Especificação do Modelo de Domínio - AxonAI

1. Objetivo

Este documento descreve o modelo de domínio revisado do AxonAI, incorporando princípios de escalabilidade, consistência e extensibilidade. Ele reflete a evolução a partir do MVP para uma base sólida de produção.

2. Agregados Principais

- **ProjectAggregate**
  - **Campos:**
    - `projectId`
    - `name`
    - `ownerId`
    - `taskIds: List<TaskId>`
    - `createdAt`, `updatedAt`, `version`
  - **Responsabilidade:** Representa o projeto e mantém apenas referências para tarefas.
  - **Métodos principais:**
    - `rename(String newName)`
    - `addTask(TaskId taskId)`
    - `archive(UserId actor)`
  - **Eventos:** `ProjectCreated`, `TaskAddedToProject`, `ProjectArchived`
- **TaskAggregate**
  - **Campos:**
    - `taskId`
    - `projectId`
    - `title`

        - `status: TaskStatus`
        - `checklistItemIds: List<ChecklistItemId>`
        - `mainConversationId`
        - `createdAt`, `updatedAt`, `version`
    - **Responsabilidade:** Gerencia o ciclo de vida da tarefa e a consistência de seus `ChecklistItems`.
    - **Estados permitidos:**
        - BACKLOG → RUNNING | CANCELED
        - RUNNING → DONE | PAUSED | CANCELED
        - PAUSED → RUNNING | CANCELED
        - DONE → RUNNING (reopen)
    - **Métodos principais:**
        - `start(UserId actor)`
        - `complete(UserId actor)`
        - `addChecklistItem(String description, UserId actor)`
        - `focusOnChecklistItem(ChecklistItemId id, UserId actor)`
    - **Eventos:** `TaskCreated`, `TaskStarted`, `TaskCompleted`, `ChecklistItemAdded`, `ChecklistItemFocused`
- **ChecklistItem**
    - **Tipo:** Entidade (não é mais Value Object)
    - **Campos:**
        - `checklistItemId`
        - `taskId`
        - `description`
        - `status: ChecklistItemStatus`
        - `focusedConversationId`
        - `createdAt`, `updatedAt`
    - **Estados permitidos:**
        - PENDING → FOCUSED | PAUSED | DONE

        - FOCUSED → DONE | PAUSED
        - PAUSED → FOCUSED | PENDING
    - **Métodos principais:**
        - `markFocused(UserId actor)`
        - `markDone(UserId actor)`
        - `pause(UserId actor)`
- **ConversationAggregate**
    - **Campos:**
        - `conversationId`
        - `ownerRef` (TaskId ou ChecklistItemId)
        - `createdAt`, `lastMessageAt`, `messageCount`
    - **Mensagens:** armazenadas separadamente em `ConversationMessage` (append-only), com: `messageId`, `conversationId`, `author`, `content`, `timestamp`.
    - **Benefício:** Escalabilidade e leitura paginada.
- **AIInteraction**
    - **Campos:**
        - `interactionId`
        - `conversationId`
        - `provider`, `model`, `tokensIn`, `tokensOut`, `estimatedCost`
        - `timestamp`, `requestHash`, `responseHash`
    - **Função:** Auditoria e controle de custos de uso da IA.

3. Regras de Negócio e Validação

- **Transições de estado controladas:** Métodos específicos nos agregados lançam `DomainException` se violarem regras de transição de estado.
- **Criação de Tarefas:** Tasks só podem ser criadas em projetos que não estejam arquivados.
- **Invariante de Foco Único (Regra Crítica):**
    - A regra de que

"apenas um `ChecklistItem` pode estar no estado `FOCUSED` por `Task`" é uma invariante protegida pelo `TaskAggregate`.

- A única maneira de alterar o foco é através do método `TaskAggregate::focusOnChecklistItem(checklistItemId, actor)`. Este método é responsável por orquestrar a transição de forma atômica, garantindo a consistência do agregado da seguinte forma:
  1. Verificar se a `Task` está em um estado que permite a operação (ex: `RUNNING`).
  2. Localizar e "desfocar" (transitar para `PENDING` ou `PAUSED`) qualquer `ChecklistItem` que esteja atualmente no estado `FOCUSED` dentro da mesma tarefa.
  3. Mover o `ChecklistItem` alvo para o estado `FOCUSED`.
  4. Emitir um evento de domínio `ChecklistItemFocused` para notificar outras partes do sistema.
- Qualquer outra tentativa de alterar o estado de um `ChecklistItem` para `FOCUSED` diretamente deve ser rejeitada pelo domínio.

4. Glossário da Linguagem Onipresente

Este glossário define os termos centrais que conectam a visão de negócio à implementação do domínio.

- **Modo Foco (Focus Mode):**
  - **Descrição de Negócio:** É a funcionalidade principal do AxonAI, que permite ao usuário iniciar uma conversa com a IA isolada e focada no contexto de um único `ChecklistItem`. Todo o histórico da tarefa pai é herdado, mas a nova interação é específica para a microtarefa.
  - **Implementação no Domínio:** O "Modo Foco" é ativado pelo método `TaskAggregate::focusOnChecklistItem()` e representado pelo estado `ChecklistItemStatus.FOCUSED` na entidade `ChecklistItem`. A conversa associada a este modo é identificada pelo `focusedConversationId`.

5. Eventos de Domínio

Eventos serão publicados de forma assíncrona e idempotente para integração com:

- Kanban externo (Trello, Jira, Linear)

- Notificações
- Métricas de uso de IA
- **Principais eventos:**
  - `ProjectCreated`
  - `TaskCreated`
  - `ChecklistItemFocused`
  - `ConversationMessageAdded`
  - `AIInteractionLogged`

6. Persistência e Concorrência

- **Persistência:** JPA/Hibernate com `@Version` para controle otimista de concorrência.
- **Estrutura:**
  - Conversas: armazenadas em coleção/tabela própria, paginada.
  - Tasks: tabela própria com chave estrangeira para `projectId`.
  - ChecklistItems: tabela própria com chave estrangeira para `taskId`.

7. Estratégia de Leituras e Evolução para CQRS

A arquitetura está preparada para a eventual implementação de CQRS, mas a abordagem inicial priorizará a simplicidade para acelerar a entrega de valor. A estratégia será dividida em fases:

- **Fase 1 (Implementação Inicial):** As consultas de leitura (queries) para visualizações complexas, como o `ProjectBoardView` (Kanban), serão realizadas diretamente contra o modelo de persistência dos agregados. Serão criados DTOs otimizados para essas leituras, mas sem a complexidade de um modelo de dados separado ou de atualização por eventos.
- **Fase 2 (Evolução Acionada por Métricas):** A transição para um modelo completo de CQRS, com projeções de leitura (`Read Models`) dedicadas e atualizadas por eventos de domínio, será considerada uma refatoração estratégica. A decisão de iniciar esta fase será acionada por um dos seguintes **gatilhos baseados em dados de monitoramento**:

- A latência no p95 (percentil 95) para as APIs de leitura de painéis (ex: carregar um projeto) exceder consistentemente **500ms**.
- A complexidade das consultas diretas ao banco de dados começar a impactar negativamente a performance das operações de escrita (comandos).

8. Segurança e Retenção

- **BYOK:** chaves criptografadas via Envelope Encryption + KMS.
- **Logs:** sem chaves em logs.
- **Retenção de conversas:** política configurável com suporte a exclusão (LGPD/GDPR).

9. Benefícios da Revisão

- Evita agregados inchados (Project não carrega milhares de tasks na memória).
- Permite escala horizontal de tasks e conversas.
- Melhora governança (auditoria de IA, controle de transições de estado).
- Abre caminho para monetização futura via `AIInteraction`.
