

# Liderança e Gestão de Equipe

## Planejamento de 15 dias (Sprint)

### Dia 1–2: Kickoff e Onboarding

- Apresentar objetivos do desafio e arquitetura do pipeline (segmentação → keypoints → regressão).
- Revisar o código do *Notebook\_colaboradores* e o *code.ipynb*, identificando gaps.
- Configurar ambiente de trabalho unificado (repositório Git, dependências, datasets).
- Pair programming inicial com Colaborador B para mostrar padrões de código e boas práticas.

### Dia 3–6: Desenvolvimento inicial

- **Líder Técnico (eu):**
  - Implementar/ajustar o módulo crítico (ex.: regressão robusta + integração com métricas de validação).
  - Definir interfaces entre os módulos e padronizar entrada/saída.
- **Colaborador A (júnior experiente):**
  - Revisar segmentação e keypoints, melhorar funções utilitárias.
  - Escrever primeiros testes unitários (pytest).
- **Colaborador B (júnior novo):**
  - Criar scripts auxiliares (pré-processamento de imagens, data loaders simples).
  - Documentar estrutura do projeto (README inicial).

### Dia 7–10: Integração e testes

- Integração dos módulos em um pipeline único (input → output de medidas).

- Criação de *notebook* de experimentação com resultados parciais.
- Revisão cruzada: cada colaborador revisa o Pull Request (PR) do outro.
- Mentoring: sessões rápidas diárias com B para reforçar testes e boas práticas.

#### **Dia 11–13: Validação e refino**

- Avaliação de métricas (MAE, RMSE das medidas previstas).
- Ajuste fino da regressão e checagem de overfitting.
- Colaborador A: expandir testes com casos de borda.
- Colaborador B: produzir documentação de uso + diagramas simples.

#### **Dia 14–15: Consolidação e Risco**

- Preparar apresentação do pipeline end-to-end.
- Mitigar riscos identificados (ex.: falhas de segmentação → fallback de keypoints).
- Revisão final do código, documentação e setup scripts.

#### **Divisão de tarefas**

- **Líder Técnico (eu):**
  - Coordenação geral.
  - Implementação do componente crítico (módulo de regressão + integração final).
  - Revisão de Pull Request (PR) e definição de padrões.
- **Colaborador A (júnior experiente):**
  - Suporte nos módulos principais (segmentação, keypoints).
  - Desenvolvimento de testes automatizados.
  - Validação de resultados e análise de métricas.
- **Colaborador B (júnior novo):**

- Desenvolvimento de utilitários de pré-processamento e scripts auxiliares.
- Documentação técnica e guia de uso.
- Apoio em testes simples, sempre com supervisão próxima.

## **Estratégias de mentoring e onboarding**

- Pair programming inicial com o Colaborador B.
- Code reviews estruturados (eu e A revisam o código de B).
- Check-ins diários de 15 min para alinhar progresso e dúvidas.
- Mini-workshops conduzidos por mim sobre boas práticas (testes, versionamento, modularização).

## **Gerenciamento de cronograma e mitigação de riscos**

- Ferramenta para rastrear tarefas, por exemplo, Trello.
- Mitigação de riscos:
  - Se a segmentação falhar em alguns casos → se a segmentação falhar (ex.: máscara ruim, buracos, bordas mal recortadas), o sistema não para — ele passa a calcular medidas usando apenas os keypoints.
  - Se regressão não atingir precisão → adotar baseline simples (ex.: regressão linear) como backup.
  - Se Colaborador B atrasar entregas → redistribuir parte da carga para A, mantendo B no suporte/documentação.

## Cronograma:

Dia	Atividades Principais	Líder Técnico (eu)	Colaborador A (Júnior experiente)	Colaborador B (Júnior novo)
1–2	Kickoff, onboarding, setup do ambiente, revisão do código	Apresentar arquitetura, revisar código	Revisar notebooks e levantar gaps	Pair programming, setup e README inicial
3–4	Implementação inicial dos módulos	Iniciar módulo crítico (regressão)	Suporte em segmentação/keypoints	Utilitários de pré-processamento
5–6	Consolidação inicial	Definir interfaces dos módulos	Escrever primeiros testes unitários (pytest)	Documentação de funções
7–8	Integração parcial dos módulos	Integrar pipeline inicial	Revisar código do B e ampliar testes	Scripts auxiliares + documentação
9–10	Revisão cruzada, primeiros resultados em notebook de experimento	Refinar módulo crítico	Validação preliminar das métricas	Ajustes em utilitários e documentação
11	Avaliação de métricas (MAE, RMSE)	Refino da regressão	Expandir testes com casos de borda	Documentar resultados parciais
12–13	Ajustes finos, checagem de overfitting, validação com baseline	Revisar integração e precisão	Análises comparativas de métricas	Diagramas e guia de uso
14	Consolidação do pipeline end-to-end	Revisão final + integração completa	Testes finais automatizados	Ajustes finais de documentação
15	Preparação da apresentação e mitigação de riscos	Preparar apresentação e plano de riscos	Apoio na demonstração e consolidação de métricas	Guia final de usuário e setup script