

Diário de Bordo: Projeto Gerador de Equipes de IA

Este documento registra as etapas de desenvolvimento, desafios encontrados, soluções aplicadas e próximos passos para o seu projeto de Gerador de Equipes de IA com CrewAI.

Fase 0: Preparação do Ambiente e Ferramentas

Problema Inicial (1): ImportError: cannot import name 'tool' from 'crewai_tools' e PydanticDeprecatedSince20 warnings.

Solução Aplicada (1): Corrigido o import da ferramenta tool de crewai_tools para crewai.tools em equipe_mestre/ferramentas.py.

Resultado (1): ImportError resolvido, permitindo o início da execução da Crew.

Fase 1: Interface do Usuário (Streamlit)

Status: A interface Streamlit (app.py) foi planejada e implementada com campos para o prompt do usuário, uploader de arquivo e área de saída.

Observações: A interface tem sido a base para todos os testes e visualização das saídas.

Fase 2: Definição da Equipe-Mestre (5 Agentes)

Status: Todos os 5 agentes (Analista, Especialista, Designer, Implementador, Validador) estão definidos em `equipe_mestre/agentes.py` com seus papéis, objetivos e ferramentas.

Fase 3: Orquestração e Correção do Fluxo de Dados

Problema Crítico Identificado: O output de cada agente não estava sendo passado como input (contexto) para o agente seguinte, pois o atributo `.output` era acessado antes da execução da `Crew`.

Solução Aplicada:

1. **Refatoração de `equipe_mestre/tarefas.py`:** As funções que criam as tarefas foram modificadas para não aceitarem mais o contexto como um parâmetro de função.
2. **Refatoração de `app.py`:** A dependência entre as tarefas agora é declarada explicitamente usando o atributo `task.context = [previous_task]`.

Resultado: O fluxo de dados entre os agentes foi conceitualmente corrigido, estabelecendo a cadeia de dependências correta para a CrewAI gerenciar.

Fase 3.1: Ajuste Fino da Sintaxe de Contexto (Novo)

Problema Crítico Identificado: Após a refatoração do fluxo, a aplicação gerou um `NameError: name 'context' is not defined`. Isso ocorreu porque as descrições das tarefas em `tarefas.py` foram definidas como f-strings (`f"..."`). O Python tentou formatar a variável `{context}` imediatamente, mas ela só existe para a CrewAI durante a execução, não durante a definição da tarefa.

Solução Aplicada:

1. **Correção em `equipe_mestre/tarefas.py`:** O prefixo `f` foi removido de todas as strings de descrição de tarefas que usavam o placeholder `{context}`. Elas agora são strings regulares (`"..."`), garantindo que o texto `{context}` seja passado literalmente para a CrewAI para que ela possa fazer a substituição no momento correto.

Resultado: O erro `NameError` foi resolvido. A aplicação agora está sintaticamente correta para permitir que o mecanismo de template da CrewAI funcione como esperado.

Próximos Passos (Prioridades Reavaliadas)

1. Execução e Validação Completa:

- **Ação:** Executar a aplicação novamente para confirmar que todo o fluxo, da análise de requisitos à geração e validação do código, ocorre sem erros.
- **Objetivo:** Obter o primeiro resultado de ponta a ponta bem-sucedido no frontend do Streamlit.

2. Implementar Loop de Feedback (Validador -> Implementador):

- **Ação:** Uma vez que a cadeia sequencial esteja 100% funcional, o próximo passo é torná-la iterativa. Isso envolverá explorar funcionalidades como processos não sequenciais ou a criação de um "Agente Gerente".
- **Objetivo:** Permitir que a equipe tente corrigir automaticamente o código que ela mesma gerou, caso a validação falhe.