**Desenvolvimento: Jogo da Vida**

**Especificações do jogo (requisitos fornecidos pelo usuário)**

a) Deverá ter um tabuleiro com, no mínimo, N=50 posições lineares.  
b) Destas, S=10 corresponderão a posições de Sorte e R=10, a posições de Revés.  
c) O jogo deverá aceitar um número 2<J<4 de Jogadores.  
d) Deverá ter uma roleta com P posições, correspondente ao número de casas a serem avançadas.  
e) Deverão ser cadastrados 20 itens de bônus e 20 de revés, com texto e valor.  
f) Ao cair em uma posição de bônus, deverá ser sorteado um item do cadastro de bônus, o qual apresentará uma mensagem (texto) e indicará quanto dinheiro (valor) o jogador ganhará. Para revés é análogo, porém, com pagamentos.  
g) Ao iniciar, os nomes dos jogadores serão ser cadastrados.  
h) O sistema sorteará a ordem de início.  
i) Quando um jogador chegar (primeiro) ao final do tabuleiro, o saldo de todos será comparado e vencerá a partida aquele jogador com maior saldo.  
j) Deverá registrar (arquivo)’ o histórico de jogadores, posição final e saldo final de cada um.

**Observações:**

O cliente pode não ter informado todos os requisitos (pode ter esquecido algum);  
2) O cliente não diz como deverá ser implementado o sistema;  
3) Algum requisito poderá demandar uma análise específica devido a sua complexidade;  
4) Não existe uma forma única para resolver o problema;  
5) Sabendo da concorrência, agregue valor/funcionalidades ao seu sistema. Cuidado para isso não consumir tempo de requisitos do cliente;  
6) Tente executar o jogo na mente ou no papel para entender bem o problema e as partes da solução;  
7) Categorize as tarefas essenciais, as importantes e as que poderão ficar para versão futura. Execute as tarefas por grau de importância;  
8) Recomendo estudar a metodologia ágil Extreme Programming (XP) (youtube.com/watch?v=ALvpFMcL-dI)   
9) Priorize módulos menores, pois poderão ser mais aproveitados ao longo do programa;  
10)Sempre faça um roteiro das tarefas da semana para evitar esquecimentos e perca do foco.

Definição para desenvolvimento

Observações:

* Este jogo esta sendo desenvolvido para ambiente online, haverá um servidor rodando em Python, este encarregado por realizar todo processamento das ações dos jogadores.
* A interface gráfica web será hospedada em domínio publico, esta não terá nenhum contato direto com o Core/Servidor, todas as informações serão armazenadas/trocadas por meio do Firebase Real time Database.
* A interface gráfica desktop desenvolvida em python, sera distribuida por meio de um dominio publico e sera executado no desktop do jogador, esta não terá nenhum contato direto com o Core/Servidor, todas as informações serão armazenadas/trocadas por meio do Firebase Real time Database.

* Desenvolvimento do Core do Game (Desenvolvimento orientado a Objetos)
  + Principais objetos
    - Game, o jogo em si
    - Player, cada jogador
    - ItemTrilha, cada casa da trilha do jogo
  + Modulos
    - Game.\_\_init\_\_, cria uma instancia do game
    - Game. AddPlayer, adiciona +1 jogador ao game
    - Game.SortTrilhaGame, Sorteia onde fica cada posição de sorte ou revez na trilha do game
    - Game. Configure, configura o game após alterar configurações
    - Game. GiraRoleta, sorteia um numero de 1 a n, sendo n o numero de possibilidades
    - Game. LastItem, retorna o ultimo elemento de um dicionário
    - Game.realizaJogada, realiza a jogada do player
    - Game. Start, da inicio ao jogo, controla de quem é a vez de jogar, e se alguém já chegou ao final da trilha
    - Item\_Trilha. \_\_init\_\_, Cada casa da trilha
    - Item\_Trilha. ExecAction, executa uma ação de bonus ou reves presente na casa
    - Item\_Trilha. Entra, acionado quando o jogador entra na casa, bloqueia a casa evitando que 2 players tentem ocupar ao mesmo tempo
    - Item\_Trilha. Sair, libera a casa para que outro possa ocupar no futuro
    - NewPlayer.\_\_init\_\_, cria um novo player
    - NewPlayer.andar, executa ações para player sair da casa atual, e entrar na próxima
  + Variaveis
    - Game.id, identificador único de cada instancia do jogo [uuid(4)]
    - Game.config,um dicionário contendo
      * Posicoes, numero de casas no tabuleiro, defaut(50)
      * NPlayers, numero de jogadores permitido, de X a Y, defaut([2,4])
      * TamRoleta, numero de possibilidades da Roleta, defaut(6)
    - Game. P\_Sorte, uma lista de listas contendo no formato [“texto do item sorte”, int(valor)]
    - Game. P\_Reves, uma lista de listas contendo no formato [“texto do item revez”, int(-valor)]
    - Item\_Trilha.number , numero da casa no Game
    - Item\_Trilha.status, 0 para livre, 1 para ocupado
    - Item\_Trilha.player, o player que esta ocupando ou None para casa vazia
    - Item\_Trilha.action, uma ação de bônus ou revez, ou None pra nenhuma ação
    - NewPlayer.id, id unico do jogador [uuid(4)]
    - NewPlayer.nome, nome do jogador, obrigatório ser fornecido
    - NewPlayer.saldo, saldo do jogador, inicial em 1000, alterado no decorrer do jogo
    - NewPlayer.posicao , posição do jogador, inicial = 0, alterado no decorrer do jogo
* Desenvolvimento da Interface Gráfica Web

Interface gráfica web

1. Interface Gráfica web
   1. Cada instancia terá um id único
   2. Cada instancia poderá comportar vários jogadores
   3. Poderá criar uma nova partida e optar por
      1. Disponibilizar para qualquer jogador online
      2. Compartilhar um link de jogo com amigos
      3. Jogar com amigos pela mesma/única instancia
   4. Participar de partidas que forem disponibilizadas de forma publica

Verifique a extrutura do banco de dados em Data.json