

UNICURITIBA  
Ciência da Computação

Gabrielle Luisa Horiuchi  
Gustavo de Freitas Crepaldi

## **RELATÓRIO DA AVALIAÇÃO A3**

Curitiba – PR  
2025

Gabrielle Luisa Horiuchi - 172416615  
Gustavo de Freitas Crepaldi - 172415709

## **RELATÓRIO DA AVALIAÇÃO A3**

Relatório sobre o trabalho Acadêmico apresentado à Universidade UNICURITIBA como requisito parcial para a avaliação da disciplina Usabilidade, desenvolvimento web, mobile e jogos, do curso de Ciência da Computação.

DOCENTE: Prof. Flavio Henrique da Silva.

## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>4</b>
<b>2 DESENVOLVIMENTO .....</b>	<b>5</b>
2.1 Divisão das Tarefas .....	5
2.2 Estrutura do Projeto .....	6
2.3 Explicação da Aplicação/Software .....	7
2.4 Orientações de execução da Aplicação/Software .....	7
2.5 Repositório .....	8
<b>3 CONCLUSÃO .....</b>	<b>9</b>
<b>Referências .....</b>	<b>10</b>

## 1 INTRODUÇÃO

Este relatório tem como objetivo apresentar como foi o processo de desenvolvimento de um jogo de Damas utilizando a linguagem de programação Python e a biblioteca Pygame, com base no conteúdo passado em sala de aula da disciplina Usabilidade, Desenvolvimento Web e Jogos.

O problema central deste trabalho consiste em desenvolver uma aplicação funcional do jogo de Damas, conseguindo exercer suas regras com fidelidade e proporcionar uma interação intuitiva entre dois jogadores. Isso envolve desafios como representar o tabuleiro e as peças, controlar os movimentos de acordo com as regras oficiais, lidar com a lógica de captura e promoção das peças, além de garantir uma interface clara e responsiva.

## 2 DESENVOLVIMENTO

O projeto foi desenvolvido com o objetivo de criar um jogo de damas funcional, aplicando os conhecimentos adquiridos ao longo da disciplina de Usabilidade, Desenvolvimento Web e Jogos. A aplicação foi construída utilizando a linguagem de programação Python, com o apoio da biblioteca Pygame, sendo organizada de forma modular a fim de garantir maior legibilidade e facilitar futuras manutenções.

O processo de desenvolvimento envolveu etapas como a implementação da lógica do jogo, a definição da interface gráfica, o controle de eventos e a estruturação dos objetos responsáveis pelo funcionamento do sistema. O código-fonte foi distribuído em arquivos distintos, cada um com responsabilidades bem definidas, visando a separação adequada entre a lógica de jogo, a interface e os objetos do tabuleiro.

O conteúdo deste relatório abrange desde a divisão de tarefas entre os integrantes do grupo até a explicação da estrutura interna do projeto, incluindo as bibliotecas utilizadas, os arquivos do código-fonte e as classes criadas. Também são apresentadas orientações para uso da aplicação, instruções de instalação e execução, além do link para acesso ao repositório completo com todos os materiais desenvolvidos.

### 2.1 DIVISÃO DAS TAREFAS

O desenvolvimento do projeto foi realizado em grupo, com a divisão das tarefas de forma a otimizar o tempo e aproveitar as habilidades individuais de cada integrante. As atividades foram distribuídas conforme descrito a seguir:

Gabrielle desenvolveu a interface gráfica, assim, ficando responsável pela classe Interface além de desenvolver todas as imagens usadas na aplicação do jogo.

Gustavo ficou responsável pela lógica do jogo, incluindo as regras de movimentação, captura e promoção das peças, desenvolveu os objetos do jogo, além disso desenvolveu o relatório técnico como requisitado na avaliação.

Além disso, Gabrielle e Gustavo colaboraram na construção do arquivo main, responsável por iniciar o jogo e integrar as funcionalidades das demais classes.

## 2.2 ESTRUTURA DO PROJETO

O projeto foi desenvolvido utilizando a biblioteca Pygame, responsável por fornecer os recursos necessários para a criação de jogos em Python. Além disso, foram utilizadas outras bibliotecas auxiliares, como o PyInstaller e PythonTurtle, que permitiu a criação do arquivo executável do jogo. O código foi organizado em cinco arquivos principais e uma pasta, seguindo boas práticas de programação: main.py, game.py, interface.py e obj.py, requirements.txt, além da pasta assets.

- main.py: arquivo principal responsável por inicializar o jogo. Nele foi criada uma classe que organiza as instâncias das demais classes que compõem o sistema, além da definição dos loops de cada execução de tela (menu, regras, partida e vencedor). Todos esses loops tratando os eventos do Pygame, como cliques do mouse, teclas pressionadas e o botão de fechar da janela.
- game.py: contém a classe Jogo, que gerencia a lógica principal das partidas de damas. Essa classe é responsável por controlar o turno dos jogadores, fazer o movimento da peça utilizando o mouse, fazer a validação de movimentos, caso haja um movimento obrigatório impedir movimento de outras peças, verificar condições de vitória e aplicar as regras específicas do jogo.
- interface.py: possui a classe Interface, responsável por desenhar os elementos gráficos do jogo e de todas as telas do jogo.
- obj.py: reúne duas classes principais:
  - Tabuleiro: A classe Tabuleiro é responsável por criar e gerenciar a disposição das peças no tabuleiro, modelo no formato de uma matriz 8x8.
  - Jogador: Representa um dos dois jogadores no jogo, cada um identificado por 'x' ou 'o', além de um nome e cor associados. Cada jogador tem um contador para as peças capturadas e também calcula a quantidade de peças restantes no tabuleiro.
- assets/: pasta destinada ao armazenamento das imagens utilizadas na interface do jogo, incluindo o logo do jogo, imagens para a tela do vencedor (nas cores laranja e roxa) e uma imagem para a tela de empate.
- requirements.txt: contém as dependências essenciais para a execução da aplicação.

## 2.3 EXPLICAÇÃO DA APLICAÇÃO/SOFTWARE

A aplicação desenvolvida consiste em um jogo de damas, estruturado com interface gráfica e funcionalidade interativa. O sistema é composto por três telas principais: tela inicial, tela de regras e tela do jogo.

Na tela inicial, ao executar o programa, o usuário encontra um menu contendo opções para iniciar uma nova partida ou acessar a seção de regras. Os elementos gráficos desta tela foram construídos utilizando a biblioteca Pygame, com o suporte de recursos visuais presentes na pasta assets do projeto.

A tela de regras apresenta ao usuário as instruções básicas do jogo de damas, incluindo informações sobre o objetivo do jogo, movimentação das peças, regras de captura, bem como o funcionamento da promoção de uma peça à dama. Esta seção tem por finalidade orientar o usuário, especialmente aqueles que não possuem familiaridade com as regras do jogo.

Ao selecionar a opção de iniciar uma partida, o usuário é direcionado para a tela do jogo. Nesta interface, o tabuleiro 8x8 é exibido juntamente com as peças dos dois jogadores. A interação ocorre por meio do mouse, permitindo que o jogador selecione uma peça e, em seguida, a casa de destino para realizar a jogada. A lógica de movimentação, verificação de jogadas válidas e alternância entre os turnos dos jogadores é controlada pela classe Jogo, enquanto os elementos visuais e a atualização gráfica são gerenciados pela classe Interface.

Durante a execução da partida, o sistema realiza automaticamente a validação das jogadas, verifica a possibilidade de capturas obrigatórias, promove peças quando estas alcançam a última linha do tabuleiro adversário e determina o vencedor ao final da partida.

## 2.4 ORIENTAÇÕES DE EXECUÇÃO DA APLICAÇÃO/SOFTWARE

Para executar o jogo desenvolvido, é necessário que o ambiente esteja com a linguagem de programação Python instalada, além de uma IDE de preferência. Durante o desenvolvimento, foi utilizada a IDE Visual Studio Code (VS Code).

Primeiramente, é necessário possuir o Python instalado no computador. A linguagem pode ser baixada por meio do site oficial: <https://www.python.org/>

Após a instalação do Python, recomenda-se que o usuário instale a IDE de sua preferência. O VS Code, por exemplo, pode ser obtido em: <https://code.visualstudio.com/>

Com esses requisitos atendidos, o usuário deverá realizar o download do projeto por meio do repositório disponível no GitHub. O jogo será disponibilizado em formato compactado (.zip). Após o download, é necessário extrair o conteúdo do arquivo para uma pasta local. Recomenda-se que essa pasta seja aberta diretamente na IDE, a fim de facilitar a navegação entre os arquivos.

O projeto pode ser executado de duas maneiras: por meio de um arquivo executável ou diretamente pelo código-fonte.

No caso do executável, basta localizar o arquivo com extensão “.exe” dentro da pasta extraída e dar um duplo clique para iniciar o jogo, sem a necessidade de configurar ambiente ou instalar bibliotecas.

Caso o usuário opte por executar o código-fonte, é necessário seguir os seguintes passos:

- Abrir o terminal na pasta do projeto.
- Criar um ambiente virtual com o nome "venv" utilizando o seguinte comando:  
`python -m venv venv`
- Ativar o ambiente virtual:
  - No Windows: `venv\Scripts\activate`
  - No Linux/macOS: `source venv/bin/activate`
- Instalar as dependências listadas no arquivo requirements.txt com o comando:  
`pip install -r requirements.txt`
- Executar o jogo pelo terminal com o seguinte comando:  
`python main.py`

Seguindo essas instruções, o jogo estará pronto para ser executado com todas as funcionalidades habilitadas.

## 2.5 REPOSITÓRIO

- <https://github.com/nallynx/Jogo-de-Damas>



### 3 CONCLUSÃO

O desenvolvimento do jogo de damas permitiu a aplicação prática dos conhecimentos adquiridos ao longo do semestre, especialmente no uso da linguagem de programação Python e da biblioteca Pygame. O projeto proporcionou uma compreensão mais aprofundada sobre a estruturação de sistemas, a divisão de responsabilidades em diferentes módulos e a criação de interfaces interativas com o usuário.

A separação lógica do código em diferentes arquivos e classes favoreceu a organização e a manutenção do software. Além disso, a construção de uma interface amigável e intuitiva contribuiu para uma melhor experiência do usuário durante a execução do jogo.

Os maiores desafios encontrados durante o desenvolvimento estiveram relacionados à criação das funções responsáveis pelas ações do jogo e à integração dessas funções com os objetos definidos no código, garantindo seu correto funcionamento durante a partida. Apesar dessas dificuldades, o resultado final atendeu aos objetivos propostos. A aplicação está funcional, com uma tela inicial, uma tela de regras e a execução completa de uma partida de damas entre dois jogadores.

## REFERÊNCIAS

PYINSTALLER DEVELOPMENT TEAM. *PyInstaller: bundle Python applications*. Disponível em: <https://pyinstaller.org>. Acesso em: 10 jun. 2025.

PYTHON SOFTWARE FOUNDATION. *Python: a programming language*. Disponível em: <https://www.python.org>. Acesso em: 10 jun. 2025.

PYTHONTURTLE. *PythonTurtle: learning programming with fun*. Disponível em: <https://pythonturtle.org>. Acesso em: 10 jun. 2025.

SHREINER, Pete; et al. *Pygame: Python Game Development*. Disponível em: <https://www.pygame.org>. Acesso em: 10 jun. 2025.