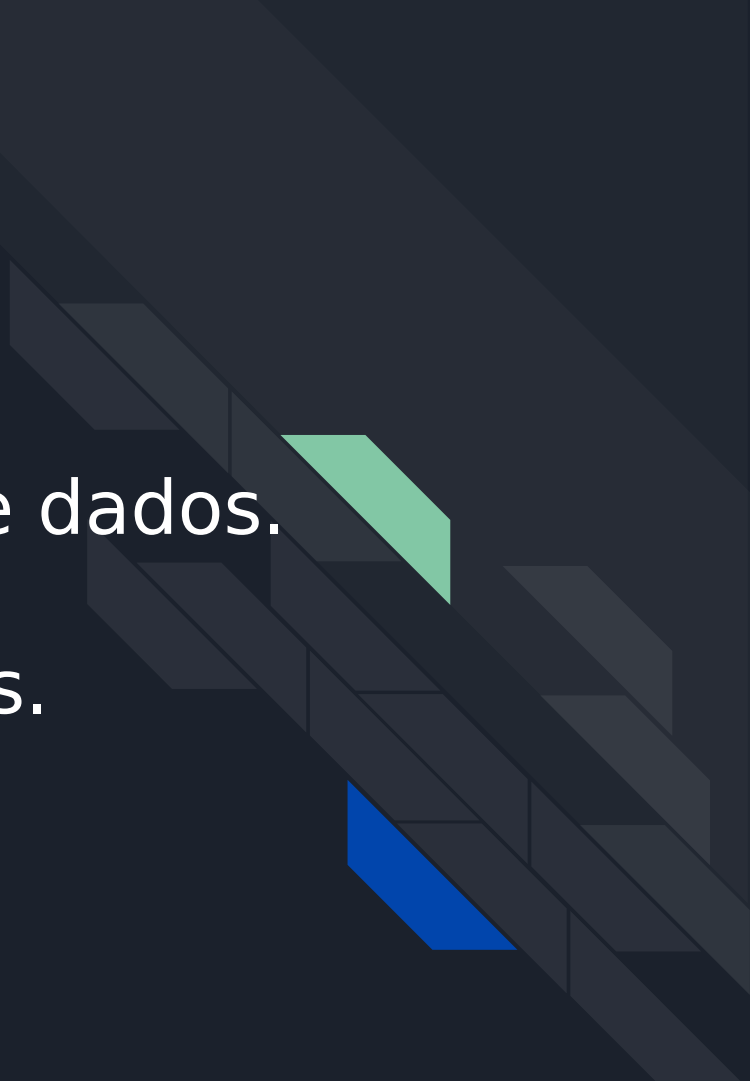


A decorative graphic on the left side of the slide consisting of overlapping geometric shapes. It includes a blue parallelogram, a light green parallelogram, and a dark grey parallelogram, all with thin black outlines.

Help Desk

Aldrijon Martinson

- Classes
 - Métodos
 - Herança
 - Interação com banco de dados.
 - Modularização.
 - Tratamento de Exceções.
 - Interface.
- 

Classes

```
import sqlite3

class Repositorio():
    """Classe Repositorio
    Utilizada para o acesso ao Repositorio (SQLite)

    arq: Base.py
    """
    def __init__(self):|
        #Estabelecendo a conexão ao Repositorio de dados
        self.conectar = sqlite3.connect('Repositorio.db')
        #Criação da relação
        self.criarTabela()
```




Classes

```
from Base import *  
  
class Desk(Repositorio):  
    def __init__(self):  
        self.conectar = sqlite3.connect('Repositorio.db')
```



Classes

```
from tkinter import *  
from tkinter import messagebox  
from Base import *  
from Desk import *  
  
class Main:  
    def init (self, master):
```



Métodos Repositorio

```
def criarTabela(self):  
    #Criando o cursor para acessar o banco de dados  
    cursor = self.conectar.cursor()  
  
    #Criando relação  
    cursor.execute("""create table if not exists chamados (  
        id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT ,  
        nome TEXT NOT NULL,  
        fone TEXT,  
        setor TEXT,  
        maquina TEXT,  
        descricao TEXT)""")  
    )  
    #confirmando informações no banco  
    self.conectar.commit()  
    #fechando cursor  
    cursor.close()
```

Métodos Desk

```
def adicionar(self,nome,fone,setor,maquina,descricao):
    try:
        self.conectar = sqlite3.connect('Repositorio.db')
        cursor = self.conectar.cursor()
        cursor.execute("insert into chamados values(NULL,?,?,?,?,?)",
            self.conectar.commit()
        return "Chamado cadastrado!"
    except:
        return "Não foi possível realizar o cadastro do chamado"

def mostrar(self):
    self.conectar = sqlite3.connect('Repositorio.db')
    cursor = self.conectar.cursor()
    select= cursor.execute("select * from chamados")
    self.conectar.commit()
    return select
```



Métodos Interface

```
def adicionar (self):  
    if self.nome.get() == "" or self.fone.get() == "":  
        messagebox.showwarning("Erro", "Por Favor insira Nome e Telefone")  
    else:  
  
def atualizar (self):  
    # uso de Herança da base de dados para mostrar na tela  
    lista = Desk.mostrar(self)  
    self.listbox.delete(0, END) #Limpar listbox  
    for i in lista:  
        self.listbox.insert(END, i)
```




Herança

```
from Base import *  
  
class Desk(Repositorio):  
    def __init__(self):  
        self.conectar = sqlite3.connect('Repositorio.db')
```