

📱 Desafio Java:

Projeto **ArattaiGrowthTracker** - Análise de Dados Virais ## 📌 Regras da Avaliação

(Estrutura e Conduta) Este é um exame prático individual com regras estritas de entrega e

conduta. | Conduta | Status | Detalhes | | :--- | :--- | :--- | | **Consulta a Materiais/IA** | ❌

PROIBIDA | Proibido usar qualquer material externo, internet ou ferramentas de IA. | |

Comunicação com Colegas | ❌ **PROIBIDA** | Qualquer forma de comunicação com

colegas. | | **Entrega do Código** (`git push`) | ❌ **ANULADA** | **Se feito fora do horário de**

aula. | | **Local do Commit** | ❌ **ANULADA** | **Se o commit for detectado fora da**

rede/localização da Universidade. | --- ## 🕒 Regras de **Commit** e Presença (ESTRITO)

Atenção: Estas regras são auditáveis e o não cumprimento anulará a sua avaliação. 1.

Horário de Aula: A prova deve ser realizada e finalizada **estritamente** no horário de aula.

2. **Registro de Presença:** Para validar sua participação, você deve assinar a lista de

presença física ao iniciar a prova. ### Validação por **Commit** (Obrigatório) * **Validação de**

Tempo: O **último commit** no seu repositório Git, contendo o código final, deve ter o

timestamp **dentro do horário de aula**. * **Exemplo:** Se o horário final é 10h00, um **commit**

às 10h01 anula a prova. --- **Esta é uma prova individual e sem consulta.** O foco é na sua

capacidade de aplicar conceitos de arrays e loops. ## Leia a notícia a seguir O aplicativo

indiano que quer desafiar a supremacia do WhatsApp Nas últimas semanas, o Arattai,

aplicativo desenvolvido pela empresa de tecnologia indiana Zoho, tornou-se uma sensação viral

no país. A companhia afirma ter registrado sete milhões de downloads em "sete dias da última

semana", sem especificar as datas. De acordo com a empresa de inteligência de mercado

Sensor Tower, os downloads do aplicativo não chegaram a 10 mil em agosto. Arattai, que

significa bate-papo no idioma tâmil, teve um lançamento discreto em 2021. O repentino

aumento em sua popularidade está sendo associado ao incentivo do governo federal à

autossuficiência, em um momento em que a Índia lida com os impactos das altas tarifas

comerciais impostas pelos Estados Unidos em seus produtos. Fonte:

<https://www.bbc.com/portuguese/articles/ced5qy7ggzgo>

Acesso em 2025/10/18 ## 💡 Contexto do Problema O aplicativo de mensagens indiano

Arattai (que significa "bate-papo" em tâmil) experimentou um aumento repentino de

popularidade, embora os dados oficiais de **downloads** sejam questionáveis. Para entender

melhor a sua trajetória, você foi contratado(a) como analista de dados júnior para processar e

analisar a taxa de crescimento diário do aplicativo. Seu desafio é utilizar **Arrays (Vetores)** e

Estruturas de Repetição (`for` ou `while`) em Java para manipular e sumarizar as

estatísticas de **downloads** ao longo de uma semana. ## 🎯 Objetivo da Prova Demonstrar o

domínio na criação, manipulação e iteração sobre **Arrays** em Java, utilizando laços de

repetição para realizar cálculos e análises estatísticas simples. ## 📋 Requisitos Técnicos Seu

código deve estar em um único arquivo Java (`ArattaiGrowthTracker.java`) e deve cumprir as

seguintes etapas, utilizando o método `main` como orquestrador. ### Requisito 1: Inicialização

e Armazenamento de Dados (Arrays) 1. Crie um **Array** do tipo **int** chamado

`downloadsDiarios` para armazenar o número de **downloads** registrados em 7 dias (uma

semana). 2. Preencha este **array** com os seguintes valores (simulados) na ordem: `[850000,`

`1050000, 1100000, 950000, 1200000, 1000000, 850000]`. 3. Crie um **Array** do tipo

String chamado `diasDaSemana` e preencha-o com os nomes dos dias: `["Dom", "Seg",`

`"Ter", "Qua", "Qui", "Sex", "Sáb"]`. ### Requisito 2: Cálculo do Total de Downloads (Loop `for`

ou `while`) 1. Utilize uma **estrutura de repetição** (preferencialmente um `for` ou `for-each`)

para percorrer o **array** `downloadsDiarios`. 2. Calcule a **soma total** de **downloads**

registrados na semana. 3. Imprima o resultado formatado (ex: "Total de Downloads na Semana:

7.000.000"). ### Requisito 3: Identificação do Pico e Mínimo (Loop e Estrutura de Decisão) 1.

Utilize uma **estrutura de repetição** para percorrer o **array** `downloadsDiarios`. 2. Encontre

e armazene o **maior** número de **downloads** (Pico) e o **menor** número de **downloads**

(Mínimo). 3. Encontre também os respectivos **dias da semana** em que esses valores

ocorreram, utilizando o **array** `diasDaSemana` em conjunto com o índice. 4. Imprima os

resultados claramente: `"Dia de Pico de Downloads: [Dia da Semana] com [Valor] downloads."`

`"Dia de Mínimo de Downloads: [Dia da Semana] com [Valor] downloads."` ### Requisito 4:

Análise de Metas (Loop e Condicional) 1. Defina uma **Meta Diária** de **downloads** igual a

`*1.000.000*`. 2. Utilize uma **estrutura de repetição** (preferencialmente um `while` ou `for`)

para verificar cada dia da semana. 3. Crie uma variável `diasAcimaDaMeta` para contar quantos dias superaram a meta. 4. Ao final, imprima: "O Arattai superou a meta diária em [X] dias esta semana." ## Massa de Teste Para te apoiar no teste do algoritmo, um analista de sistemas levantou os requisitos para voce: ~~~ === Análise de Crescimento do Arattai (7 dias) === [R2] Total de Downloads na Semana: 7.000.000 [R3] Análise de Extremos: Dia de Pico de Downloads: Qui com 1.200.000 downloads. Dia de Mínimo de Downloads: Dom com 850.000 downloads. [R4] Análise de Metas (Meta Diária: 1000000): > Dom: Não atingiu. (850.000 downloads) > Seg: SUCESSO! (1.050.000 downloads) > Ter: SUCESSO! (1.100.000 downloads) > Qua: Não atingiu. (950.000 downloads) > Qui: SUCESSO! (1.200.000 downloads) > Sex: Não atingiu. (1.000.000 downloads) > Sáb: Não atingiu. (850.000 downloads) [Relatório Final] O Arattai superou a meta diária em 3 dias esta semana.

===== ~~~ ## 🌟 Critérios de Avaliação 1. ****Declaração e Inicialização:**** Uso correto da sintaxe de ***Arrays*** (`int[]`, `String[]`). 2. ****Iteração:**** Uso eficiente e correto de ****loops**** (`for` ou `while`) para percorrer os ***arrays***. 3. ****Lógica:**** Implementação correta das lógicas de soma, identificação de máximo/mínimo e contagem condicional. 4. ****Clareza e Legibilidade:**** O código deve ser organizado e a saída para o console deve ser clara para o usuário. ## 📝 Entrega 1. Crie um novo repositório chamado: una-psc-prova-a1-matricula-par-202502. (2 pts) 2. Adicione um arquivo .gitignore, um arquivo licença e suba este arquivo readme. (3 pts) 3. Crie um arquivo chamado `ArattaiGrowthTracker.java`. (5 pts) 4. Insira todo o seu código Java neste arquivo. (5 pts) 5. Inclua seu nome completo e número de matrícula como comentário no cabeçalho do arquivo. (1 pts) 6. Submeta o arquivo `ArattaiGrowthTracker.java` no seu repositório. (5 pts) 7. Teste seu algoritmo e envie uma imagem dele em funcionamento. (9 pts) Qualquer descumprimento de regras ou de alguma solicitacao da prova deve acarretar em perda total ou parcial dos pontos. Entregue o link do seu repositorio na plataforma. ****Lembre-se: A clareza do seu código e a facilidade de compreensão da saída são cruciais. Confie no seu conhecimento!**** ****Bom trabalho e sucesso!**** ***