

PRIORIZAÇÃO DE REQUISITOS

Priorização de Requisitos do Projeto Kronos (Método MoSCoW)

Este documento utiliza o método MoSCoW para classificar os requisitos funcionais do projeto Kronos, com o objetivo de definir o escopo do Produto Mínimo Viável (MVP) e planejar as fases seguintes do desenvolvimento.

1. Must-Have (Tem que ter)

Estes requisitos são **essenciais e inegociáveis** para a primeira versão do produto. Sem eles, o sistema não funciona ou não entrega o seu valor principal. O foco aqui é construir o fluxo completo da **História de Utilizador 1 (HU1)**.

ID	Requisito	Justificação
RF01	Upload de Mídia	Ponto de partida de toda a operação. Sem upload, nada acontece.
RF02	Extração de Áudio	Lógica central para processar vídeos, que é o caso de uso principal.
RF03	Iniciar Transcrição	Ação fundamental do utilizador para começar o processo.
RF04	Integração com a API de Transcrição	O coração técnico do projeto; sem a API externa, não há transcrição.
RF05	Armazenamento da Transcrição	É crucial persistir o resultado para que o utilizador possa vê-lo.
RF06	Prevenção de Duplicidade	Requisito de negócio chave para otimizar custos e evitar reprocessamento.
RF07	Exibição do Resultado	O utilizador precisa de ver o produto final do seu pedido.
RF10	Tratamento de Erros	Garante uma experiência minimamente funcional, informando o utilizador sobre falhas básicas (ex: formato inválido).

2. Should-Have (Deveria ter)

Estes requisitos são muito importantes, mas não são vitais para a primeira versão. Devem ser implementados logo após o MVP para enriquecer a experiência do utilizador. Correspondem, em grande parte, à **História de Utilizador 2 (HU2)**.

ID	Requisito	Justificação
RF09	Listagem de Transcrições	Adiciona um grande valor, permitindo que os utilizadores revejam trabalhos anteriores, mas o sistema funciona sem isso inicialmente.

| ****RF08**** | Feedback de Processamento | Melhora significativamente a usabilidade, mas um simples "a carregar..." é suficiente para o MVP. Um feedback mais detalhado pode vir depois. |

3. Could-Have (Poderia ter)

Estes requisitos são desejáveis, mas menos importantes. Podem ser vistos como melhorias que seriam implementadas se houvesse tempo e recursos disponíveis, após a entrega dos "Must" e "Should".

| ID | Requisito (Sugestão) | Justificação |

| :--- | :--- | :--- |

| ****CH01**** | Editar Transcrição | Permitir que o utilizador corrija pequenos erros no texto gerado pela IA. |

| ****CH02**** | Exportar para .txt ou .docx | Adicionar a funcionalidade de exportar o texto para diferentes formatos de ficheiro. |

| ****CH03**** | Suporte a mais formatos de multimédia | Expandir a compatibilidade para além dos formatos mais comuns. |

| ****CH04**** | Notificação de Conclusão | Enviar uma notificação no navegador quando uma transcrição longa for concluída. |

4. Won't-Have (Não terá - por agora)

Funcionalidades que foram consideradas, mas que estão explicitamente fora do escopo do projeto inicial para evitar o "scope creep" (aumento descontrolado do escopo).

| ID | Requisito (Excluído) | Justificação |

| :--- | :--- | :--- |

| ****WH01**** | Contas de Utilizador e Autenticação | Aumenta muito a complexidade. A versão inicial será de uso público e anónimo. |

| ****WH02**** | Transcrição em Tempo Real (Live) | Requer uma arquitetura completamente diferente (websockets, streaming) e está fora do objetivo inicial. |

| ****WH03**** | Funcionalidades Colaborativas | Partilhar e editar transcrições em equipa está fora do escopo. |

| ****WH04**** | Suporte a Múltiplos Idiomas | O foco inicial será apenas num idioma (ex: português ou inglês) para simplificar a integração com a API. |

Conclusão do Planeamento

O plano de desenvolvimento deve focar-se em entregar todos os requisitos ****Must-Have**** primeiro para construir um MVP robusto. Em seguida, o foco deve passar para os itens ****Should-Have**** para completar as funcionalidades mais pedidas.

Revision #1

Created 10 October 2025 16:47:35 by Paulo Cerqueira

Updated 10 October 2025 16:48:08 by Paulo Cerqueira