

## ■ ***Etapa 1 – Fundamentos (Setup do projeto)***

- Criar o arquivo main.c e incluir bibliotecas necessárias (stdio.h, stdlib.h, string.h, time.h).
- Definir as structs principais:
- Personagem
- Inimigo
- Item
- Criar função para inicializar personagem (nome, força inicial, velocidade = 1).
- Testar no main imprimindo os dados do personagem.

## ■ ***Etapa 2 – Mapa***

- Criar função para alocar dinamicamente uma matriz (malloc).
- Perguntar ao usuário altura e largura do mapa.
- Calcular  $X = (\text{altura} \times \text{largura})/4$ .
- Criar funções para posicionar:
- personagem (posição inicial aleatória)
- inimigos (X inimigos, força aleatória)
- itens (X itens, tipo = força ou velocidade)
- Criar função para exibir o mapa.
- Primeiro mostrar apenas o personagem.
- Depois mostrar inimigos e itens para depuração.

## ■ ***Etapa 3 – Exploração***

- Criar função para movimentar personagem (cima, baixo, esquerda, direita).
- Implementar lógica de velocidade (andar de 1 em 1, ou 2 em 2 casas se velocidade = 2).
- Atualizar posição no mapa a cada movimento.
- Testar movimentando o personagem pelo mapa e imprimindo o resultado.

## ■ ***Etapa 4 – Interações***

- Criar função para verificar se o personagem encontrou:
- inimigo → chama função de combate

- item → chama função de coleta
- Criar função de coleta de item:
- força → soma na força
- velocidade → aumenta a velocidade
- Criar função de combate:
- comparar forças
- vencedor ganha a força do perdedor
- atualizar struct personagem

## ■ ***Etapas 5 – Status do Jogo***

- Criar função para imprimir status do personagem (nome, força, velocidade).
- Criar função para imprimir resumo do jogo (inimigos restantes, itens restantes, posição no mapa).
- Testar rodando alguns turnos de movimento + interações.

## ■ ***Etapas 6 – Persistência***

- Criar função para salvar jogo em arquivo (save.txt):
- dados do personagem
- dados dos inimigos restantes
- dados dos itens restantes
- posição atual no mapa
- Criar função para carregar jogo do arquivo.
- Criar menu inicial:
- 1. Novo jogo
- 2. Carregar jogo
- Testar salvar, fechar o programa, abrir e carregar o estado anterior.

## ■ ***Etapas 7 – Modularização***

- Separar o código em múltiplos arquivos:
- main.c → apenas loop principal
- personagem.c/h → criação, status, combate
- mapa.c/h → alocação, exibição, movimentação
- jogo.c/h → salvar/carregar, menu principal

- Ajustar ponteiros e referências para manter o código organizado.

## ■ ***Etapas 8 – Finalização para Entrega***

- Revisar se todas as funcionalidades pedidas no PDF foram implementadas.
- Inserir comentários e nomes significativos em todas as funções/variáveis.
- Testar cada passo da sequência de apresentação (mapa vazio, criação de personagem, coleta de item, combate, salvar, carregar, etc.).
- Gravar o vídeo demonstrando todas as etapas pedidas.
- Compactar tudo em .zip com:
  - Código fonte
  - Nomes da equipe
  - Link do vídeo no YouTube