

# Relatório de Desenvolvimento de Projeto

**Douglas Wink**

Matrícula: 202512320

Curso: Engenharia da Computação

**Marcus Silveira**

Matrícula: 202511463

Curso: Engenharia da Computação

Santa Maria, Rio Grande do Sul, Brasil  
11 de Julho de 2025

## Informações Gerais do Projeto

- **Nome do Projeto:** Labirinto Sombrio
- **Instituição:** Universidade Federal de Santa Maria (UFSM)
- **Curso:** Engenharia da Computação
- **Disciplina:** Algoritmo e Programação
- **Professor(a):** Dr. Marcelo Zanetti
- **Data de Entrega:** 11 de Julho de 2025

## Descrição do Projeto

O "Labirinto Sombrio" é um jogo 2D de horror e exploração desenvolvido utilizando a biblioteca Allegro 5 em linguagem C. O jogador assume o papel de um protagonista que precisa entrar em uma casa assombrada, cheia de fantasmas, para desvendar o mistério por trás da morte de um amigo próximo.

## Mecânicas Principais

- **Exploração do Labirinto:** O jogador navega por um labirinto escuro, utilizando uma lanterna.
- **Enfrentamento de Fantasmas:** Fantasmas patrulham o labirinto e perseguem o jogador ao se aproximarem. A lanterna pode ser usada para dissipar os fantasmas após um tempo de exposição.
- **Coleta de Itens:** Para progredir, o jogador deve coletar 5 itens essenciais, que aparecem sequencialmente em locais aleatórios do mapa.
- **Área de Fuga:** Após coletar todos os itens, uma área de fuga é ativada no ponto inicial do mapa, onde o jogador deve chegar para vencer o jogo.
- **Condições de Fim de Jogo:** O jogo termina se o jogador colidir com um fantasma (Game Over) ou se alcançar a área de fuga após coletar todos os itens (Vitória).
- **Interface:** Inclui uma tela de menu inicial com tutorial e opções de reiniciar após o fim do jogo, além de um HUD que exibe o tempo de jogo e a quantidade de itens coletados.

## Tecnologias Utilizadas

- **Linguagem de Programação:** C
- **Biblioteca Gráfica e Multimídia:** Allegro 5

- **Ambiente de Desenvolvimento Integrado (IDE):** Code::Blocks
- **Compilador:** GNU GCC Compiler (MinGW)

## Dificuldades Enfrentadas e Soluções

Durante o desenvolvimento do projeto, a equipe se deparou com alguns desafios técnicos que exigiram estudo e persistência para serem superados:

- **Geração e Desenho do Mapa:** A criação de um labirinto complexo e a sua representação visual eficiente foram um desafio inicial. A geração e o desenho do mapa de forma que permitisse a colisão precisa e a exploração fluida se mostraram demorados. A dificuldade foi superada com a implementação de um sistema de matriz de caracteres, onde cada caractere representa um tipo de tile (parede, chão, ponto de início), otimizando tanto o armazenamento do layout do mapa quanto o processo de renderização e detecção de colisões.
- **Comportamento da Lanterna e Colisão com Paredes:** A implementação da lanterna inicialmente apresentava o problema de que seus feixes visuais e sua lógica de iluminação não interagiam corretamente com as paredes do labirinto, atravessando-as. Esta dificuldade foi superada por Marcus Silveira, que implementou um algoritmo de *raycasting* para detectar a colisão dos feixes de luz com as paredes, garantindo que a iluminação se comportasse de forma realista, sendo bloqueada pelos obstáculos do mapa.
- **Carregamento e Reprodução de Áudio:** A integração de áudio (música de fundo e efeitos sonoros) representou um desafio inesperado. Embora o código de reprodução e os arquivos de áudio estivessem corretos, a música não era carregada nem reproduzida. A dificuldade residia primariamente no **\*\*caminho de arquivo relativo e na localização das dependências\*\***. A solução para este problema foi simples, porém crucial: garantir que os arquivos de áudio no formato OGG (como `piano-terror.ogg` e `ghost-scream.ogg`) fossem **\*\*posicionados diretamente na pasta `debug` do executável do projeto\*\*** (`bin/Debug/`), em vez de estarem em uma subpasta `audio` ou em outro local. Isso permitiu que a biblioteca Allegro localizasse e carregasse os arquivos de áudio com sucesso, habilitando a funcionalidade de som no jogo.
- **Gerenciamento de Sprites e Redimensionamento de Pixel Art:** A transição de primitivas gráficas para sprites pixel art exigiu atenção ao redimensionamento das imagens. Observou-se que o redimensionamento padrão por software resultava em sprites borradas, descaracterizando o estilo pixel art. A solução encontrada foi a utilização de ferramentas de edição de imagem com o método de interpolação “Nearest Neighbor” para manter a nitidez e o estilo original das sprites ao serem ajustadas ao tamanho dos blocos do jogo.

## Observações Adicionais para a Demo

- **Assets Visuais:** As sprites de personagem e inimigos foram criadas em estilo pixel art, e os tiles de chão e parede foram ajustados para o tamanho do bloco do jogo.
- **Áudio:** A música de fundo e efeitos sonoros agora estão totalmente funcionais, contribuindo significativamente para a imersão na atmosfera de horror e suspense do jogo.

## Agradecimentos

O grupo gostou muito de realizar essa atividade, pois mostrou uma necessidade de programação junto com desafios que deveriam ser enfrentados, e através do esforço do grupo, conseguimos realizar essa atividade resultando em um jogo que gostamos de jogar, e como pensávamos em fazer