Arquitetura de Preservação Digital com Microsserviço Mapoteca e Kafka

Imagine seu sistema de preservação digital como uma **fábrica digital altamente automatizada**.

- O **Kafka** é o **sistema de esteiras e quadros de aviso digitais** que conecta todos os departamentos, permitindo que eles trabalhem de forma independente e eficiente.
- O **Microsserviço Mapoteca** é o **gerente de projetos central** da fábrica. Ele recebe todos os pedidos (novas "caixas" para processar, pedidos de "devolução", pedidos para "apagar" ou "renomear"), organiza o trabalho, delega tarefas aos departamentos corretos e acompanha o progresso, mantendo o Front-End atualizado.
- O **MinIO** é o **grande armazém** da fábrica, onde as "caixas" (arquivos) finais são guardadas.

Conceitos Essenciais para a Fábrica Digital

- **SIP (Submission Information Package - Pacote de Informação de Submissão):** É a **caixa original de arquivos e informações** que chega na "portaria" da fábrica (vem de fora). É o que o produtor entrega.
- **AIP (Archival Information Package - Pacote de Informação Arquivística):** É a **caixa de arquivos perfeitamente organizada, validada e preservada** pela fábrica. Contém os arquivos originais, as cópias para preservação e todos os "papéis" (metadados) que garantem sua autenticidade e história. É o produto final da preservação.
- **DIP (Dissemination Information Package - Pacote de Informação de Disseminação):** É uma **cópia dos arquivos pronta para ser entregue ao público**, muitas vezes em formatos fáceis de usar (ex: um PDF para leitura, em vez do formato de preservação). É o que a fábrica "expõe" ou "vende" para consulta.
- **Kafka (O Mensageiro Principal):** Um sistema de mensagens de alta performance. Departamentos (microsserviços) **publicam "notícias"** (mensagens/eventos) em **"canais" (tópicos do Kafka)**, e outros departamentos interessados **"assinam"** esses canais para receber as notícias e agir. Garante que nenhuma mensagem se perca e que a comunicação seja escalável.

Componentes da Arquitetura e Suas Funções

Camadas Externas

- **Front-End:** A interface que seus usuários e administradores utilizam. É a "mesa de controle" que envia todos os pedidos (upload, download, deleção, renomeação) diretamente ao **Microsserviço Mapoteca**.
- **Middleware (API Gateway/Proxy Reverso):** O "porteiro de segurança" da fábrica. É o único ponto de entrada para todas as requisições que vêm de fora (do Front-End). Ele as direciona para o **Microsserviço Mapoteca**. Também pode lidar com segurança inicial (autenticação básica) e direcionar o tráfego.

Microsserviços Node.js / TypeScript

Estes são os "departamentos" da fábrica que lidam com a **interação externa, gerenciam os pedidos de clientes, operam as portas de entrada e saída (I/O)** e atuam como **gerentes de nível superior** para orquestrar o trabalho. Node.js e TypeScript são ideais para essas tarefas.

1. **Microsserviço Mapoteca (O Gerente de Projetos Central)**
    - **Função Principal:** O **cérebro operacional**. É o **único serviço que o Front-End conversa**. Recebe todos os pedidos (upload, download, deleção, renomeação) e os gerencia de ponta a ponta. Ele não faz o trabalho pesado de processamento de arquivos, mas **orquestra** quem deve fazer, quando e o que. **Ele tem seu próprio banco de dados** para rastrear o status e detalhes de cada pedido.
    - **Detalhes:**
        - **API REST:** Recebe e responde a todas as requisições do Front-End.
        - **Gestão de Pedidos:** Registra, atualiza e consulta o estado de cada pedido (ID, status, progresso, etc.) em seu banco de dados.
        - **Orquestração:** Toma decisões sobre qual serviço deve ser acionado para cada etapa do fluxo (ex: Ingestão para iniciar, MinIO para armazenar, Gestão de Dados para atualizar).
        - **Comunicações:** Conversa diretamente (API REST) com Front-End, Ingestão, MinIO e Gestão de Dados. Assina tópicos do Kafka para receber notificações de eventos importantes de outros serviços (ex: Processamento concluído).
2. **Microsserviço de Ingestão (A Portaria da Fábrica)**
    - **Função Principal:** O "recepcionista". Recebe as **caixas originais (SIPs)** do **Microsserviço Mapoteca**, faz as primeiras checagens e as coloca na esteira de entrada para o próximo departamento.
    - **Detalhes:**
        - **API REST:** Recebe o SIP (arquivos e metadados) do Mapoteca.
        - **Armazenamento Temporário:** Salva os arquivos do SIP em um diretório temporário no servidor (área de "recebimento").
        - **Kafka:** Publica uma mensagem no tópico `ingest-requests` no Kafka, avisando o **Microsserviço de Processamento** que um novo SIP está disponível para trabalho.

- **Comunicações:** API REST (recebe do Mapoteca), Kafka (publica para Processamento).

3. **Microsserviço MinIO (O Zelador do Armazém)**
   - **Função Principal:** O guardião e operador do **armazém de objetos (MinIO)**. Ele só sabe como **colocar (upload), apagar** ou **pegar (download)** caixas (arquivos) no MinIO.
   - **Detalhes:**
     - **API REST Interna:** Expõe endpoints para todas as operações de arquivos (upload, download, delete).
     - **Restrição Importante: Só pode ser chamado pelo Microsserviço Mapoteca**. NENHUM outro microsserviço (Processamento, Gestão de Dados, Acesso) pode se comunicar diretamente com ele. Isso centraliza o controle do armazenamento no Mapoteca.
   - **Comunicações:** API REST (recebe comandos SOMENTE do Mapoteca).
4. **Microsserviço de Acesso (A Vitrine da Fábrica)**
   - **Função Principal:** Ajuda os "clientes" (usuários, ou o próprio Mapoteca) a encontrar e consultar informações sobre as **cópias para o público (DIPs)**.
   - **Detalhes:**
     - **API REST:** Responde a consultas sobre metadados dos DIPs (pedindo ao **Microsserviço de Gestão de Dados**).
     - **Integração com Mapoteca:** Quando um cliente quer baixar um arquivo (DIP), o Microsserviço de Acesso (ou o Front-End) solicita ao Mapoteca. O Mapoteca então coordena com o MinIO para o download.
   - **Comunicações:** API REST (Front-End, Mapoteca, Gestão de Dados).

Microsserviços Python

Estes são os departamentos que lidam com o **trabalho pesado de processamento de dados**, a **inteligência de preservação** e a **gestão central de informações**, onde Python tem um ecossistema mais forte para manipular arquivos e dados complexos.

1. **Microsserviço de Processamento (O Departamento de Qualidade e Montagem)**
    - **Função Principal:** O "operário" da preservação. Pega as **caixas originais (SIPs)** da esteira do Kafka, as processa e as transforma em **caixas perfeitamente preservadas (AIPs)**.
    - **Detalhes:**
        - **Kafka:** Assina o tópico `ingest-requests` para receber mensagens de novos SIPs.
        - **Trabalho Pesado:**
            - Calcula **Checksums** (a "impressão digital" do arquivo) para verificar sua integridade.
            - **Identificação de Formatos:** Tenta descobrir o tipo exato de cada arquivo (ex: é um PDF? É um JPEG?) usando bibliotecas Python puras (nota: **esta identificação será mais básica sem ferramentas especializadas**).
            - **Validação de Formatos:** Verifica se o arquivo parece "saudável" e não tem erros óbvios de estrutura (nota: **esta validação será limitada em profundidade sem ferramentas como JHOVE**).
            - **Caracterização:** Extrai metadados técnicos (tamanho, data de criação, etc.).
            - **Normalização:** Converte os arquivos para formatos mais seguros para preservação ou mais fáceis de acessar (ex: um documento de texto para PDF/A ou TXT puro), usando bibliotecas Python como Pillow para imagens.
        - **Criação do AIP:** Monta a estrutura final do **AIP** (pasta, arquivos, logs) e gera os "papéis" importantes de metadados como **METS e PREMIS** (usando bibliotecas XML do Python).
        - **Comunicação (API):** Envia todos os metadados gerados (checksums, formatos, etc.) para o **Microsserviço de Gestão de Dados** para registro.
        - **Comunicação (API): Notifica o Microsserviço Mapoteca** que o **AIP** está pronto e onde ele está temporariamente no servidor (para o Mapoteca mandar para o MinIO).
    - **Comunicações:** Kafka (consome da Ingestão), API REST (chama Gestão de Dados, chama Mapoteca).
2. **Microsserviço de Gestão de Dados (O Arquivo Central de Informações)**
    - **Função Principal:** O "historiador" da fábrica. É o guardião de **TODAS as informações e metadados** sobre cada SIP, AIP e DIP. Responde a perguntas sobre esses dados.
    - **Detalhes:**
        - **API REST:** Expõe endpoints para criar, ler, atualizar e marcar para deletar (deleção lógica) metadados.

        - **Banco de Dados:** Guarda todas essas informações em um banco de dados (ex: PostgreSQL para dados estruturados, e/ou Elasticsearch para buscas rápidas em texto).
        - **Deleção Lógica:** Ao receber um pedido do Mapoteca para deletar, ele apenas "marca" o item como deletado em seu registro, mas **não apaga os arquivos físicos**. Ele informa ao Mapoteca quais arquivos no MinIO precisam ser apagados.
        - **Renomeação Lógica:** Ao receber um pedido de renomeação do Mapoteca, ele atualiza o nome do item em seu banco de dados.
    - **Comunicações:** API REST (interage com Mapoteca, Processamento, Acesso, Planejamento).
3. **Microsserviço de Planejamento de Preservação (O Analista de Tendências)**
    - **Função Principal:** O "futurólogo" da fábrica. Monitora as tecnologias e formatos digitais para prever quais podem se tornar obsoletos no futuro e planejar como preservar as informações mesmo assim.
    - **Detalhes:** Pode analisar os formatos que a fábrica está guardando (consultando a **Gestão de Dados**) e sugerir ações.
    - **Comunicações:** API REST (chama Gestão de Dados).

Fluxos de Comunicação Detais

O Papel do Kafka (Tópicos Principais)

- `ingest-requests` **(Tópico Kafka):** Onde o **Microsserviço de Ingestão** publica que um novo SIP está pronto. O **Microsserviço de Processamento** assina este tópico.

Comunicação Síncrona (APIs REST):

- **Front-End** <-> **Middleware** <-> **Microsserviço Mapoteca** (TUDO passa por aqui)

- **Microsserviço Mapoteca** <-> **Microsserviço de Ingestão** (para enviar os dados do SIP no início)
- **Microsserviço Mapoteca** <-> **Microsserviço MinIO** (para todas as operações de armazenamento: upload final, download, deleção física)
- **Microsserviço Mapoteca** <-> **Microsserviço de Gestão de Dados** (para consultar metadados, iniciar deleção lógica, iniciar renomeação lógica)
- **Microsserviço Mapoteca** <-> **Microsserviço de Processamento** (para receber a notificação de AIP pronto)
- **Microsserviço de Processamento (Python)** <-> **Microsserviço de Gestão de Dados (Python)** (para enviar e atualizar metadados)
- **Microsserviço de Acesso (TS)** <-> **Microsserviço de Gestão de Dados (Python)** (para buscar metadados para exibição)
- **Microsserviço de Planejamento de Preservação (Python)** <-> **Microsserviço de Gestão de Dados (Python)** (para consultar metadados de formatos).

Fluxos Detalhados de Operação

2.1. Fluxo de Upload (Ingestão)

1. **Front-End** envia o **SIP** (arquivos + metadados) `--- HTTP --->` **Middleware** `--- HTTP --->` **Microsserviço Mapoteca (TS)**.
2. **Microsserviço Mapoteca (TS)**: Registra o pedido de upload. `--- HTTP (API Interna) --->` **Microsserviço de Ingestão (TS)** (passa os arquivos e metadados).
3. **Microsserviço de Ingestão (TS)**: Salva o SIP temporariamente. `--- Publica no Kafka (tópico: ingest-requests) --->` **Kafka**.
4. **Kafka** `--- Entrega Mensagem --->` **Microsserviço de Processamento (Python)**.

5. **Microsserviço de Processamento (Python)**: Consome a mensagem. Realiza checksums, identificação/validação (Python puro, limitada), caracterização, normalização, e cria o **AIP** temporariamente.
6. **Microsserviço de Processamento (Python)** `--- HTTP (API) --->` **Microsserviço de Gestão de Dados (Python)** (envia todos os metadados gerados do AIP para registro).
7. **Microsserviço de Processamento (Python)** `--- HTTP (API) --->` **Microsserviço Mapoteca (TS)** (notifica que o AIP está pronto e informa sua localização temporária).
8. **Microsserviço Mapoteca (TS)**: Ao receber a notificação, `--- HTTP (API) --->` **Microsserviço MinIO (TS)** (ordena o upload do AIP para o armazenamento final no MinIO).
9. **Microsserviço MinIO (TS)**: Executa o upload. `--- HTTP (API) --->` **Microsserviço Mapoteca (TS)** (confirma o upload bem-sucedido).
10. **Microsserviço Mapoteca (TS)**: Atualiza o status final do pedido em seu próprio BD e, opcionalmente, notifica o Front-End sobre a conclusão.

2.2. Fluxo de Download (Acesso)

1. **Front-End** solicita um download `--- HTTP --->` **Middleware** `--- HTTP --->` **Microsserviço Mapoteca (TS)**.

2. **Microsserviço Mapoteca (TS)**: Registra o pedido de download. `--- HTTP (API) --->` **Microsserviço de Gestão de Dados (Python)** (solicita metadados e localização do DIP/AIP pelo ID).
3. **Microsserviço de Gestão de Dados (Python)**: Retorna os metadados e a localização (bucket/path no MinIO) do arquivo solicitado.
4. **Microsserviço Mapoteca (TS)**: `--- HTTP (API) --->` **Microsserviço MinIO (TS)** (ordena o download do arquivo do MinIO).
5. **Microsserviço MinIO (TS)**: Recupera o arquivo do MinIO. `--- HTTP (stream) --->` **Microsserviço Mapoteca (TS)**.
6. **Microsserviço Mapoteca (TS)**: `--- HTTP (stream) --->` **Front-End** (entrega o arquivo ao usuário).

2.3. Fluxo de Deleção

1. **Front-End** envia pedido de deleção (ex: `DELETE /items/:id`) `--- HTTP --->` **Middleware** `--- HTTP --->` **Microsserviço Mapoteca (TS)**.

2. **Microsserviço Mapoteca (TS)**: Registra o pedido de deleção. `--- HTTP (API) --->` **Microsserviço de Gestão de Dados (Python)** (solicita a "deleção lógica" do item, enviando o ID).
3. **Microsserviço de Gestão de Dados (Python)**: Marca o item como `deleted` em seu banco de dados (deleção lógica). **Retorna ao Mapoteca** os IDs ou caminhos de todos os objetos (AIP, DIP, etc.) no MinIO que devem ser fisicamente excluídos.
4. **Microsserviço Mapoteca (TS)**: Recebe a lista de objetos MinIO. Para cada objeto na lista, `--- HTTP (API) --->` **Microsserviço MinIO (TS)** (ordena a exclusão física do objeto).
5. **Microsserviço MinIO (TS)**: Executa a exclusão no armazenamento MinIO. `--- HTTP (API) --->` **Microsserviço Mapoteca (TS)** (confirma a exclusão de cada objeto).
6. **Microsserviço Mapoteca (TS)**: Ao ter a confirmação de todas as exclusões físicas, `--- HTTP (API) --->` **Microsserviço de Gestão de Dados (Python)** (confirma a deleção física total para o item).
7. **Microsserviço de Gestão de Dados (Python)**: Atualiza o status do item para `physically_deleted`.
8. **Microsserviço Mapoteca (TS)**: Atualiza o status do pedido em seu próprio BD e notifica o Front-End sobre o sucesso.

2.4. Fluxo de Renomeação

1. **Front-End** envia pedido de renomeação (ex: `PUT /items/:id/rename`, com o novo nome) `--- HTTP --->` **Middleware** `--- HTTP --->` **Microsserviço Mapoteca (TS)**.

2. **Microsserviço Mapoteca (TS)**: Registra o pedido de renomeação. `--- HTTP (API) --->` **Microsserviço de Gestão de Dados (Python)** (solicita a atualização do nome para o item com o ID fornecido).
3. **Microsserviço de Gestão de Dados (Python)**: Localiza o item pelo ID em seu banco de dados e **atualiza o campo de nome descritivo** ou caminho lógico (metadado).
4. **Microsserviço de Gestão de Dados (Python)**: Retorna a confirmação de sucesso.
5. **Microsserviço Mapoteca (TS)**: Recebe a confirmação, atualiza o status do pedido em seu próprio BD e, opcionalmente, notifica o Front-End.
    - **Nota:** Esta renomeação é sobre o **nome descritivo** ou **caminho lógico** do item no sistema de metadados. Não implica em mover ou renomear arquivos físicos no MinIO, pois a preservação se baseia em identificadores persistentes (UUIDs, hashes).

Arquitetura de Preservação Digital com Microsserviço Mapoteca e Kafka

Imagine seu sistema de preservação digital como uma **fábrica digital altamente automatizada**.

- O **Kafka** é o **sistema de esteiras e quadros de aviso digitais** que conecta todos os departamentos, permitindo que eles trabalhem de forma independente e eficiente.
- O **Microsserviço Mapoteca** é o **gerente de projetos central** da fábrica. Ele recebe todos os pedidos (novas "caixas" para processar, pedidos de "devolução", pedidos para "apagar" ou "renomear"), organiza o trabalho, delega tarefas aos departamentos corretos e acompanha o progresso, mantendo o Front-End atualizado.
- O **MinIO** é o **grande armazém** da fábrica, onde as "caixas" (arquivos) finais são guardadas.

Conceitos Essenciais para a Fábrica Digital

- **SIP (Submission Information Package - Pacote de Informação de Submissão):** É a **caixa original de arquivos e informações** que chega na "portaria" da fábrica (vem de fora). É o que o produtor entrega.
- **AIP (Archival Information Package - Pacote de Informação Arquivística):** É a **caixa de arquivos perfeitamente organizada, validada e preservada** pela fábrica. Contém os arquivos originais, as cópias para preservação e todos os "papéis" (metadados) que garantem sua autenticidade e história. É o produto final da preservação.
- **DIP (Dissemination Information Package - Pacote de Informação de Disseminação):** É uma **cópia dos arquivos pronta para ser entregue ao público**, muitas vezes em formatos fáceis de usar (ex: um PDF para leitura, em vez do formato de preservação). É o que a fábrica "expõe" ou "vende" para consulta.
- **Kafka (O Mensageiro Principal):** Um sistema de mensagens de alta

performance. Departamentos (microsserviços) **publicam "notícias"** (mensagens/eventos) em **"canais" (tópicos do Kafka)**, e outros departamentos interessados **"assinam"** esses canais para receber as notícias e agir. Garante que nenhuma mensagem se perca e que a comunicação seja escalável.

Componentes da Arquitetura e Suas Funções

Camadas Externas

- **Front-End:** A interface que seus usuários e administradores utilizam. É a "mesa de controle" que envia todos os pedidos (upload, download, deleção, renomeação) diretamente ao **Microsserviço Mapoteca**.
- **Middleware (API Gateway/Proxy Reverso):** O "porteiro de segurança" da fábrica. É o único ponto de entrada para todas as requisições que vêm de fora (do Front-End). Ele as direciona para o **Microsserviço Mapoteca**. Também pode lidar com segurança inicial (autenticação básica) e direcionar o tráfego.

Microsserviços Node.js / TypeScript

Estes são os "departamentos" da fábrica que lidam com a **interação externa, gerenciam os pedidos de clientes, operam as portas de entrada e saída (I/O)** e atuam como **gerentes de nível superior** para orquestrar o trabalho. Node.js e TypeScript são ideais para essas tarefas.

1. ## Microsserviço Mapoteca (O Gerente de Projetos Central)
   - **Função Principal:** O **cérebro operacional**. É o **único serviço que o Front-End conversa**. Recebe todos os pedidos (upload, download, deleção, renomeação) e os gerencia de ponta a ponta. Ele não faz o trabalho pesado de processamento de arquivos, mas **orquestra** quem deve fazer, quando e o que. **Ele tem seu próprio banco de dados** para rastrear o status e detalhes de cada pedido.
   - ### Detalhes:
     - **API REST:** Recebe e responde a todas as requisições do Front-End.
     - **Gestão de Pedidos:** Registra, atualiza e consulta o estado de cada pedido (ID, status, progresso, etc.) em seu banco de dados.
     - **Orquestração:** Toma decisões sobre qual serviço deve ser acionado para cada etapa do fluxo (ex: Ingestão para iniciar, MinIO para armazenar, Gestão de Dados para atualizar).

        - **Comunicações:** Conversa diretamente (API REST) com Front-End, Ingestão, MinIO e Gestão de Dados. Assina tópicos do Kafka para receber notificações de eventos importantes de outros serviços (ex: Processamento concluído).
2. **Microsserviço de Ingestão (A Portaria da Fábrica)**
    - **Função Principal:** O "recepcionista". Recebe as **caixas originais (SIPs)** do **Microsserviço Mapoteca**, faz as primeiras checagens e as coloca na esteira de entrada para o próximo departamento.
    - **Detalhes:**
        - **API REST:** Recebe o SIP (arquivos e metadados) do Mapoteca.
        - **Armazenamento Temporário:** Salva os arquivos do SIP em um diretório temporário no servidor (área de "recebimento").
        - **Kafka:** Publica uma mensagem no tópico **ingest-requests** no Kafka, avisando o **Microsserviço de Processamento** que um novo SIP está disponível para trabalho.
    - **Comunicações:** API REST (recebe do Mapoteca), Kafka (publica para Processamento).
3. **Microsserviço MinIO (O Zelador do Armazém)**
    - **Função Principal:** O guardião e operador do **armazém de objetos (MinIO)**. Ele só sabe como **colocar (upload), apagar** ou **pegar (download)** caixas (arquivos) no MinIO.
    - **Detalhes:**
        - **API REST Interna:** Expõe endpoints para todas as operações de arquivos (upload, download, delete).
        - **Restrição Importante: Só pode ser chamado pelo Microsserviço Mapoteca**. NENHUM outro microsserviço (Processamento, Gestão de Dados, Acesso) pode se comunicar diretamente com ele. Isso centraliza o controle do armazenamento no Mapoteca.
    - **Comunicações:** API REST (recebe comandos SOMENTE do Mapoteca).
4. **Microsserviço de Acesso (A Vitrine da Fábrica)**
    - **Função Principal:** Ajuda os "clientes" (usuários, ou o próprio Mapoteca) a encontrar e consultar informações sobre as **cópias para o público (DIPs)**.
    - **Detalhes:**
        - **API REST:** Responde a consultas sobre metadados dos DIPs (pedindo ao **Microsserviço de Gestão de Dados**).
        - **Integração com Mapoteca:** Quando um cliente quer baixar

um arquivo (DIP), o Microsserviço de Acesso (ou o Front-End) solicita ao Mapoteca. O Mapoteca então coordena com o MinIO para o download.

- **Comunicações:** API REST (Front-End, Mapoteca, Gestão de Dados).

Microsserviços Python

Estes são os departamentos que lidam com o **trabalho pesado de processamento de dados**, a **inteligência de preservação** e a **gestão central de informações**, onde Python tem um ecossistema mais forte para manipular arquivos e dados complexos.

1. ## Microsserviço de Processamento (O Departamento de Qualidade e Montagem)
   - **Função Principal:** O "operário" da preservação. Pega as **caixas originais (SIPs)** da esteira do Kafka, as processa e as transforma em **caixas perfeitamente preservadas (AIPs)**.
   - ## Detalhes:
     - **Kafka:** Assina o tópico **ingest-requests** para receber mensagens de novos SIPs.
     - ## Trabalho Pesado:
       - Calcula **Checksums** (a "impressão digital" do arquivo) para verificar sua integridade.
       - **Identificação de Formatos:** Tenta descobrir o tipo exato de cada arquivo (ex: é um PDF? É um JPEG?) usando bibliotecas Python puras (nota: **esta identificação será mais básica sem ferramentas especializadas**).
       - **Validação de Formatos:** Verifica se o arquivo parece "saudável" e não tem erros óbvios de estrutura (nota: **esta validação será limitada em profundidade sem ferramentas como JHOVE**).
       - **Caracterização:** Extrai metadados técnicos (tamanho, data de criação, etc.).
       - **Normalização:** Converte os arquivos para formatos mais seguros para preservação ou mais fáceis de acessar (ex: um documento de texto para PDF/A ou TXT puro), usando bibliotecas Python como Pillow para imagens.
     -