



DISCENTES:

Ana Beatriz Almeida da Silva

Hillary Diniz Saldanha

Letícia Maria Gonçalves de Moraes

Rubens Alexandre de Sousa Ferreira

**Documento de Visão
Nome do projeto: BitStart**

Versão 1.0



Histórico de Revisões

Data	Versão	Descrição	Autor
06/10/2025	1.0	Versão inicial	Hillary e Alexandre

Equipe: Hillary, Alexandre, Ana Beatriz e Letícia Morais.

Sumário

1. INTRODUÇÃO.....	4
2. IDENTIFICAÇÃO DO PROJETO.....	4
3. METODOLOGIA DEFINIDA.....	4
4. CRONOGRAMA DO PROJETO.....	6
5. STACK DE DESENVOLVIMENTO.....	8
6. CONCLUSÃO.....	8
7. DOCUMENTAÇÃO.....	9

Documento de Visão

1. Introdução

O BitStart é um jogo educacional voltado ao ensino introdutório de algoritmos e lógica de programação para alunos ingressantes no curso de BTI - UFERSA. Com uma identidade visual inspirada em jogos pixelados 8-bit, o aplicativo busca tornar o aprendizado de conceitos básicos de lógica e programação mais interativo, acessível e lúdico. O jogador entra em um mundo digital corrompido, onde os algoritmos deixaram de funcionar corretamente, onde sua missão é restaurar a lógica resolvendo desafios que envolvem raciocínio, ordem, repetição e decisão — os fundamentos da programação. Cada “fase” é um nó de rede ou setor do sistema, e cada setor representa um conceito lógico (não dito como “lição”, mas como obstáculo narrativo). Assim, o aprendizado acontece indiretamente, por meio de puzzles e mecânicas.

2. Identificação do Projeto

Projeto	Bit Start
Requisitante	Náthalee Cavalcanti / UFERSA George Vieira/ UFERSA
Gerente de Projetos	Hillary Diniz Saldanha
Equipe de desenvolvimento	Ana Beatriz Almeida da Silva - 2023011203 Hillary Diniz Saldanha - 2023011160 Letícia Maria Gonçalves de Moraes - 2023011312 Rubens Alexandre de Sousa Ferreira - 2023011112

3. Metodologia Definida

A metodologia definida e o modelo escolhido foi o Modelo Incremental com práticas do Scrum

3.1. Justificativa da metodologia

O desenvolvimento do BitStart será dividido em incrementos jogáveis, permitindo testar e melhorar o jogo progressivamente. Como o projeto possui prazos curtos (outubro a dezembro), o modelo incremental garante entregas frequentes, enquanto o Scrum traz organização e acompanhamento das sprints semanais (reuniões curtas, backlog e revisão).

3.2. Vantagens da metodologia

- Feedback contínuo e adaptação de requisitos.
- Teste de jogabilidade desde as primeiras entregas.
- Melhoria incremental dos módulos (engine, blocos, interface, mapa, etc.).

4. Cronograma do projeto

4.1. Etapas previstas e entregas parciais

Etapa	Período	Entregáveis
Planejamento e Design	06/10 – 12/10	Documento técnico, logo e layout base.
Protótipo Inicial (Engine e Bot)	13/10 – 26/10	Estrutura de grid, movimentação e interpretação básica de comandos.
Interface de Blocos (Lógica Visual)	27/10 – 09/11	Protótipo com interface visual de blocos lógicos.
Documentação Final (v1.0)	10/11 – 15/11	Documento completo de requisitos e GDD.
Mundo e Desafios Avançados	16/11 – 30/11	Novos mapas e puzzles lógicos.
Testes e Empacotamento Final	01/12 – 16/12	Versão executável (.exe/.zip) e relatório final.

4.2. Papéis da Equipe

Função	Responsável	Atividades
Scrum Master / Documentação	Hillary Diniz Saldanha	Organização do cronograma e reuniões de acompanhamento
Desenvolvedor 1	Letícia Maria Gonçalves de Moraes	Implementação da lógica dos blocos, parser e execução
Desenvolvedor 2	Ana Beatriz Almeida da Silva	Interface do jogo, HUD, menus e interações visuais.
Designer e arte	Rubens Alexandre de Sousa Ferreira	Criação dos sprites, mapa, ícones e animações.
QA	Rubens Alexandre de Sousa Ferreira	Testes e controle de qualidade

4.3. Cronograma de Projeto

Marco/Etapa	Início	Fim	Responsável
Kickoff e definição de requisitos	06/10	12/10	Equipe toda
Protótipo base (grid + bot)	13/10	26/10	Devs
Interface de blocos	27/10	12/11	Devs
Documentação final (v1.0)	13/11	16/11	QA / Documentação
Adição de desafios e fases	20/11	06/11	Equipe toda
Empacotamento e testes	06/12	16/12	QA + Devs

O acompanhamento será feito via GitHub Projects, com colunas: Backlog, Em Desenvolvimento, Em Teste, Concluído.

5. Stack de Desenvolvimento

5.1. Frontend

- Biblioteca: Pygame
- Linguagem: Python
- Função: Renderização gráfica, interface, intervenção e eventos de teclado

5.2. Backend / Engine

- Linguagem: Python
- Módulos Principais: engine.blocks, engine.interpreter, game.level, game.actor
- Função: Gerenciar comandos, interpretar sequências e atualizar o estado do jogo.

5.3. Banco de Dados

- PostgreSQL: Armazena progresso, pontuação e fases desbloqueadas.
- JSON: Formato principal para leitura de fases e parâmetros.

5.4. Versionamento

- Git + GitHub: Controle de versão, branches e commits organizados.
 - Branch principal: main
 - Branch de features: feature/interface, feature/interpreter, etc.

6. Conclusão

O Documento de Planejamento do BitStart consolidou as bases necessárias para o desenvolvimento do jogo, alinhando escopo, metodologia, cronograma e stack tecnológica. A definição de um modelo incremental com práticas ágeis do Scrum mostrou-se adequada ao contexto do projeto, permitindo entregas parciais e contínuas, essenciais para um ciclo de desenvolvimento curto e focado em resultados.

A maior dificuldade enfrentada pelo grupo durante a construção deste documento foi equilibrar a divisão de tarefas entre os integrantes, considerando a multidisciplinaridade da equipe, e estabelecer prazos realistas para cada etapa, especialmente devido à natureza acadêmica do projeto e aos prazos apertados. No entanto, superamos esse desafio por meio de uma comunicação constante e da adoção de ferramentas de organização, como o GitHub e reuniões semanais de acompanhamento.

Com esse planejamento estruturado, a equipe sente-se preparada para avançar nas fases de implementação, testes e entrega final, sempre com o objetivo de oferecer uma ferramenta educacional inovadora e alinhada às necessidades dos alunos de BTI da UFERSA.

7. Documentação