Planejamento de Testes

Projeto Sudoku

Engenharia de Software IV - ES4A4

Any Gomes - SP3050122

Gabriel Pinheiro Brants Gonçalves - SP3013456

Matheus Casagrande - SP3013944

2021

ÍNDICE

1. CASOS DE TESTE 4
2. TIPOS DE TESTE 10
3. RECURSOS 12
4. CRONOGRAMA 13
5. REFERÊNCIAS 14

INTRODUÇÃO

Este documento descreve os requisitos do projeto do Sudoku a testar, os tipos de testes definidos para cada um deles, os recursos de hardware e software a serem empregados e o cronograma dos testes ao longo do projeto. As seções referentes aos requisitos, recursos e cronograma servem para acompanhar a evolução dos testes do projeto.

1. CASOS DE TESTE

a. Histórias de usuário de teste

- Como jogador de sudoku, eu quero um sistema de ranqueamento local, para que eu possa saber o quão bem eu estou jogando.
- Como jogador de sudoku, eu quero ter a opção de resolver automaticamente um jogo que eu tenha iniciado, para saber como se resolve um jogo caso eu não consiga.

b. Casos de teste

- **Caso de teste 1**: sistema de ranqueamento

Descrição: o programa deve ler um arquivo de texto onde os recordes dos jogadores são salvos.

Ator: jogador.

Pré-condições: detectar o arquivo de texto.

Fluxo principal: ao fim da partida, o programa pede o nome do jogador para gravar seu recorde em um arquivo de texto.

1. No início e fim de jogo, o programa lê o horário da máquina onde está sendo executado e esse intervalo é salvo com o nome do jogador no documento de texto do ranqueamento.
2. O jogador clica no botão “Ranking” para consultar sua classificação.
3. O programa deve detectar o arquivo de texto com os dados de jogo.
4. O programa exibe os 5 melhores resultados.

Fluxo alternativo: o arquivo de texto já contém 5 posições e o jogador não consegue pontuação suficiente para entrar no ranking.

1. Ao final do jogo, a pontuação e nome do jogador são gravados no arquivo de texto.

O programa deve organizar todas as posições de acordo com a pontuação.

2. A pior pontuação é apagada do arquivo e não é exibida no ranking.

Exceção: o arquivo de texto não é detectado.

1. Ao clicar no botão de “Ranking”, o programa não detecta o arquivo de texto.
2. O programa retorna “false”.

- **Caso de teste 2**: verificação de lógica de criação de uma linha do ranking.

Descrição: o programa irá verificar se o arquivo “ranking.txt” foi criado com a lógica correta.

Ator: sistema.

Pré-condições: detectar o arquivo de texto.

Fluxo principal: verifica a lógica de criação de uma linha no ranking.

1. Caso o programa detecte o arquivo do ranking, ele retorna “true”.
2. Ele divide a lista neste arquivo após cada quebra de linha.
3. O programa verifica se cada linha cumpre tais requisitos: posição, tempo de conclusão e nome.
4. Para tal, ele irá conferir se cada linha tem: 1 elemento inteiro, 1 elemento float e 1 elemento string.

Exceção: arquivo “ranking.txt” não encontrado.

1. Caso o programa não detecte o arquivo ranking.txt, o programa retorna “false”.

- **Caso de teste 3**: posição do placar.

Descrição: testa se o placar está organizando corretamente o ranking.

Ator: programa.

Pré-condições: que o arquivo ranking.txt esteja formatado corretamente.

Fluxo principal: o placar deve organizar o ranking do menor para o maior tempo de resolução do jogo.

1. Popula um array com os números das posições do ranking.
2. Popula um array com os números dos tempos do ranking.
3. Cria um array para posições e tempo e arruma.
4. Confere se o array arrumado é igual a sua versão anterior.
5. Caso sim, confere se as posições estão em ordem crescente.

Exceção: arquivo ranking.txt inválido.

1. O algoritmo lê o arquivo ranking.txt.
2. Caso não esteja formatado corretamente, retorna false.

- **Caso de teste 4**: conferir se a matriz sudoku a ser resolvida é válida.

Descrição: testa se a matriz que está em sudoku.txt é válida como um jogo de sudoku.

Ator: sistema.

Pré-condições: o programa deve o arquivo “sudoku.txt”.

Fluxo principal: confere a formatação da matriz no arquivo de texto.

1. O programa lê o arquivo sudoku.txt.
2. Divide a string de leitura do tabuleiro em uma lista de strings.
3. Quebra a string em uma lista.
4. Adiciona a linha formatada em uma matriz formatada.
5. Converte cada elemento da matriz formatada em inteiro.

Exceção: matriz não válida como um jogo de sudoku.

1. O programa lê o arquivo sudoku.txt.
2. Não encontra uma string no arquivo
3. Retorna “false”.

- **Caso de teste 5**: conferir se o jogo resolvido pelo algoritmo foi resolvido corretamente

Descrição: ao clica no botão “solve”, o programa resolve um jogo iniciado. Este teste visa conferir se o algoritmo responsável pela resolução do jogo está cumprindo seu papel corretamente.

Ator: sistema.

Pré-condições: deve receber um jogo válido, verificado pelo teste anterior.

Fluxo principal: confere se todas as posições do jogo resolvido estão corretas

1. Recebe e guarda uma posição analisada.
2. Confere a linha toda.
3. Confere a coluna toda.
4. Confere um quadrado 3x3.
5. Se passar por estes testes e não retorna false, retorna true.

Exceção: recebe uma posição incorreta.

1. Recebe e guarda uma posição analisada.
2. Confere a linha toda.
3. Confere a coluna toda.
4. Confere um quadrado 3x3.
5. Se receber false de algum dos testes, o algoritmo retorna false.

2. TIPOS DE TESTE

a. Caso de teste 1 - sistema de ranqueamento

Objetivo:	O programa deve ler um arquivo de texto onde os recordes dos jogadores são salvos.
Técnica:	☒ Manual ☐ Automática
Estágio do teste: ☒ Unidade ☐ Aceitação	**Abordagem do teste:** ☒ Caixa branca ☐ Caixa preta
Responsável:	*Gabriel*

b. Caso de teste 2 - verificação de lógica de criação de uma linha do ranking.

Objetivo:	o programa irá verificar se o arquivo “ranking.txt” foi criado com a lógica correta.
Técnica:	☒ Manual ☐ Automática
Estágio do teste: ☒ Unidade ☐ Aceitação	**Abordagem do teste:** ☒ Caixa branca ☐ Caixa preta
Responsável:	*Gabriel*

c. Caso de teste 3 - posição do placar.

Objetivo:	testa se o placar está organizando corretamente o ranking.
Técnica:	☒ Manual ☐ Automática
Estágio do teste: ☒ Unidade ☐ Aceitação	**Abordagem do teste:** ☒ Caixa branca ☐ Caixa preta
Responsável:	*Matheus*

d. **Caso de teste 4 -** conferir se a matriz sudoku a ser resolvida é válida.

Objetivo:	*testa se a matriz que está em sudoku.txt é válida como um jogo de sudoku.*
Técnica:	☒ Manual ☐ Automática
Estágio do teste: ☒ Unidade ☐ Aceitação	**Abordagem do teste:** ☒ Caixa branca ☐ Caixa preta
Responsável:	*Any*

e. **Caso de teste 5 -** conferir se o jogo resolvido pelo algoritmo foi resolvido corretamente.

Objetivo:	*ao clica no botão "solve", o programa resolve um jogo iniciado. Este teste visa conferir se o algoritmo responsável pela resolução do jogo está cumprindo seu papel corretamente.*
Técnica:	☒ Manual ☐ Automática
Estágio do teste: ☒ Unidade ☐ Aceitação	**Abordagem do teste:** ☒ Caixa branca ☐ Caixa preta
Responsável:	*Any*

f. **Caso de teste 6 –** teste de aceitação

Objetivo:	*Nesta iteração será testado se o programa está aceito para produção.*
Técnica:	☒ Manual ☐ Automática
Estágio do teste: ☐ Unidade ☒ Aceitação	**Abordagem do teste:** ☐ Caixa branca ☒ Caixa preta
Responsável:	*Matheus*

3. RECURSOS

De extrema importância para o bom andamento dos testes, os recursos a serem utilizados durante os testes são descritos nessa seção. Os recursos estão divididos nas subseções que se seguem.

a. Ambiente de Teste – Software & Hardware

O hardware utilizado para testes foi um computador com processador Ryzen 3 2200g, com 16gb de RAM, 120gb de SSD para armazenamento.

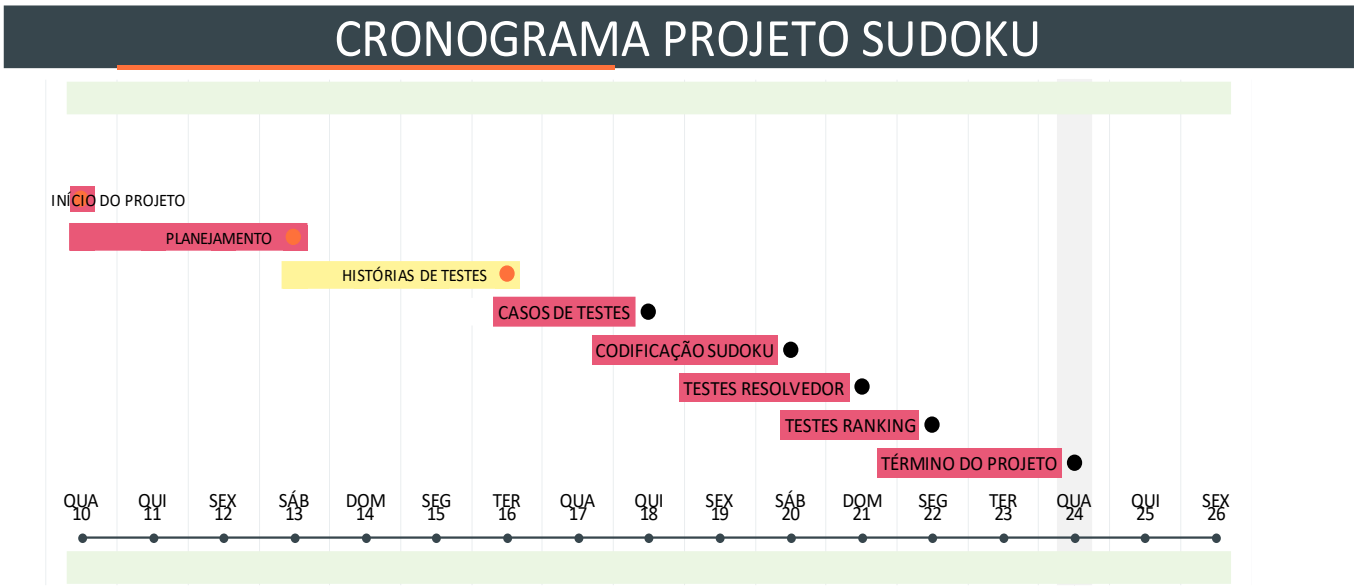
O sistema operacional utilizado foi o Windows 10 Pro. A IDE utilizada foi o Visual Studio Code.

b. Ferramentas de Teste

As ferramentas usadas para teste foi o Visual Studio Code, com PyGame.

4. CRONOGRAMA

O cronograma refere-se as fases do projeto de desenvolvimento e testes unitários do projeto do Sudoku.



ATIVIDADE	INÍCIO	TÉRMINO
Início do projeto	10/03/2021	
Planejamento	10/03/2021	13/03/2021
Histórias de Testes	13/03/2021	16/03/2021
Casos de Testes	15/03/2021	18/03/2021
Codificação Sudoku	16/03/2021	20/03/2021
Testes Resolvedor	20/03/2021	21/03/2021
Testes Ranking	20/03/2021	22/03/2021
Término do projeto	24/03/2021	

5. REFERÊNCIAS

Abordagens de Testes. URL: <https://anielacole.wordpress.com/2010/08/16/abordagens-de-testes/>.

Plano de Teste. URL: <https://www.devmedia.com.br/plano-de-teste-um-mapa-essencial-para-teste-de-software/13824>.