

Jogo Campo Minado

Neste projeto vamos implementar a versão clássica de um Jogo de Campo Minado. Para saber mais sobre o jogo, jogue um pouco em [Jogo 894151488 - Minesweeper Online](#).

Os projetos poderão ser realizados em dupla e cada dupla semanalmente determinará a velocidade com a qual irá evoluir os projetos determinando quais entregas fará na semana seguinte. Para isso, será utilizada a plataforma do Trello (veja o Anexo I).

O roteiro de execução do projeto é dividido em vários passos que envolvem assistir aulas, consolidar o aprendizado e desenvolver o projeto. Todo o desenvolvimento do projeto deverá possuir um diagrama que explique o que foi feito com o código salvo num espaço no Github (veja o Anexo II).

Ao assistir as aulas, um relatório individual deve sempre ser gerado. O relatório consiste em um resumo do conteúdo da aula - preferencialmente na estrutura de mapa mental.

A consolidação do aprendizado pode considerar classes criadas em etapas anteriores da própria consolidação.

IMPORTANTE 1: As duplas não devem fazer o projeto juntas para depois submeter no GitHub. As atividades do projeto devem ser divididas entre cada membro da equipe, de forma que cada um faça sua parte e coloque no repositório para que um outro integrante pegue e evolua. As duplas serão avaliadas pelos *commits* realizados no repositório.

IMPORTANTE 2: Existem ao longo do roteiro de execução vários *checkpoints* relacionados ao projeto. A cada checkpoint, o projeto deve ser extraído do Github e ter seu *upload* feito no Google Classroom. Só é possível considerar o avanço na próxima etapa quando os *checkpoints* anteriores tiverem sido realizados.

IMPORTANTE 3: as implementações da consolidação do aprendizado devem ser realizadas individualmente. Para isso, deve ser feito o *upload* da consolidação na atividade correspondente do Google Classroom.

IMPORTANTE 4: os relatórios de aprendizado requisitados a cada aula devem ser escritos individualmente. Os mesmos deverão ter o *upload* feito na atividade correspondente no Google Classroom.

IMPORTANTE 5: A cada semana, deve ser registrado as tarefas e avanços no Trello incluindo a atualização do diagrama de classes. Logo depois deve tirar um print do Trello e armazenado no classroom.

IMPORTANTE 6: o aprendizado de um paradigma de programação é não linear. Pense a respeito!

Roteiro

1. [Assista a Aula 01 - Introdução a OO e escreva sobre o que aprendeu](#)
2. [Assista a Aula 02 – Conceitos básicos e escreva sobre o que aprendeu](#)
3. **Consolidação do aprendizado 1**
4. **Consolidação do aprendizado 2**
5. [Assista a Aula 03 – String e Arrays e escreva sobre o que aprendeu](#)

6. **Consolidação do aprendizado 3**
7. **Consolidação do aprendizado 4**
8. **Assista a Aula 04 – Objetos, Classes e Métodos e escreva sobre o que aprendeu**
9. **Projeto:** Crie um diagrama de classe que inclua as **principais** classes, atributos e métodos.
10. **Projeto:** Identifique as classes do jogo Campo Minado.
11. **Projeto:** Identifique os atributos de cada classe.
12. **Projeto:** Identifique os métodos de acesso dos atributos quando aplicável, incluindo as validações e restrições necessárias.
Ex: `public void setColuna(int coluna) { if(coluna >= 0 && coluna < 7) {this.coluna = coluna;}}`
13. **Projeto:** identifique os demais métodos nas diferentes classes criadas.
14. **Projeto:** Crie um diagrama de classe que inclua **todas** as classes, atributos e métodos identificados.
15. **Projeto:** Implemente o código do diagrama criado que represente o clique do usuário numa célula.
16. **Consolidação do aprendizado 5**
17. **CHECKPOINT 1 – Upload do projeto**
18. **Assista a Aula 05 – Herança e escreva sobre o que aprendeu**
19. **Projeto:** considerando que uma célula pode ser bomba, vizinha a bomba ou espaço vazio, expanda a classe utilizada para representar cada célula do tabuleiro utilizando o conceito de herança.
20. **Consolidação do aprendizado 6**
21. **Projeto:** Inclua no diagrama de classes, as classes para comunicação com o usuário e que permitirão a execução do jogo.
22. **Projeto:** permita que o usuário marque uma célula fechada com bandeira (candidata a ter bomba). Células marcadas com bandeira não podem ser abertas se a bandeira não for retirada antes.
23. **Projeto:** ao abrir uma célula que não tem bomba na vizinhança, todas suas células vizinhas devem ser abertas de maneira sucessiva.
24. **Projeto:** Implemente o código do diagrama atualizado que seja necessário para que 2 usuários consigam interagir.
25. **Assista a Aula 06 – Encapsulamento, Pacotes e Modificadores e escreva sobre o que aprendeu**
26. **Projeto:** separe as classes do jogo em pacotes (ao menos dois)
27. **Consolidação do aprendizado 7**
28. **Projeto:** revise o código e verifique o nível de encapsulamento de cada classe e se as boas práticas estão sendo respeitadas.
29. **CHECKPOINT 2 – Upload do projeto**
30. **Assista a Aula 07 – Polimorfismo e escreva sobre o que aprendeu**
31. **Projeto:** crie uma alternativa ao Campo Minado chamada Campo Minado Maluco. Em Campo Minado Maluco, uma célula pode ser ou não maluca. Quando a célula for maluca, o jogador ao marcar a célula como bomba colocando uma bandeira, a célula poderá alterar seu status de bomba com uma probabilidade determinada pelo nível de maluquice do jogo.
Obs: pense em alguma estratégia para que os vizinhos da célula maluca sejam devidamente notificados para quando abertos exibirem corretamente o número de bombas ao seu redor caso a célula maluca tenha seu status de bomba alterado.

32. **Consolidação do aprendizado 8**

33. **Assista a Aula 08 – Classes Abstratas e Interfaces e escreva sobre o que aprendeu**

34. **Projeto:** identifique quais classes podem ser abstratas e faça a devida alteração.

35. **Projeto:** adicione uma interface para aumentar o desacoplamento da classe que representa o tabuleiro da classe que representa o jogo.

36. **Assista a Aula 09 – Exceções e escreva sobre o que aprendeu**

37. **Projeto:** inclua uma classe que represente uma exceção de valor de atributo inválido. Esta classe deve ser capaz de receber em seu construtor uma String que represente uma mensagem associada à situação de exceção eventualmente ocorrida.

38. **Projeto:** identifique exceções para cada método set ao verificar se o valor passado para preenchimento é aceitável, e outras possíveis exceções do jogo.

39. **Projeto:** Atualize o diagrama de classes considerando diferentes tamanhos de tabuleiro e quantidade de bombas para os níveis inicial, intermediário e difícil.

40. **Projeto:** Implemente o diagrama completo dando o tratamento adequado para conseguir o comportamento esperado.

41. **CHECKPOINT 3 – Upload do projeto**

42. **Assista a Aula 10 – Generics e Estruturas de Dados e escreva sobre o que aprendeu**

43. **Projeto:** implemente a informação de recorde do jogo exibindo pelo menos os 10 maiores vencedores.

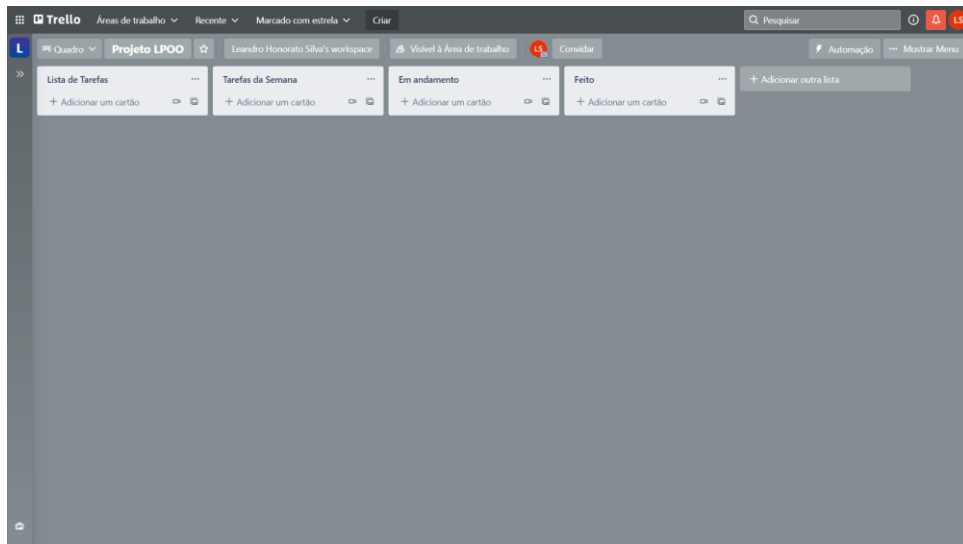
44. **Projeto:** implemente a interface gráfica do jogo.

45. **CHECKPOINT 4 – Upload do projeto**

Anexo I

Trello

O [Trello](#) será o nosso ambiente colaborativo para gerenciamento do projeto. Para isso, cada membro da equipe deve criar uma conta no Trello e criar um Quadro. O nome do Quadro deve ser “LPOO 2021.1 <nome equipe>”. O quadro deve possuir quatro listas, conforme ilustrado abaixo: Lista de tarefas, Tarefas da Semana, Em andamento e Feito.



A Lista de Tarefas deve conter todas as tarefas previstas ao longo do semestre (podem ser adicionadas mais tarefas ao longo da execução do projeto). **Semanalmente, haverá uma reunião de acompanhamento do projeto.** Nesta reunião, o grupo deve apresentar as atividades que foram realizadas na semana que passou (estão na lista Feito) e quais atividades serão realizadas na semana seguinte (Tarefas da Semana).

Anexo II

Github

O GitHub é o local onde os códigos desenvolvidos devem ser armazenados. As etapas para criação de um repositório no GitHub são encontradas aqui: [Introdução ao GitHub - GitHub Docs](#).

De início, você não precisa se preocupar com funcionalidades mais avançadas como *branch* (bifurcações). É possível utilizar exclusivamente o *browser* para interagir com o repositório, embora seja mais fácil utilizar um cliente Git (conforme descrito no tutorial).

Para fins de avaliação, será avaliado o registro de *commits* no projeto.