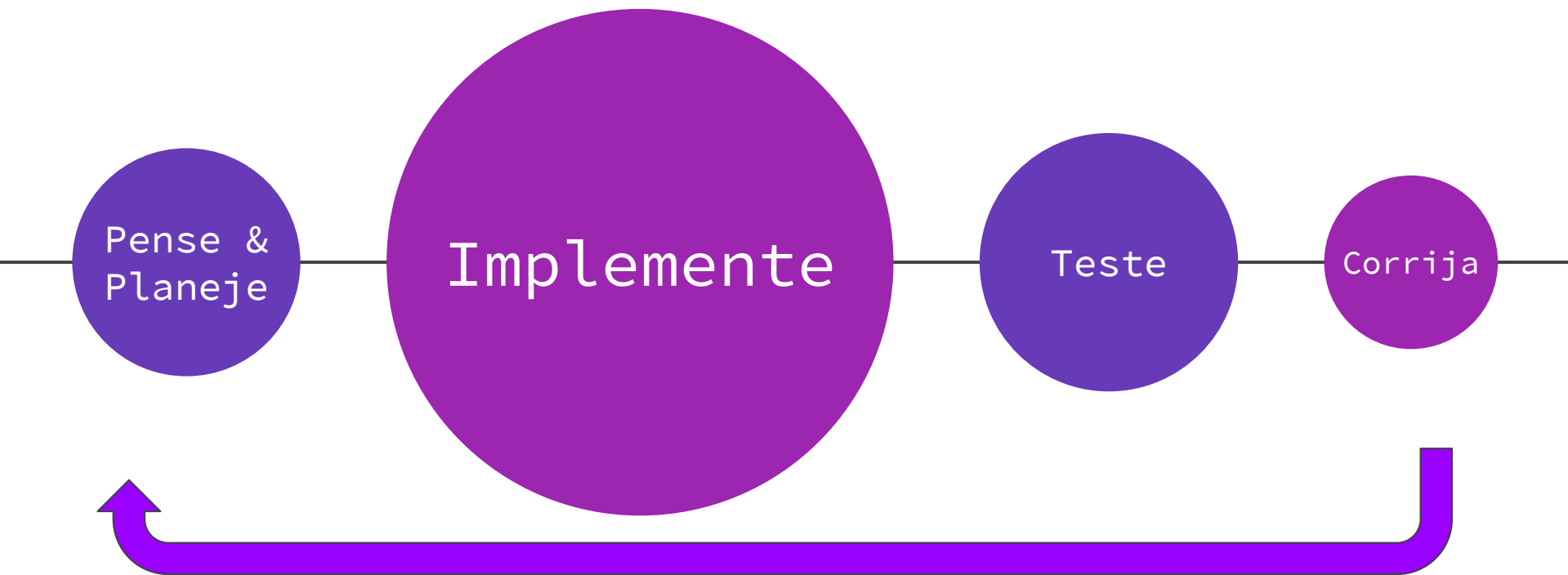


---

## 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

## **Próximos passos**

- >> Escolha sua equipe
- >> Escolha o tema do seu projeto
- >> Fale com o professor
- >> Bom coding :)

— — —

Muita gente precisa  
de tecnologia.

Pense nas pessoas.



Pense que esse projeto  
pode estar no seu  
portfolio

— — —

Experiências acadêmicas  
que contam como  
experiências  
profissionais.

