



Nome: _____

Aproveitamento: _____

E-mail: _____

AVALIAÇÃO PYTHON

As atividades descritas a seguir contemplam e pretendem avaliar o aproveitamento dos seguintes cursos: Python 3 I e Python 3 II, realizados na plataforma Alura durante o programa.

Proposta: Analisar e desenvolver um jogo de aventura baseado em narrativa e situações de escolha.

Instruções Gerais

1) A avaliação é *individual* e com consulta (aos cursos, a documentação oficial e o material produzido a partir deles disponível no repositório individual);

2) Baixar o conteúdo de <https://github.com/myreli/pdb-python> que serve como material de apoio. As únicas bibliotecas externas permitidas são: *sys* e *random*;

3) É **obrigatório** o uso de listas, laços de repetição, estruturas condicionais, funções e outros recursos vistos no curso. Por outro lado, é **vetado** o uso de orientação a objetos. Além disso, espera-se código otimizado, lógica de programação, blocos reutilizáveis e boa organização;

4) Ao concluir a avaliação, enviar por e-mail para myreli.barros@compasso.com.br e leticia.cecotti@compasso.com.br o link do repositório, o arquivo zip do projeto (SeuNome.zip) e quaisquer observações necessárias com o assunto **[PDB][2018] Avaliação Python**.

May the odds be ever in your favor...

Atividade 00: first steps...

1) verificar a instalação de Git, Visual Studio Code e Python na máquina e reportar quaisquer problemas;

2) clonar o repositório privado individual do projeto e baixar o material para a avaliação;

3) os itens obrigatórios para o jogo são:

- a. 5 personagens (jogador + 4);
- b. 2 atributos do personagem (vida e dano);
- c. 4 cenários (cenário inicial, 2 intermediários e final);

Atividade 01: versionando com Git II

Instruções gerais para versionamento do projeto no repositório individual, mais questões de versionamento estarão presentes no decorrer da avaliação.

1) Criar para a avaliação a *branch* **pdb_python** e fazer um *commit* inicial, criando a estrutura e um README.md do projeto;

2) Seguir o padrão de mensagens de *commit*:

o atividade [numero]: descrição objetiva

descrição longa, detalhando o que foi feito e observações sobre o que falta, etc...

▪ Exemplo:

atividade 02: start - menu principal

criado menu principal para o usuário, faltando opção de sair do jogo.

3) **EXTRA:** Criar uma *branch* para cada atividade, com o prefixo **pdb_python/** (ex.: **pdb_python/fights** ou **pdb_python/atividade02**), fazer os *commits* nessa *branch* e o **merge** das *branches* de atividades na *branch* **pdb_python**;

4) Fazer **merge** da *branch* **pdb_python** na **master**.

Atividade 02: start

Inicializar o jogo com configurações básicas (nomes de personagens e nome do usuário, ...) e os elementos iniciais necessários. Essa atividade deve englobar:

- menu de inicialização do jogo com opção para sair ou iniciar;
- receber nome do jogador;
- ler o arquivo que contém uma lista de nomes "personagens" anexado ao projeto e utilizar para nomear os outros personagens do jogo.

Atividade 03: welcome

Apresentar as primeiras interações com o enredo e os outros elementos. Essa atividade deve, no mínimo:

- apresentar a narrativa introdutória;
- apresentar no mínimo 2 tomadas de decisões (ex.: entrar em um cenário ou outro, aceitar ou não amizade, conversar ou não com outro personagem, ...);

- a partir dessas escolhas do usuário, estabelecer relações (exemplos: amigos, inimigos, locais de preferência, medos, etc...) entre ele e os outros elementos através de **dicionários**;
- **GIT**: fazer merge do desenvolvimento atual para a **master**.

Atividade 04: fight for life

Apresentar os conflitos que o personagem enfrenta. Considerando:

- ao menos dois conflitos Personagem VS Personagem;
- todos os conflitos devem envolver tomadas de decisões e interferir no enredo principal (exemplo: ataca ou foge);
- **GIT**: durante o desenvolvimento da atividade 04 na branch `pdb_python`, **antes de commitar**, fazer `stash` das alterações atuais (conflitos e mudanças de enredo – ou todas) e considerar o seguinte:
 - voltar a **Atividade 02: start** e incluir no personagem principal um atributo de `SCORE`, que sirva para registrar sua pontuação. Fazer `commit` dessa alteração e fazer merge para a branch **master**.
 - Para ser avaliada, é necessário incluir os outputs (`print` ou `arquivo.txt`) dos seguintes comandos:
 - `git stash list`
 - `git stash show stash@`
 - `git diff master stash@{0}`
 - Na branch de desenvolvimento (`pdb_python`), aplicar as alterações do `stash` e continuar com o desenvolvimento.

Atividade 05: Game Over

Apresentar o desfecho final e encerramento do jogo + oportunidade de reiniciar.

- 1) apresentar o desfecho da narração com mais algum conflito final Personagem versus Personagem;
- 2) apresentar o vencedor e dar a opção de voltar ao meu inicial;
- 3) **DESAFIO (EXTRA)**: armazenar em um arquivo o histórico dos jogadores (score e nome) e acrescentar a opção no menu principal de ver os scores.