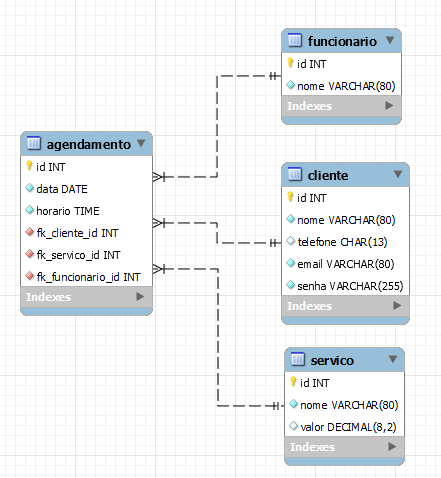
**Prova:** Estilo do Rei App



**Rotas**

/ - ***página principal***

/novofuncionario

/novocliente

/novoservico

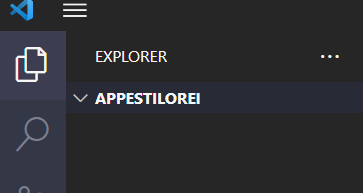
/novoagendamento

/login

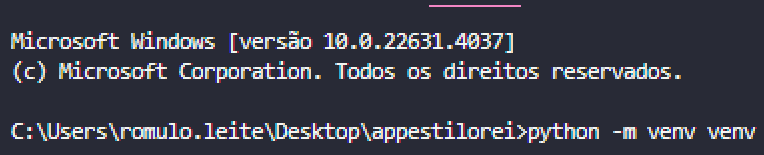
/logout

**Item 1 -** Criar ambiente virtual python (venv)

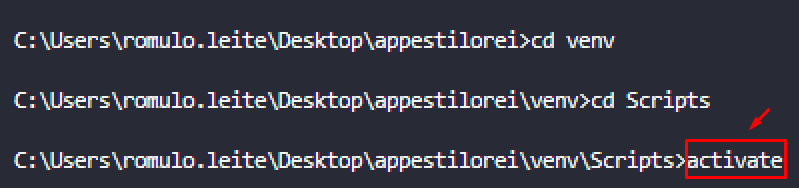
**Habilidades**



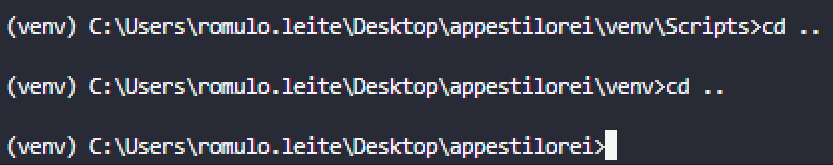
habilidade 1 - criar uma pasta e abrir no Visual Studio Code



habilidade 2 - Criar a venv manualmente (via código)

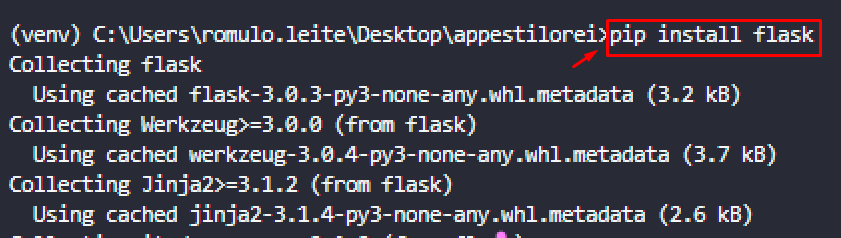


habilidade 3 - Entrar na venv e ativá-la

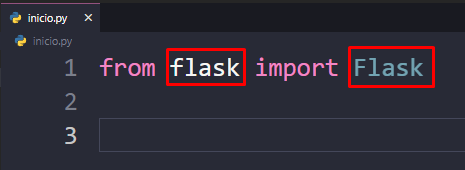


habilidade 4 - sair da venv

Item 2 – Instalar e Importar o Flask e outras bibliotecas para programação web



habilidade 5 - Instalar o Flask



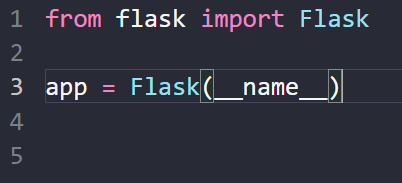
habilidade 6 - Entender a importação do Flask

Entender:

**flask :** módulo (biblioteca)

**Flask**: classe

A partir de agora concentre-se no **Flask**



**app**: objeto da classe **Flask**

**Flask( )** : construtor do objeto

**\_\_name\_\_** : contexto da aplicação

|  |  |  |
| --- | --- | --- |
| **item** | **Avaliar** | **nota** |
| 1 | Criar ambiente virtual python | ótimo |
| 2 | Instalar o Flask | ótimo |
| 3 | Entender a importação do Flask | ótimo |
| 4 | Criar um objeto flask | ótimo |
| 5 | Entender o que é um objeto    **Método** (poo)  **Função** (estruturado e funcional) | ótimo |
| 6 | Entender o que é uma classe | ótimo |
| 7 | Entender o que é um construtor | ótimo |
| 8 | Entender método | ótimo |
| 9 | Diferenciar o que é função de método no ***Python***  **Função** – não tem classe  **Método** – tem classe | ótimo |
| 10 | Entender a estrutura mínima de um projeto Flask    habilidade 7 - Entender a estrutura mínima | ótimo |
| 11 | habilidade 8 - Executar o Aplicativo | ótimo |
| 12 | habilidade 9 - Criar novas rotas  Seguindo o exemplo da habilidade 9 e considerando as novas rotas.  Crie as novas rotas.  @app.route('/novocliente')  def cadastrar\_cliete():      nome = request.form['nome']      telefone = int(request.form['telefone'])      email = request.form['email']      senha = request.form['senha']          cliente = cliente(nome=nome, telefone=telefone,email=email,senha=senha)      try:        session.add(cliente)        session.commit()      except:        session.rollback()        raise      finally:         session.close()      mensagem = f"Cadastro do cliente {nome} efetuado com sucesso!"  @app.route('/novoservico')  def cadastrar\_servico():      servico = request.form['servico']      valor = request.form['valor']      novoservico = novoservico(servico=servico, valor=valor)      try:        session.add(novoservico)        session.commit()      except:        session.rollback()        raise      finally:         session.close()      mensagem2 = f"O {servico} foi cadastrado com sucesso no sistemas!"      return render\_template('cadastro.html', mensagem2=mensagem2)  @app.route('/novoagendamento')  def novo\_agendamento():      data = request.form['data']      hora = request.form['hora']      cliente = request.form['cliente']      servico = request.form['servico']      funcionario = request.form['funcionario']      novoagendamento = novoagendamento(data=data, hora=hora,cliente=cliente,servico=servico,funcionario=funcionario)      try:        session.add(agendamento)        session.commit()      except:        session.rollback()        raise      finally:         session.close()      mensagem3 = f"O {servico} foi agendado para o dia {data}, horário {hora} com o(a) {funcionario}!"      return render\_template('agendamento.html', mensagem3=mensagem3)  @app.route('/login', methods=['POST'])  def login():      if request.method == 'POST':          email = request.form.get('email')          senha = request.form.get('senha')            usuario = usuario.query.filter\_by(email=email).first()          if usuario and usuario.senha == senha:              login\_usuario(usuario)              mensagem4= 'Login feito com sucesso!'              return render\_template('home.html', mensagem4=mensagem4)          else:              mensagem5='Email ou senha errado'              return render\_template('login.html', mensagem5=mensagem5)  @app.route('/logout')  def logout():      session.clear()      return render\_template('index.html') |  |
| 13 | habilidade 10 - identificar as rotas no navegador | ótimo |