API (Application Programming Interface)

Integração de sistemas(união de diferentes sistemas para compartilhar informações) de maneira simplificada, sem a necessidade de telas --- respostas

\* Responde chamadas de sistemas como recursos

\* recursos

\* Publica/Privada(requer autenticação)/Terceiros(um produto para ser integrado)

os dados trafegados em api são xml ou json

as repostas e requições são feitas através do protocolo HTTP

verbos do http: get(resgatar), post(enviar), delete(remover registro), put(atualizar), patch(atualizar parcial)

endpoint é uma URL que acessamos para extrair uma resposta da API --- dependendo totalmente do verbo

REST É UM PADRÃO DE ARQUITETURA PARA CRIAÇÃO DE API E PROTOCOLO HTTP

composto por 6 partes

Uniformidade, Desacoplação, Stateless, Cache, Arquitetura de camadas, Code on demand

HTTP é um protocolo de comunicação que permite a transferência de informações na internet. BASE ENTRE BROWSER E SERVIDOR WEB

http usa o tcp com protocolo de transporte para garantir a entrega confiável

* 1xx: Informações
* 2xx: Sucesso
* 3xx: Redirecionamento
* 4xx: Erro do cliente
* 5xx: Erro do servidor

os padrões de projeto foram criados para solução de problemas de software, anteriormente foi feito uma análise em diversos software e identificaram problemas semelhantes/iguais, sendo assim para evitar essa falha ou mal funcionamento, criaram soluções semelhantes para esses problemas

sendo esses padrões de projeto divididos em três categorias:

padrão de criação: são padrões utilizados para criação de objetos/entidades

padrão de estrutura: lidam com a ORGANIZAÇÃO de objetos tentando simplificar a comunicação entre objetos e o acoplamento

padrão de comportamento: lidam com a comunicação , buscando definir comportamento

existem diversas apis no mundo java para ser utilizadas para facilitar a construção de uma aplicação em java, customizando o tempo para aplicar em logica de negocio do que problemas técnicos

uma das bibliotecas que convertem um json (estrtura de dados de chave:valor) em objeto tem no MVN repositor como GSON

para implementação de dependências(bibliotecas; API’s) utiliza-se os sistema de gerenciamento de pacotes

serialização é o processo de converter um objeto em formato que pode ser guardado ou enviado

try catch ->utilizado para tratar exceções

finally -> é usado para executar uma ação independente de ocorrer uma exceção ou não, ele é sempre executado

todas as exceções em java estão em uma hierarquia de classeDiagrama

O conteúdo gerado por IA pode estar incorreto.

 IOException, que indica algum problema relacionado com leitura/escrita de dados.

As exceções que herdam da classe **Error** representam erros irrecuperáveis pelo sistema, como falta de memória ou falhas internas

Além disso, existe ainda a classe de exceção **RuntimeException**, que é uma subclasse direta de Exception, e as classes que herdam dela são chamadas de exceções não verificadas (unchecked exception). As exceções não verificadas indicam erros lógicos no código, como a NullPointerException, que indica o acesso a algum atributo ou método de um objeto que é nulo, ou seja, que não foi instanciado ou foi atributo ao valor null.

O código vai gerar um erro de compilação, pois a exceção mais genérica, no caso a Exception, deve ser declarada no último bloco catch.

Pacote java.io

As classes **FileReader** e **FileWriter** são usadas para ler e escrever dados em arquivos de texto, sendo que a classe FileReader lê os caracteres de um arquivo de texto, enquanto a classe FileWriter escreve os caracteres.

Iterando sobre o arq com char

Texto

O conteúdo gerado por IA pode estar incorreto.

Texto

O conteúdo gerado por IA pode estar incorreto.

* **BufferedReader** e **BufferedWriter**: são usadas para ler e gravar arquivos de texto de maneira eficiente, lendo e escrevendo uma linha por vez. Elas usam um buffer para armazenar os dados, o que torna a leitura e escrita mais rápida do que quando feita um caractere por vez;
* **FileInputStream** e **FileOutputStream**: são usadas para ler e gravar dados binários em um arquivo. Eles são usados para ler e gravar dados em arquivos que não são de texto, como imagens e arquivos de áudio;
* **ObjectInputStream** e **ObjectOutputStream**: são usadas para ler e gravar objetos em arquivos. Isso permite que você armazene objetos Java em arquivos para uso posterior ou para transferência entre diferentes aplicações.
* Aprendemos a salvar um arquivo, via classe FileWriter, mas caso você queira fazer o caminho contrário, ou seja, ler o conteúdo de um arquivo existente em Java, pode fazer isso com a utilização da classe **Scanner**.
* A classe Scanner é uma classe padrão do Java que permite ler dados de diferentes fontes, incluindo arquivos. Para ler um arquivo com essa classe, basta criar uma instância dela e passar como argumento um objeto do tipo File, contendo o caminho do arquivo.

Texto

O conteúdo gerado por IA pode estar incorreto.

O Gson é uma biblioteca do Google para Java que facilita a conversão entre objetos Java e JSON. Ele é amplamente utilizado para serializar objetos Java em JSON (transformá-los em uma estrutura legível) e desserializar JSON de volta em objetos Java. Isso é útil em aplicações que se comunicam via APIs, armazenam dados em JSON ou trabalham com configurações dinâmicas. Além disso, o Gson lida bem com coleções, tipos genéricos e até mesmo objetos complexos.

O GsonBuilder é uma classe do Gson usada para configurar e personalizar a instância de Gson antes de criá-la. Enquanto o Gson por padrão já oferece funcionalidades essenciais para conversão entre JSON e objetos Java, o GsonBuilder permite ajustes mais refinados, como:

* **Definir formatação do JSON** (setPrettyPrinting()) para tornar a saída mais legível.
* **Ignorar campos nulos** (serializeNulls(false)) para não incluir valores null no JSON gerado.
* **Configurar estratégias de exclusão** (excludeFieldsWithModifiers()) para omitir certos atributos.
* **Criar adaptadores personalizados** (registerTypeAdapter()) para manipular tipos específicos.

A FORMA DE COMO VOCE INTERAGE COM ALGUMA COISA DENTRO DO SEU SISTEMA