**Revisão Desenvolvimento rápido de aplicações em python**

**1-O que é RAD em python?**

**RAD (Rapid Application Development) em Python é uma metodologia de desenvolvimento ágil que prioriza a prototipagem rápida e ciclos iterativos para acelerar a criação de aplicações.**

**Ele permite ajustes contínuos com base no feedback, reduzindo o tempo de desenvolvimento. Ferramentas como Django, Flask e Tkinter são frequentemente usadas para implementar RAD em Python.**

**2-Explique sobre as fases da RAD.**

**O modelo RAD possui cinco fases: definição do fluxo de informações, refinamento dos dados, organização dos processos, criação de protótipos funcionais e testes ágeis.**

**Essa abordagem agiliza o desenvolvimento, permitindo ajustes contínuos e correção rápida de erros a cada iteração. Isso reduz o tempo total de produção e melhora a eficiência do projeto.**

**3-Como funciona o ciclo de desenvolvimento da RAD?**

**O ciclo de desenvolvimento da RAD é baseado em prototipagem rápida e iterações contínuas, permitindo ajustes frequentes conforme o feedback dos usuários.**

**Ele começa com a modelagem do negócio e dos dados, seguida pela estruturação dos processos. Depois, os protótipos são gerados e testados em ciclos curtos, garantindo um desenvolvimento ágil e flexível.**

**4-Cite vantagens e desvantagens da RAD.**

**Vantagens: Entregas rápidas e com qualidade, flexibilidade para mudanças, maior participação dos usuários e testes contínuos que aumentam a estabilidade.**

**Desvantagens: Não é ideal para projetos pequenos ou mal definidos, exige alta colaboração, pode ser caro e nem todos os sistemas suportam RAD.**

**5-O que é a manipulação de dados em arquivos no python?**

**A manipulação de dados em arquivos no Python envolve operações como leitura, escrita e edição de arquivos. Utilizando funções como open(), read(), write() e close(), é possível acessar e modificar arquivos de texto, CSV, JSON e outros formatos.**

**6-Cite todos os caracteres e suas devidas funções na manipulação de arquivos em python.**

**‘r’ abre o arquivo para leitura (default).**

**‘w’ abre o arquivo para escrita, truncando o arquivo primeiro.**

**‘x’ cria um arquivo para escrita e falha, caso ele exista.**

**'a' Abre o arquivo para escrita, acrescentando conteúdo ao final do arquivo, caso ele exista.**

**'b' Modo binário.**

**'t' Modo texto (default).**

**'+' Abre o arquivo para atualização (leitura ou escrita).**

**7-Como funciona a manipulação de arquivos-texto em python?**

**A manipulação de arquivos-texto em Python envolve abrir, ler, escrever e fechar arquivos.**

**O comando `open("arquivo.txt", "r")` abre um arquivo para leitura, `arquivo.write("Texto")` permite escrever, e `arquivo.close()` fecha o arquivo. O bloco `with open("arquivo.txt", "r") as arquivo:` garante o fechamento automático.**

**8-O que é a manipulação de strings em python?**

**A manipulação de strings em Python permite modificar, analisar e transformar textos. Exemplos incluem:**

**Concatenação: `"Olá" + " Mundo"` resulta em `"Olá Mundo"`**

**Fatiamento: `"Python"[0:3]` retorna `"Pyt"`**

**-Maiúsculas e minúsculas: `"python".upper()` retorna `"PYTHON"` e `"PYTHON".lower()` retorna `"python"`**

**Substituição: `"banana".replace("a", "o")` gera `"bonono"`**

**Remoção de espaços: `" Olá ".strip()` vira `"Olá"`**

**Divisão: `"maçã,banana,uva".split(",")` retorna `["maçã", "banana", "uva"]`**

**Junção: `"-".join(["maçã", "banana", "uva"])` resulta em `"maçã-banana-uva"`**

**9-Como funciona o tratamento de exceções em python?**

**O tratamento de exceções em Python permite lidar com erros de forma controlada, evitando que o programa quebre inesperadamente. Isso é feito com a estrutura `try-except`. Aqui está como funciona:**

**Bloco try: Contém o código que pode gerar uma exceção.**

**Bloco except: Captura e trata a exceção, evitando que o programa pare abruptamente.**

**Bloco else: Executa código se \*\*nenhuma\*\* exceção ocorrer.**

**Bloco finally: Executa código \*\*sempre\*\*, independentemente da ocorrência de exceções.**

**10-Como funciona a manipulação de diretório em python?**

**A manipulação de diretórios em Python permite criar, excluir, navegar e modificar pastas no sistema de arquivos. Para isso, são utilizadas principalmente as bibliotecas os e pathlib.**

**os: Biblioteca tradicional do Python para interagir com o sistema operacional, permitindo criar, excluir, listar e navegar em diretórios de maneira procedural.**

**pathlib: Introduz uma abordagem orientada a objetos para manipular caminhos e diretórios, tornando o código mais intuitivo e legível.**

**11-Explique sobre o dataclass.**

**Em Python, dataclass é um recurso do módulo dataclasses introduzido na versão 3.7 que facilita a criação de classes para armazenar dados.**

**Ele reduz a necessidade de escrever métodos comuns manualmente, como \_\_init\_\_, \_\_repr\_\_ e \_\_eq\_\_.**

**12-Como funciona um banco de dados no python?**

**Em Python, um banco de dados funciona como um sistema de armazenamento e recuperação de dados estruturados.**

**Python oferece várias opções, como SQLite, MySQL, PostgreSQL e MongoDB. Ex: CREATE TABLE define uma tabela, INSERT INTO adiciona registros, SELECT \* FROM recupera dados, UPDATE modifica valores e DELETE FROM remove informações.**

**13-O que é o modelo MVC?**

**O modelo MVC (Model-View-Controller) é um padrão de arquitetura de software que separa a aplicação em três componentes, facilitando a organização e manutenção do código.**

**Model (Modelo): Gerencia os dados e a lógica. Ex: um sistema de vendas pode ter modelos para produtos e pedidos.**

**View (Visão): Exibe os dados de forma visual. Ex: uma página da web mostrando uma lista de produtos.**

**Controller (Controlador): Processa ações do usuário e organiza a atualização dos outros componentes. Ex: ao adicionar um produto ao carrinho, ele registra a ação.**

**14-Diferencie o SQLite e SQLAlchemy.**

**O SQLite é um banco de dados leve e embutido, ideal para aplicações menores que não precisam de um servidor separado. Ele funciona diretamente com arquivos locais, sendo rápido e fácil de usar.**

**O SQLAlchemy é uma biblioteca que facilita o uso de diferentes bancos de dados, incluindo SQLite, MySQL e PostgreSQL. Além de permitir consultas mais flexíveis, ele também suporta ORM, tornando o gerenciamento de dados mais estruturado.**

**15-Como funciona o modulo OS em python?**

**O módulo os em Python fornece funções para interagir com o sistema operacional. Ele permite acessar informações do sistema, manipular diretórios e arquivos, executar comandos e gerenciar processos. Alguns exemplos comuns incluem:**

**os.listdir() – lista arquivos e diretórios.**

**os.mkdir() – cria um novo diretório.**

**os.remove() – exclui um arquivo.**

**os.system("comando") – executa comandos no terminal.**

**16-O que é um GUI em python?**

**Uma GUI (Graphical User Interface) em Python é uma interface gráfica que permite a interação visual dos usuários com um programa, utilizando janelas, botões, menus e outros elementos gráficos, tornando a experiência mais intuitiva.**

**Para desenvolver GUIs, Python oferece bibliotecas como Tkinter, que é simples e integrada ao Python, PyQt, baseada no framework Qt e muito flexível, Kivy, ideal para dispositivos móveis com suporte a gestos e multitouch, e PyGTK, usada em ambientes Linux para interfaces GTK. Essas ferramentas facilitam a criação de aplicações visuais interativas.**

**17-O que é o tkinter e suas características?**

**Tkinter é a biblioteca padrão do Python para a criação de interfaces gráficas. Ele fornece um conjunto de ferramentas para desenvolver aplicações visuais com janelas, botões, menus, caixas de texto e outros elementos interativos. Como está incluído na instalação padrão do Python, não é necessário instalar pacotes extras para usá-lo.**

**Entre suas características, destaca-se a facilidade de uso, sendo ideal para aplicações simples. Ele possui integração com Tcl/Tk, tornando-se eficiente e multiplataforma. Seu design é básico, mas pode ser personalizado com estilos e configurações.**

**18-Qual comando para confirmar uma tarefa no sqlite?**

**Commit()**

**19-Qual comando para ler um arquivo TXT linha por linha?**

**Arquivo.readlines()**

**20-Dentro do nosso dia a dia utilizamos TRY para utilizarmos tratamento para possíveis erros, quando a parte ELSE de TRY-EXCEPT-ELSE será executada?**

**Quando nenhuma EXCEPT ocorre.**

**21-Qual widget Tkinter é usado para criar uma caixa de entrada de texto onde o usuário pode digitar informações?**

**Entry()**

**22-Qual widget Tkinter é usado para exibir texto em uma interface gráfica?**

**Label.**