

Cenários de Teste (TP de SO)

GRUPO: ARTHUR MENDES, BERNARDO ALVAREZ

Como usar:

1. Inicie o sistema de arquivos como root ou um usuário específico, por exemplo:
`java -cp . filesystem.Main root`
2. Utilize as opções numéricas do menu (1 a 9) para chamar os comandos, fornecendo os caminhos e parâmetros solicitados.
3. Observe a saída no console para verificar se o comportamento esperado (sucesso, erro com exceção) foi atingido.

1. mkdir (Criar Diretório)

Objetivo: Testar a criação de diretórios, incluindo caminhos intermediários e validações.

- **Cenário 1: Criar diretório simples**
 - **Usuário:** root
 - **Comando:** `mkdir /pasta1`
 - **Saída Esperada:** Diretório 'pasta1' criado em '/'.
- **Cenário 2: Criar diretórios intermediários (comportamento -p padrão)**
 - **Usuário:** root
 - **Comando:** `mkdir /projetos/equipeA/docs`
 - **Saída Esperada:** Diretório 'projetos' criado em '/'. Diretório 'equipeA' criado em '/projetos'. Diretório 'docs' criado em '/projetos/equipeA'.
- **Cenário 3: Tentar criar diretório já existente**
 - **Usuário:** root
 - **Pré-condição:** /pasta1 já existe.
 - **Comando:** `mkdir /pasta1`
 - **Saída Esperada:** Erro: Não é possível criar diretório '/pasta1': já existe um diretório com este nome. (CaminhoJaExistenteException)
- **Cenário 4: Tentar criar diretório onde já existe um arquivo**
 - **Usuário:** root
 - **Pré-condição:** Criar /arquivo.txt (`touch /arquivo.txt`)
 - **Comando:** `mkdir /arquivo.txt`
 - **Saída Esperada:** Erro: Não é possível criar diretório 'arquivo.txt': já existe um arquivo com este nome. (CaminhoJaExistenteException)
- **Cenário 5: Tentar criar diretório sem permissão de escrita no pai**
 - **Usuário:** root

- **Pré-condição:** Criar /privado (mkdir /privado). Alterar permissão: chmod /privado root other --- (remover escrita para 'other').
- **Trocar Usuário:** su joao
- **Comando:** mkdir /privado/secreto
- **Saída Esperada:** Erro: Permissão negada: não pode criar diretório em '/privado' para o usuário 'joao'. (PermissionException)

2. touch (Criar Arquivo Vazio ou Atualizar Timestamp)

Objetivo: Testar a criação de arquivos e a atualização de metadados.

- **Cenário 1: Criar arquivo vazio**
 - **Usuário:** root
 - **Comando:** touch /documento.pdf
 - **Saída Esperada:** Arquivo 'documento.pdf' criado em '/'.
- **Cenário 2: "Tocar" em arquivo existente (atualizar timestamp)**
 - **Usuário:** root
 - **Pré-condição:** /documento.pdf existe.
 - **Comando:** touch /documento.pdf
 - **Saída Esperada:** Arquivo 'documento.pdf' já existe, tempo de modificação atualizado.
- **Cenário 3: Tentar criar arquivo onde já existe um diretório**
 - **Usuário:** root
 - **Pré-condição:** /config já existe (mkdir /config).
 - **Comando:** touch /config
 - **Saída Esperada:** Erro: Não é possível criar arquivo 'config': já existe um diretório com este nome. (CaminhoJaExistenteException)
- **Cenário 4: Tentar criar arquivo em diretório pai inexistente**
 - **Usuário:** root
 - **Comando:** touch /naoexiste/arquivo.txt
 - **Saída Esperada:** Erro: Caminho pai '/naoexiste' não existe ou não é um diretório. (CaminhoNaoEncontradoException)
- **Cenário 5: Tentar criar arquivo sem permissão de escrita no pai**
 - **Usuário:** root
 - **Pré-condição:** Criar /publico (mkdir /publico). Alterar permissão: chmod /publico root other r-x (remover escrita para 'other').
 - **Trocar Usuário:** su maria
 - **Comando:** touch /publico/relatorio.txt
 - **Saída Esperada:** Erro: Permissão negada: não pode criar arquivo em '/publico' para o usuário 'maria'. (PermissionException)

3. write (Escrever em Arquivo)

Objetivo: Testar a escrita de dados em arquivos, incluindo sobrescrita e anexação.

- **Cenário 1: Escrever em arquivo vazio (sobrescrever)**
 - **Usuário:** root
 - **Pré-condição:** /log.txt existe (touch /log.txt).
 - **Comando:** write /log.txt false "Primeira linha do log"
 - **Saída Esperada:** <número> bytes escritos em '/log.txt'. (Verifique com read).
- **Cenário 2: Anexar dados em arquivo**
 - **Usuário:** root
 - **Pré-condição:** /log.txt contém "Primeira linha do log".
 - **Comando:** write /log.txt true "Segunda linha do log"
 - **Saída Esperada:** <número> bytes escritos em '/log.txt'. (Verifique com read se o conteúdo é "Primeira linha do logSegunda linha do log").
- **Cenário 3: Escrever em arquivo existente (sobrescrever)**
 - **Usuário:** root
 - **Pré-condição:** /log.txt contém "Primeira linha do logSegunda linha do log".
 - **Comando:** write /log.txt false "Conteudo totalmente novo"
 - **Saída Esperada:** <número> bytes escritos em '/log.txt'. (Verifique com read se o conteúdo é apenas "Conteudo totalmente novo").
- **Cenário 4: Tentar escrever em diretório (erro de tipo)**
 - **Usuário:** root
 - **Pré-condição:** /home existe.
 - **Comando:** write /home false "dados"
 - **Saída Esperada:** Erro: '/home' não é um arquivo ou não existe para escrita. (CaminhoNaoEncontradoException)
- **Cenário 5: Tentar escrever em arquivo inexistente**
 - **Usuário:** root
 - **Comando:** write /naoexiste.txt false "dados"
 - **Saída Esperada:** Erro: '/naoexiste.txt' não é um arquivo ou não existe para escrita. (CaminhoNaoEncontradoException)
- **Cenário 6: Tentar escrever sem permissão de escrita no arquivo**
 - **Usuário:** root
 - **Pré-condição:** Criar /somente_leitura.txt (touch /somente_leitura.txt). Alterar permissão: chmod /somente_leitura.txt root other r--.
 - **Trocar Usuário:** su tiago
 - **Comando:** write /somente_leitura.txt false "tentativa"
 - **Saída Esperada:** Erro: Permissão negada: não pode escrever em

'/somente_leitura.txt' para o usuário 'tiago'. (PermissaoException)

4. read (Ler Arquivo)

Objetivo: Testar a leitura de dados de arquivos.

- **Cenário 1: Ler arquivo existente**
 - **Usuário:** root
 - **Pré-condição:** Escrever conteúdo em /meu_arquivo.txt (write /meu_arquivo.txt false "Conteudo de teste para leitura.").
 - **Comando:** read /meu_arquivo.txt
 - **Saída Esperada:** Conteúdo lido: "Conteudo de teste para leitura."
- **Cenário 2: Tentar ler arquivo inexistente**
 - **Usuário:** root
 - **Comando:** read /naoexiste_arquivo.txt
 - **Saída Esperada:** Erro: '/naoexiste_arquivo.txt' não é um arquivo ou não existe para leitura. (CaminhoNaoEncontradoException)
- **Cenário 3: Tentar ler diretório (erro de tipo)**
 - **Usuário:** root
 - **Pré-condição:** /dados existe (mkdir /dados).
 - **Comando:** read /dados
 - **Saída Esperada:** Erro: '/dados' não é um arquivo ou não existe para leitura. (CaminhoNaoEncontradoException)
- **Cenário 4: Tentar ler sem permissão de leitura no arquivo**
 - **Usuário:** root
 - **Pré-condição:** Criar /confidencial.txt (touch /confidencial.txt). Alterar permissão: chmod /confidencial.txt root other ---. Escrever conteúdo: write /confidencial.txt false "Dados secretos"
 - **Trocar Usuário:** su carla
 - **Comando:** read /confidencial.txt
 - **Saída Esperada:** Erro: Permissão negada: não pode ler '/confidencial.txt' para o usuário 'carla'. (PermissaoException)

5. ls (Listar Conteúdo de Diretório)

Objetivo: Testar a listagem de arquivos e subdiretórios, recursivamente e com validação de permissão.

- **Cenário 1: Listar conteúdo da raiz**
 - **Usuário:** root
 - **Comando:** ls /
 - **Saída Esperada:** Lista de filesystems, exception, lib, tests, users, etc.

- **Cenário 2: Listar diretório específico (não recursivo)**
 - **Usuário:** root
 - **Pré-condição:** Criar /tmp/tempfile.txt e /tmp/subdir
 - **Comando:** ls /tmp
 - **Saída Esperada:** Lista apenas tempfile.txt e subdir dentro de /tmp.
- **Cenário 3: Listar diretório recursivamente**
 - **Usuário:** root
 - **Pré-condição:** Criar /home/user/docs/report.txt e /home/user/code/main.java
 - **Comando:** ls /home true
 - **Saída Esperada:** Listagem em árvore de /home e todo seu conteúdo.
- **Cenário 4: Listar informações de um arquivo (ls em arquivo)**
 - **Usuário:** root
 - **Pré-condição:** /documento.txt existe.
 - **Comando:** ls /documento.txt
 - **Saída Esperada:** documento.txt (arquivo, <tamanho> bytes) - <permissões> <dono>
- **Cenário 5: Tentar listar caminho inexistente**
 - **Usuário:** root
 - **Comando:** ls /diretorio_fantasma
 - **Saída Esperada:** Erro: ls: '/diretorio_fantasma': Nenhum arquivo ou diretório encontrado. (CaminhoNaoEncontradoException)
- **Cenário 6: Tentar listar sem permissão de leitura (read) no diretório**
 - **Usuário:** root
 - **Pré-condição:** Criar /secretaria (mkdir /secretaria). Alterar permissão: chmod /secretaria root other --x (não permite leitura, só navegação).
 - **Trocar Usuário:** su atendente
 - **Comando:** ls /secretaria
 - **Saída Esperada:** Erro: Permissão negada: não pode ler diretório '/secretaria' para o usuário 'atendente'. (PermissaoException)
- **Cenário 7: Tentar listar sem permissão de execução (execute/navegação) no diretório**
 - **Usuário:** root
 - **Pré-condição:** Criar /restrito (mkdir /restrito). Alterar permissão: chmod /restrito root other rw- (permite leitura/escrita, mas não navegação).
 - **Trocar Usuário:** su visitante
 - **Comando:** ls /restrito
 - **Saída Esperada:** Erro: Permissão negada: não pode navegar (executar) no diretório '/restrito' para o usuário 'visitante'. (PermissaoException)

6. mv (Mover/Renomear Arquivo/Diretório)

Objetivo: Testar a movimentação e renomeação de itens, considerando todos os casos e permissões.

- **Cenário 1: Renomear arquivo**
 - **Usuário:** root
 - **Pré-condição:** touch /velho_nome.txt
 - **Comando:** mv /velho_nome.txt /novo_nome.txt
 - **Saída Esperada:** Item '/velho_nome.txt' movido/renomeado para '/novo_nome.txt'.
- **Cenário 2: Mover arquivo para outro diretório**
 - **Usuário:** root
 - **Pré-condição:** mkdir /destino_pasta. touch /origem_arquivo.txt
 - **Comando:** mv /origem_arquivo.txt /destino_pasta/origem_arquivo.txt
 - **Saída Esperada:** Arquivo '/origem_arquivo.txt' movido para '/destino_pasta/origem_arquivo.txt'.
- **Cenário 3: Renomear diretório**
 - **Usuário:** root
 - **Pré-condição:** mkdir /temp_dir_velho
 - **Comando:** mv /temp_dir_velho /temp_dir_novo
 - **Saída Esperada:** Diretório '/temp_dir_velho' movido e/ou renomeado para '/temp_dir_novo'.
- **Cenário 4: Mover diretório para outro diretório**
 - **Usuário:** root
 - **Pré-condição:** mkdir /projetoX. mkdir /pastas_comuns.
 - **Comando:** mv /projetoX /pastas_comuns/projetoX
 - **Saída Esperada:** Diretório '/projetoX' movido para '/pastas_comuns/projetoX'.
- **Cenário 5: Tentar mover caminho de origem inexistente**
 - **Usuário:** root
 - **Comando:** mv /nao_existe_origem /algum_destino
 - **Saída Esperada:** Erro: mv: '/nao_existe_origem': Nenhum arquivo ou diretório encontrado. (CaminhoNaoEncontradoException)
- **Cenário 6: Tentar mover para diretório pai de destino inexistente**
 - **Usuário:** root
 - **Pré-condição:** touch /temp_file.txt
 - **Comando:** mv /temp_file.txt /destino_inexistente/novo_nome.txt
 - **Saída Esperada:** Erro: Caminho de destino pai não existe para '/destino_inexistente/novo_nome.txt'. (CaminhoNaoEncontradoException)
- **Cenário 7: Tentar mover a raiz**

- **Usuário:** root
- **Comando:** mv / /nova_raiz
- **Saída Esperada:** Erro: Não é possível mover o diretório raiz. (PermissionException)
- **Cenário 8: Tentar mover sem permissão de escrita no diretório pai de origem**
 - **Usuário:** root
 - **Pré-condição:** mkdir /source_protected. touch /source_protected/my_doc.txt. chmod /source_protected root other r-x (sem escrita para 'other').
 - **Trocar Usuário:** su func
 - **Comando:** mv /source_protected/my_doc.txt /tmp/new_doc.txt (Assumindo /tmp tem escrita para func)
 - **Saída Esperada:** Erro: Permissão negada: não pode remover item em '/source_protected' para o usuário 'func'. (PermissionException)
- **Cenário 9: Tentar mover sem permissão de escrita no diretório pai de destino**
 - **Usuário:** root
 - **Pré-condição:** mkdir /dest_protected. chmod /dest_protected root other r-x. touch /meu_doc_src.txt.
 - **Trocar Usuário:** su func
 - **Comando:** mv /meu_doc_src.txt /dest_protected/meu_doc_dest.txt (Assumindo func pode remover /meu_doc_src.txt)
 - **Saída Esperada:** Erro: Permissão negada: não pode escrever no diretório de destino '/dest_protected' para o usuário 'func'. (PermissionException)
- **Cenário 10: Tentar mover arquivo para destino onde já existe um diretório**
 - **Usuário:** root
 - **Pré-condição:** touch /my_file. mkdir /my_dir.
 - **Comando:** mv /my_file /my_dir
 - **Saída Esperada:** Arquivo '/my_file' movido para '/my_dir/my_file'. (Neste caso, o mv move o arquivo *para dentro* do diretório. Se você tentasse mv /my_file /my_dir/my_dir_name, onde my_dir_name já é um diretório, o erro seria CaminhoJaExistenteException encapsulado em PermissionException).
 - **Cenário 10.1: Tentar mover arquivo para nome que já é um diretório**
 - **Usuário:** root
 - **Pré-condição:** touch /arquivo_src. mkdir /diretorio_destino.
 - **Comando:** mv /arquivo_src /diretorio_destino
 - **Saída Esperada:** Arquivo '/arquivo_src' movido para '/diretorio_destino/arquivo_src'. (Comportamento padrão: move para dentro)

- **Cenário 11: Tentar mover diretório para destino onde já existe um arquivo**
 - **Usuário:** root
 - **Pré-condição:** mkdir /dir_src. touch /file_dest.txt.
 - **Comando:** mv /dir_src /file_dest.txt
 - **Saída Esperada:** Erro: Não é possível mover diretório para um arquivo existente: '/file_dest.txt'. (PermissionException encapsulando CaminhoJaExistenteException)
- **Cenário 12: Tentar mover um diretório para dentro de um de seus subdiretórios**
 - **Usuário:** root
 - **Pré-condição:** mkdir /parent/child
 - **Comando:** mv /parent /parent/child
 - **Saída Esperada:** Erro: Não é possível mover um diretório para um de seus subdiretórios. (IllegalArgumentException)
- **Cenário 13: Mover arquivo sobre arquivo existente (sobrescrever)**
 - **Usuário:** root
 - **Pré-condição:** touch /file_to_move.txt. write /file_to_move.txt false "Conteúdo original.". touch /target_file.txt. write /target_file.txt false "Conteúdo a ser sobrescrito."
 - **Comando:** mv /file_to_move.txt /target_file.txt
 - **Saída Esperada:** Arquivo '/file_to_move.txt' movido e renomeado para '/target_file.txt'. (Conteúdo de /target_file.txt deve ser "Conteúdo original.").

7. cp (Copiar Arquivo/Diretório)

Objetivo: Testar a cópia de itens, recursivamente e com validação de permissão.

- **Cenário 1: Copiar arquivo (origem para novo arquivo)**
 - **Usuário:** root
 - **Pré-condição:** touch /original.txt. write /original.txt false "Dados importantes."
 - **Comando:** cp /original.txt /copia.txt
 - **Saída Esperada:** Arquivo 'original.txt' copiado para '/copia.txt'.
- **Cenário 2: Copiar arquivo (origem para diretório existente)**
 - **Usuário:** root
 - **Pré-condição:** touch /documento.txt. mkdir /backup.
 - **Comando:** cp /documento.txt /backup
 - **Saída Esperada:** Arquivo 'documento.txt' copiado para '/backup/'. (Resulta em /backup/documento.txt)
- **Cenário 3: Copiar arquivo (origem para arquivo existente - sobrescrever)**
 - **Usuário:** root

- **Pré-condição:** touch /src.txt. write /src.txt false "Conteúdo A". touch /dest.txt. write /dest.txt false "Conteúdo B"
- **Comando:** cp /src.txt /dest.txt
- **Saída Esperada:** Arquivo 'src.txt' copiado para '/dest.txt' (sobrescrito). (Verifique se dest.txt agora tem "Conteúdo A")
- **Cenário 4: Copiar diretório (recursivo)**
 - **Usuário:** root
 - **Pré-condição:** mkdir /projeto_antigo/src. touch /projeto_antigo/main.c. touch /projeto_antigo/src/helper.h. mkdir /novo_repositorio.
 - **Comando:** cp /projeto_antigo /novo_repositorio
 - **Saída Esperada:** Diretório '/projeto_antigo' e seu conteúdo copiados para '/novo_repositorio'. (Resulta em /novo_repositorio/projeto_antigo com todo o conteúdo)
- **Cenário 5: Tentar copiar origem inexistente**
 - **Usuário:** root
 - **Comando:** cp /nao_existe_origem /destino
 - **Saída Esperada:** Erro: cp: '/nao_existe_origem': Nenhum arquivo ou diretório encontrado. (CaminhoNaoEncontradoException)
- **Cenário 6: Tentar copiar para diretório pai de destino inexistente**
 - **Usuário:** root
 - **Pré-condição:** touch /temp_copy.txt
 - **Comando:** cp /temp_copy.txt /inexistente_dir/copia_final.txt
 - **Saída Esperada:** Erro: cp: Destino '/inexistente_dir/copia_final.txt': Caminho pai não existe. (CaminhoNaoEncontradoException)
- **Cenário 7: Tentar copiar sem permissão de leitura na origem**
 - **Usuário:** root
 - **Pré-condição:** touch /dados_senssiveis.txt. chmod /dados_senssiveis.txt root other ---.
 - **Trocar Usuário:** su intruso
 - **Comando:** cp /dados_senssiveis.txt /intruso_home/dados_roubados.txt
 - **Saída Esperada:** Erro: Permissão negada: não pode ler arquivo de origem '/dados_senssiveis.txt' para o usuário 'intruso'. (PermissaoException)
- **Cenário 8: Tentar copiar sem permissão de escrita no destino**
 - **Usuário:** root
 - **Pré-condição:** mkdir /area_restrita. chmod /area_restrita root other r-x. touch /meu_arquivinho.txt.
 - **Trocar Usuário:** su dev
 - **Comando:** cp /meu_arquivinho.txt /area_restrita/novo_arquivo.txt
 - **Saída Esperada:** Erro: Permissão negada: não pode escrever no diretório de

destino '/area_restrita' para o usuário 'dev'. (PermissaoException)

- **Cenário 9: Tentar copiar diretório para um arquivo existente**
 - **Usuário:** root
 - **Pré-condição:** mkdir /source_dir_cp. touch /target_file_cp.txt.
 - **Comando:** cp /source_dir_cp /target_file_cp.txt
 - **Saída Esperada:** Erro: Não é possível copiar diretório para um arquivo existente: '/target_file_cp.txt'. (CaminhoJaExistenteException)
- **Cenário 10: Tentar copiar um diretório para dentro de um de seus subdiretórios**
 - **Usuário:** root
 - **Pré-condição:** mkdir /principal/sub
 - **Comando:** cp /principal /principal/sub
 - **Saída Esperada:** Erro: Não é possível copiar um diretório para um de seus subdiretórios. (IllegalArgumentException)

8. chmod (Alterar Permissões)

Objetivo: Testar a alteração de permissões para arquivos e diretórios.

- **Cenário 1: Alterar permissão de um arquivo como dono**
 - **Usuário:** root
 - **Pré-condição:** touch /my_script.sh. write /my_script.sh false "echo 'Hello'"
 - **Comando:** chmod /my_script.sh root rwx
 - **Saída Esperada:** Permissões de '/my_script.sh' para o usuário 'root' alteradas para 'rwx'.
- **Cenário 2: Alterar permissão de um diretório como dono**
 - **Usuário:** root
 - **Pré-condição:** mkdir /shared_folder.
 - **Comando:** chmod /shared_folder root rwx
 - **Saída Esperada:** Permissões de '/shared_folder' para o usuário 'root' alteradas para 'rwx'.
- **Cenário 3: Alterar permissão como root (em item que não é seu)**
 - **Usuário:** root
 - **Pré-condição:** Logue como user1 e crie /user1_file.txt.
 - **Trocar Usuário:** su root
 - **Comando:** chmod /user1_file.txt user1 r--
 - **Saída Esperada:** Permissões de '/user1_file.txt' para o usuário 'user1' alteradas para 'r--'.
- **Cenário 4: Tentar alterar permissão de arquivo/diretório que não existe**
 - **Usuário:** root

- **Comando:** `chmod /nao_existe.txt root rwx`
- **Saída Esperada:** Erro: `chmod: '/nao_existe.txt': Nenhum arquivo ou diretório encontrado. (CaminhoNaoEncontradoException)`
- **Cenário 5: Tentar alterar permissão sem ser dono ou root**
 - **Usuário:** root
 - **Pré-condição:** `touch /file_owned_by_root.txt`.
 - **Trocar Usuário:** `su estranho`
 - **Comando:** `chmod /file_owned_by_root.txt estranho rwx`
 - **Saída Esperada:** Erro: Permissão negada: Somente o dono ou 'root' pode alterar permissões de '/file_owned_by_root.txt'. (`PermissaoException`)
- **Cenário 6: Tentar alterar permissão com formato inválido**
 - **Usuário:** root
 - **Pré-condição:** `touch /temp_file_perms.txt`.
 - **Comando:** `chmod /temp_file_perms.txt root rwx_extra`
 - **Saída Esperada:** Erro: Formato de permissão inválido. Use 'rwx', 'rw-', 'r-x', '---', etc. (`IllegalArgumentException`)