

Aluno: _____ Nota: _____

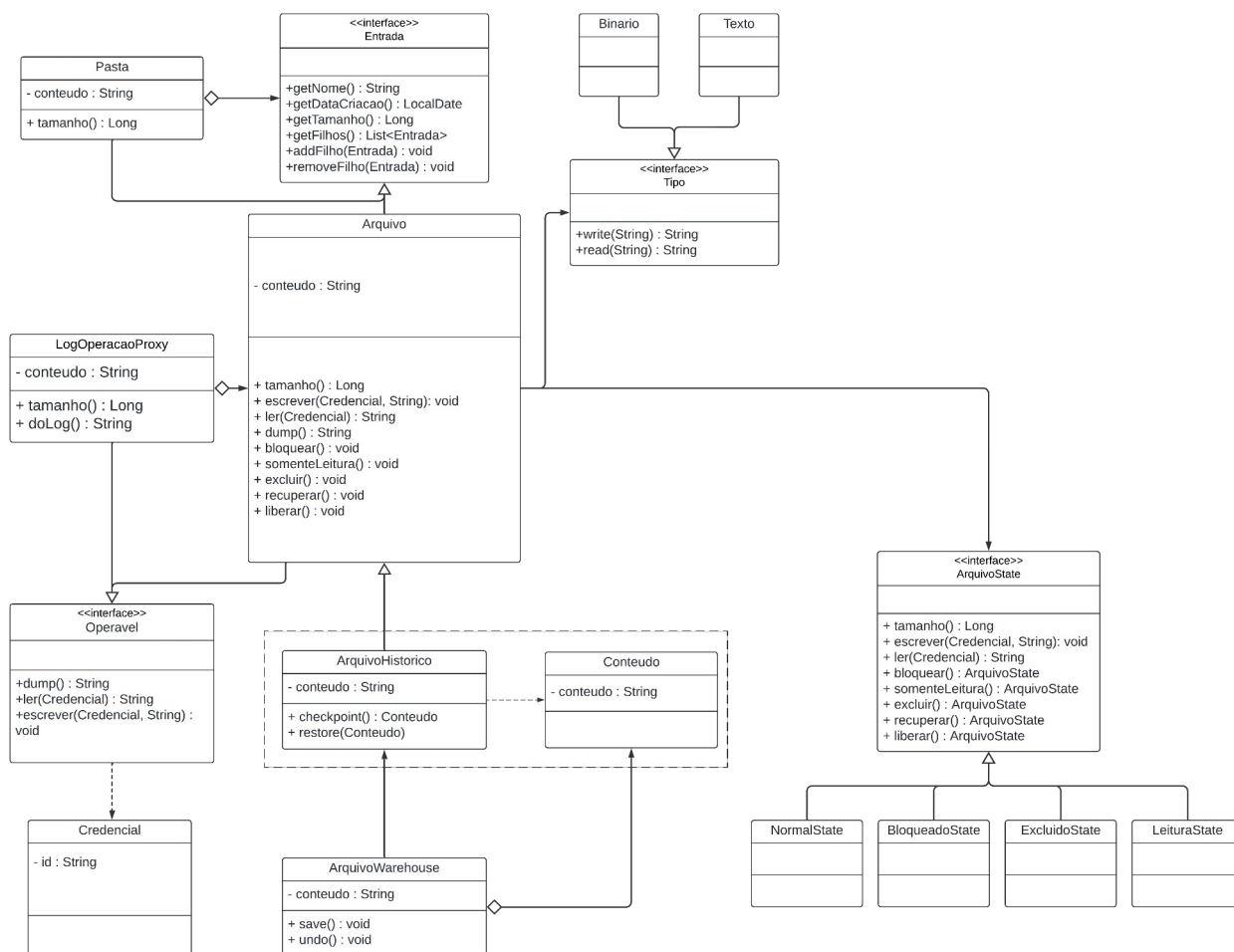
Avaliação Final

Você está trabalhando no desenvolvimento de um sistema de arquivos para um novo sistema operacional. O sistema precisa ser capaz de organizar arquivos e pastas de forma hierárquica, de forma similar aos sistemas de arquivos tradicionais, onde pastas podem armazenar arquivos, além de outras pastas. A forma de tratar as pastas e arquivos deve ser a mais transparente possível, para a classe cliente (Sistema Operacional).

O projeto deve contemplar também mecanismos de log para garantir o registro de acesso (leitura, escrita e dump) de alguns arquivos (protegidos). Estes registro de acesso deve registrar a credencial que realizou o acesso para leitura de cada arquivo protegido, bem como o número de leituras de cada arquivo protegido por cada credencial. O número de credenciais que podem acessar cada arquivo é arbitrário.

O sistema de arquivo foi construído de forma a possibilitar diversas representações internas do seu conteúdo, sem que isso implicasse em explosão de classes. Além disso, alguns arquivos podem ser mantidos com operações de desfazer e o comportamento do arquivo depende do seu ciclo de vida (exibido na máquina de estados da figura 2).

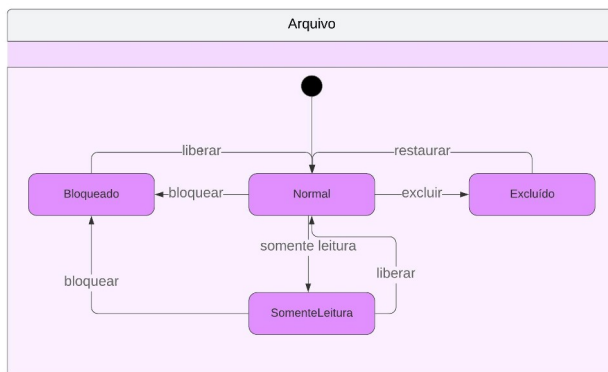
O diagrama de classes atual do sistema de arquivos é exibido na figura 1.



De modo a garantir os requisitos apresentados, a equipe de projeto optou pelo uso de Composite, Proxy, State, Bridge e Memento.

Nessa solução os participantes do

- Composite são: a interface *Entrada (Component)*, a classe *Pasta (Composite)* e a classe *Arquivo (Leaf)*;
- Proxy são: a interface *Operavel (ServiceInterface)*, a classe *LogOperacaoProxy (Proxy)* e a classe *Arquivo (Service)*;
- Bridge são *Arquivo (Abstraction)*, *Tipo (Implementation)*, *Binário e Texto (Concrete Implementations)*;
- Memento são: *ArquivoHistorico (Originator)*, *Conteudo (Memento)* e *ArquivoWarehouse (Caretaker)*
- State são: *Arquivo (Context)*, *ArquivoState (State)* e *NormalState, BloqueadoState, ExcluidoState e LeituraState (ConcreteStates)*.



Estado	Ler	Escrever	Tamanho
Normal	OK	OK	retorna o número de caracteres do conteúdo
Somente Leitura	OK	X	retorna o número de caracteres do conteúdo
Bloqueado	X	X	retorna o número de caracteres do conteúdo
Excluído	X	X	retorna 0

Neste ponto do projeto, novos requisitos listados a seguir foram apresentados. Sua tarefa é identificar os melhores padrões de projeto para implementar os requisitos elencados. A partir daí modificar (**apenas**) a estrutura das classes envolvidas e acrescentar outras que sejam necessárias para implementar os padrões escolhidos.

ATENÇÃO

- As soluções devem ser esboçadas em código JAVA.
- Identifique claramente o padrão que será aplicado em cada questão, bem como quais são os papéis (Participantes) no padrão, assumidos pelas classes do projeto.

NOVOS REQUISITOS

I – Deseja-se acrescentar meta-informações a alguns arquivos, estas meta-informações são exibidas antes ou depois de seu conteúdo, quando o arquivo é lido. São exemplos de meta-informações: Cabeçalhos, Assinaturas, Rodapés, Informações de Localização, etc. Alguns arquivos podem receber uma ou várias meta-informações. As meta-informações podem ser acrescentadas ou retiradas dinamicamente dos arquivos já existentes.

Como exemplo, implemente a meta-informação que permita adicionar uma assinatura a um arquivo (a assinatura é uma string exibida após o conteúdo do arquivo).

II – Deseja-se implementar um mecanismo de manutenção de consistência do estado de outras estruturas, que dependam do conteúdo de um arquivo. Por exemplo, um arquivo que tenha seu conteúdo modificado precisa ser atualizado em uma estrutura de backup incremental ou em um editor de conteúdo colaborativo.

Como exemplo, implemente um novo tipo de ArquivoWarehouse, que faça autosave (*ArquivoWarehouseAutoSave*). Neste caso, sempre que uma operação de escrever esteja acontecendo no arquivo, o seu Warehouse com AutoSave irá automaticamente disparar uma operação de save, de forma a armazenar o conteúdo do arquivo antes de sua modificação.