**Estimativa por LOC**

Linguagem de programação: JAVA

Complexidade: Alta/Média/Baixa

Produtividade: 40 linhas de código por hora

| **Funcionalidade** | **Complexidade** | **LOC Otimista** | **LOC Provável** | **LOC**  **Pessimista** | **LOC**  **Estimado** |
| --- | --- | --- | --- | --- | --- |
| Criação de usuário | Média | 100 | 150 | 230 | 160 |
| Criação de currículo | Média/Alta | 150 | 300 | 450 | 300 |
| Upload de documentos | Média | 50 | 100 | 180 | 110 |
| Compartilhamento de currículos | Baixa/Média | 70 | 100 | 160 | 110 |
| Exportação de currículo | Alta | 120 | 180 | 300 | 200 |
| **Total** | - | 490 | 830 | 1320 | 880 |

**Estimativa por Pontos por Função**

| **Parâmetro** | **Contagem** | **Fator de Ponderação** | **Resultado** |
| --- | --- | --- | --- |
| nº de entradas do usuário | 15 | Média  (4) | 60 |
| nº de saídas do usuário | 2 | Média  (5) | 10 |
| nº de consultas do usuário | 3 | Simples  (3) | 9 |
| nº de arquivos | 1 | Média  (10) | 10 |
| nº de interfaces internas | 2 | Simples  (5) | 10 |
| **Contagem Total:** | - | - | 99 |





Classificação: Médio/Alto

**Estimativa por modelo COCOMO**

= 2.4

= 1.05

= 2.5

= 2.38

= Esforço

= Tempo de Desenvolvimento

= 0.88 (880 linhas de código estimadas / 1000)

**Métricas**

Esforço Estimado (Horas de Trabalho)

E=2.01 (esforço em meses-persona)

T=19.5 horas estimadas para desenvolvimento.

Essa métrica representa o esforço estimado com base no número de linhas de código.

Produtividade (LOC por Hora)

Com uma produtividade de 40 LOC/hora para uma equipe utilizando Java, as funções do sistema estão distribuídas conforme as LOC:

Criação de usuário: 160 LOC

Criação de currículo: 300 LOC

Upload de documentos: 110 LOC

Compartilhamento de currículos: 110 LOC

Exportação de currículo: 200 LOC

Produtividade esperada: 880LOC / 40 LOC/H = 22Horas

Cronograma (Tempo Total e Duração das Atividades)

O projeto envolve a criação de diversas funcionalidades, como sistema de usuários, currículos, e upload de documentos, que serão implementados em fases interdependentes conforme a rede de tarefas.

O diagrama de Gantt sugere uma sequência de atividades que pode ser seguida para a entrega gradual das funcionalidades, com marcos de controle de qualidade (SQA) ao longo do projeto.

O cronograma, portanto, precisa considerar o tempo para completar cada fase do projeto e os tempos de teste e controle.

Estimativas Baseadas em Pontos por Função

A contagem total de pontos é 99, categorizado como médio/alto. Isso pode ser utilizado para determinar a complexidade e esforço necessário para atender às funcionalidades do sistema.

Métricas de Qualidade

Tempo de resposta: O sistema deve ter um tempo de resposta de menos de 3 segundos em 95% das requisições (RQ\_02.1).

Disponibilidade: O sistema deve ter uma disponibilidade de 99,9% (RQ\_05.1).

Escalabilidade: O sistema deve suportar até 100.000 usuários simultâneos mantendo desempenho adequado (RQ\_03.1).

Métricas de Controle de Qualidade (SQA)

No gráfico de Gantt, a fase de SQA (Controle de Qualidade) está presente em múltiplos pontos, o que sugere a aplicação de métricas de teste como:

- Cobertura de Testes: Medida pela porcentagem de funcionalidades testadas.

- Taxa de Erro: Definida como o número de erros encontrados por hora de teste.

Métricas de Desempenho

- Taxa de Crescimento de Tarefas: Medir quantas tarefas ou funcionalidades são concluídas por semana.

- Desvios no Cronograma: Avaliar os desvios entre o cronograma estimado e o real.