

Ciclo de vida clássico (Cascata)

Ciclo de vida em que cada etapa é focada uma única vez.

1. Comunicação

Um brainstorm foi feito onde definiu-se em linhas gerais o que seria o jogo.

- 1.1 **Recurso humano:** Equipe do projeto;
- 1.2 **Recurso de hardware:** Nenhum;
- 1.3 **Recurso de software:** Nenhum;
- 1.4 **Documentos:** Nenhum;

2. Planejamento

Baseado nas linhas gerais definimos mais detalhes sobre o jogo, plataforma de desenvolvimento e as funcionalidades do jogo além de acrescentar os paradigmas de desenvolvimento e métrica de software.

- 2.1 **Recurso humano:** Equipe do projeto;
- 2.2 **Recurso de hardware:** Notebook de todos;
- 2.3 **Recurso de software:** Google docs, bloco de notas, github e project libre;
- 2.4 **Documentos:** High Concept Design;

3. Modelagem

Fizemos os requisitos funcionais e não funcionais e o diagrama de classes.

- 3.1 **Recurso humano:** Equipe do projeto;
- 3.2 **Recurso de hardware:** Notebook do Marim;
- 3.3 **Recurso de software:** ReqView para modelar os requisitos e StarUML para modelar as classes;
- 3.4 **Documentos:** Levantamento de requisitos e digrama de classes;

4. Construção

Iremos dividir o grupo e começar a desenvolver na IDE do RPG Maker VX Ace.

- 4.1 **Recurso humano:** Equipe do projeto;
- 4.2 **Recurso de hardware:** Notebook de todos;
- 4.3 **Recurso de software:** RPG Maker VX Ace;
- 4.4 **Documentos:** Documentação do projeto;

5. Implantação

Depois de desenvolver os scripts, juntamos todos no rpg maker e testamos o projeto final.

- 5.1 **Recurso humano:** Quase toda a equipe do projeto;
- 5.2 **Recurso de hardware:** Notebook de todos;
- 5.3 **Recurso de software:** RPG Maker VX Ace;
- 5.4 **Documentos:** Artigo do projeto;

6. Motivação do paradigma utilizado

Pelo tempo curto não temos como retomarmos tarefas, então prezando pelo final do projeto utilizaremos o ciclo de vida em cascata em que cada etapa é focada em uma única vez.

7. Métrica de software utilizada


















- Número de scripts desenvolvidos;
- Número de funcionalidades abandonadas;
- Número de funcionalidades acrescentadas após o planejamento;
- Tempo de gameplay para testar todas as funcionalidades atualmente disponíveis no jogo;

7.1 Motivação da métrica

- Verifica a complexidade do projeto;
- Verifica a viabilidade inicial do projeto;
- Verifica o quão bem foi modelado o projeto;
- Verifica o avanço real do projeto;

8. Cronograma do projeto

O cronograma do projeto foi elaborado utilizando a ferramenta “Project Libre”.

	Name	Duration	Start	Finish
	[-] Criação do Jogo Digital	76.125 days	3/22/18 4:00 PM	7/6/18 5:00 PM
	 (Parte 2) Criação do repositório no gitHu	17 days	3/22/18 4:00 PM	4/16/18 4:00 PM
	 (Parte 3) Levantamento de requisitos e c	7 days	4/13/18 8:00 AM	4/23/18 5:00 PM
	[-] Programação	44.125 days	4/30/18 8:00 AM	6/29/18 9:00 AM
	Prototipagem	9 days	4/30/18 8:00 AM	5/10/18 5:00 PM
	Level Design	21 days	5/12/18 8:00 AM	6/11/18 5:00 PM
	Art/Audio	11 days	5/20/18 8:00 AM	6/4/18 5:00 PM
	Alfa Test	8 days	6/1/18 8:00 AM	6/12/18 5:00 PM
	Beta Test	7 days	6/12/18 8:00 AM	6/20/18 5:00 PM
	Quality Assurance	6 days	6/21/18 9:00 AM	6/29/18 9:00 AM
	 (Parte 4) Estrutura Analítica	8 days	4/23/18 8:00 AM	5/2/18 5:00 PM
	 (Parte 5) Implementação do protótipo - A	9 days	4/30/18 8:00 AM	5/10/18 5:00 PM
	Escrever artigo sobre o jogo	49 days	5/1/18 8:00 AM	7/6/18 5:00 PM