

Áreas: TI & Computação	UC: Programação e Soluções Computacionais e Modelagem de Software	
Professores: Gustavo Fernandes, Rafaela Moreira, Samara Leal	Trabalho Final (A3)	
Grupo:	Data Entrega: 07/12/21 - Documentação 09/12/21 - Código + vídeo	

PROJETO

Instruções

1. A turma deve ser dividida em grupos de **no máximo** 5 alunos. O grupo deve criar uma organização no GitHub, adicionar os professores (rafapcmor) e preencher o [FORMULÁRIO](#) (apenas 1 pessoa por grupo responde ao formulário) **Obs: todos devem comitar.**
2. O programa deverá ser desenvolvido seguindo todas as técnicas necessárias da orientação a objetos (herança, polimorfismo, encapsulamento, tratamento de exceções). Não basta funcionar, tem que seguir os princípios de OO.
3. Deve ser entregue o código funcionando, o arquivo sql do banco com os dados, a documentação e um vídeo como repositórios na organização. **Mudanças após às 23:59 do dia 09/12/2021 não serão consideradas.**
4. Vale ressaltar que não será admitida **cópia de trabalhos**, de espécie alguma. Entendam **não admitir cópia** como sendo **premiação com nota zero para ambos os grupos.**
5. **07/12/2021** apresentar documentação e fazer a avaliação 360°. Todos os integrantes devem comparecer no dia. O aluno que não demonstrar conhecimento sobre o trabalho será penalizado na nota.
6. **09/12/2021** entrega do código + vídeo. Alterações após essa data não serão consideradas.

Bom trabalho a todos!

“É fazendo que se aprende a fazer aquilo que se deve aprender a fazer.” [Aristóteles](#)

Objetivo

O objetivo deste trabalho é desenvolver um jogo de perguntas e respostas.

Descrição do Trabalho

- O jogo deve permitir uma batalha de 2 jogadores.
- Os jogadores escolhem os personagens que deverão já estar cadastrados no sistema.
- Deve-se definir uma política para decidir quem começará o jogo.
- A jogada deve ser intercalada.
- Se o jogador errar, o tempo de vida é reduzido.
- Novos atributos e novas classes podem ser adicionados conforme a necessidade.
- Um nome criativo para o jogo deve ser escolhido e mostrado na tela inicial.
- Tratar as situações de exceção. Exemplo: não é permitido começar uma partida sem os personagens, não é permitido ter tempo de vida negativo,... Faça todo o tratamento necessário.
- Use polimorfismo sempre que possível. Um bom teste para verificar o uso correto do polimorfismo em seu projeto é imaginar a inclusão de um novo tipo ao sistema, as alterações no código devem ser mínimas.
- Para cada classe, deve-se obedecer as seguintes orientações:
 - • Atributos: devem ser privados ou protegidos.
 - • Método construtor que inicializa os atributos necessários através de parâmetros.
 - • Métodos get e set para obter e modificar atributos, quando necessário.
- O sistema deve ser claro para qualquer usuário que o utilizar.
- Criar um repositório **Code** e adicionar o código.

I. Classes essenciais

- **Personagem**
 - Cada personagem tem um nome, que será utilizado para referenciá-lo durante o jogo, status (ativo ou inativo), habilidades(exemplo: força, inteligência) e um tempo de vida que inicia em 100%.
- **Pergunta**
 - Cada pergunta tem um enunciado e uma lista de alternativas.
- **Alternativa**
 - Cada alternativa tem um enunciado e um status (**true** para alternativa correta e **false** para incorreta.)
- **Jogador**
 - Cada jogador tem um nome, apelido, email, telefone.

II. Base de Dados (**opcional**)

Uma base de dados para salvar as perguntas e respostas deve ser criada utilizando a linguagem sql.

- Criar um repositório **Database** e adicionar o arquivo .sql
- A base de dados deve ter no mínimo 10 perguntas relacionadas a Programação Orientada à Objetos ou Modelagem de Software.

- O código da conexão com o banco de dados e exemplo de CRUD será disponibilizado na pasta **PROJETO** no Drive.

III. Elementos visuais e interativos

Console	GUI (opcional)
Menu Inicial	Tela Inicial
<ul style="list-style-type: none"> • Inicializar o jogo. • Informações dos desenvolvedores. • Explicação do jogo. • Sair. 	<ul style="list-style-type: none"> • Composta por um botão para inicializar o jogo. • Um item de menu com informações dos desenvolvedores. • Um item de menu com a explicação do jogo.
Cadastro de Jogador	Tela de Cadastro de Jogador
<ul style="list-style-type: none"> • Podem ser cadastrados somente 2 jogadores por vez. • Os atributos mínimos já foram especificados no item I. 	<ul style="list-style-type: none"> • Podem ser cadastrados somente 2 jogadores por vez. • Os atributos mínimos já foram especificados no item I.
Menu Escolha de Personagem	Tela de Escolha de Personagem
<ul style="list-style-type: none"> • Apresentará os personagens previamente cadastrados. • No mínimo 5 personagens devem ser cadastrados. • Cada jogador pode escolher apenas um deles. • Quando um personagem é escolhido, não pode aparecer como opção para o outro jogador. 	<ul style="list-style-type: none"> • Essa janela apresentará os personagens previamente cadastrados. • No mínimo 5 personagens devem ser cadastrados. • Cada jogador pode escolher apenas um deles. • Quando um personagem é escolhido, não pode aparecer como opção para o outro jogador.
Partida	Tela da Partida
<ul style="list-style-type: none"> • Deve apresentar os personagens escolhidos e a porcentagem de vida. • Abaixo, deve ter o enunciado e as alternativas da rodada com letras. • Quando escolhida uma alternativa, uma mensagem perguntando se tem certeza da resposta deve aparecer. • Se escolher não, volta para a pergunta. • Se escolher sim e a resposta estiver correta 	<ul style="list-style-type: none"> • Deve apresentar os personagens escolhidos e suas respectivas barras de vida. • Abaixo, deve ter o enunciado em uma caixa de texto e as alternativas da rodada em botões. • Quando escolhida uma alternativa, uma caixa de diálogo deve ser acionada perguntando se tem certeza da resposta. • Se escolher não, volta para a tela anterior. • Se escolher sim e a resposta estiver correta

<ul style="list-style-type: none"> ○ Exibe uma mensagem parabenizando. ○ Exibe o conteúdo da pergunta seguinte. ● Se escolher sim e a resposta estiver errada <ul style="list-style-type: none"> ○ Exibe uma mensagem de erro. ○ Erro de resposta é acionado. ○ Reduz a vida do personagem em 5%. ○ Exibe o conteúdo da pergunta seguinte se o jogo não acabar. ● Deve perguntar se quer parar o jogo. Mas para isso, os dois jogadores devem estar de acordo. Se o outro jogador não estiver, 15% do tempo de vida do personagem do jogador solicitante é repassado para o personagem do outro jogador. Só depois verifica o vencedor (quem tem mais tempo de vida). ● O jogo termina quando a vida de um dos personagens é zerada. 	<ul style="list-style-type: none"> ○ Exibe uma mensagem parabenizando. ○ Exibe o conteúdo da pergunta seguinte. ● Se escolher sim e a resposta estiver errada <ul style="list-style-type: none"> ○ A tela de erro é acionada. ○ Reduz a vida do personagem em 5%. ○ Exibe o conteúdo da pergunta seguinte se o jogo não acabar. ● Deve ter um botão de parar. Mas para isso, os dois jogadores devem estar de acordo. Se o outro jogador não estiver, 15% do tempo de vida do personagem do jogador solicitante é repassado para o personagem do outro jogador. Só depois verifica o vencedor (quem tem mais tempo de vida). ● O jogo termina quando a vida de um dos personagens é zerada.
Erro de Resposta	Tela de Erro de Resposta
<ul style="list-style-type: none"> ● Deve conter o enunciado e seu gabarito. 	<ul style="list-style-type: none"> ● Acionada quando o jogador errar uma pergunta. ● Deve conter o enunciado e seu gabarito.
Pontuação	Tela de Pontuação
<ul style="list-style-type: none"> ● Após o término da partida, apresentar a pontuação final do jogador (tempo de vida do personagem), que, considerando ser um jogo educativo, não pode ser abaixo de zero. Voltar para a tela inicial. 	<ul style="list-style-type: none"> ● Após o término da partida, apresentar a pontuação final do jogador (tempo de vida do personagem), que, considerando ser um jogo educativo, não pode ser abaixo de zero. Deve apresentar um botão para voltar à tela inicial.

IV. Interface Gráfica (extra – 3 pontos)

- A interface gráfica do jogo é opcional, mas vale a pena o esforço! **Use a criatividade!!**

V. Trilha sonora (extra – 2 pontos)

- Uma trilha para o início do jogo, uma para quando o usuário acertar e outra para quando ele errar.

VI. Vídeo

- Criar um repositório **Vídeo** e adicionar um vídeo **no máximo** 3 min mostrando o projeto.

VII. Documentação

- Criar um repositório **Documentação** no github e no Readme adicionar:
 - Objetivo do Sistema
 - Definições e Siglas
 - Adicionar um link para definições e siglas
 - Cronograma previsto
 - adicionar um link com o cronograma
 - Colaboradores
 - Adicionar os integrantes do grupo
 - Backlog do Produto
 - Montar um kanban com ID, Nome, Importância, PH, Demonstrar, Notas (histórias de usuário)
 - Sprints
 - Definir e descrever os Sprints no Kanban.
 - Diagrama de Casos de uso do sistema.
 - Adicionar um link ou imagem.
 - Diagrama de Classes do sistema.
 - Adicionar um link ou imagem.
 - Modelo Entidade Relacionamento da base de dados do sistema.
 - Adicionar um link ou imagem.
 - Modelo Relacional da base de dados do sistema.
 - Adicionar um link ou imagem.

Obs: deixar link público.