

Trabalho Prático – Jogo da Velha Multiplayer Web Oracle

Etapa 1 – Elaboração do Universo de Discurso (UdD)

O projeto que aqui quer ser apresentado é de um jogo da velha multiplayer desenvolvido em HTML, CSS, Javascript, ASP.NET Core e C#, consumindo dados de um banco de dados relacional Oracle XE (eXpress Edition) 11.2g. Projeto tal qual está em paralelo com a disciplina de Programação III, lecionada pelo professor Jonathan Borges.

Nesse jogo, teremos então as seguintes entidades: Jogadores (Tabela PLAYER), salas (Tabela ROOM), Jogadores em uma sala (Tabela ROOM_PLAYER), Tabuleiro com a jogada de cada jogador (Tabela GAME_BOARD), pontuação de cada jogador (Tabela SCORE) e um histórico de partidas jogadas para todos os jogadores (Tabela MATCH_HISTORY).

Entidade Jogador

A entidade “jogador” será responsável por armazenar o nickname do jogador, a senha (criptografada por md5 Oracle), a pontuação individual do jogador e se ele está conectado ou não na aplicação (esse último para fins de mensageria e amizade entre jogadores, que acabou não sendo desenvolvida). O ID do jogador será auto incremento alimentado pelo sequence “PLAYER_ID”. O nickname será um campo do tipo texto com no máximo 16 caracteres. A senha também será um campo do tipo texto, mas com no máximo 100 caracteres, esse tamanho se da pela criptografia do texto em md5 do Oracle, porém na aplicação só permitidos até no máximo 25 caracteres de comprimento. A pontuação será inicialmente o valor 0, sendo incrementa de 10 em 10 a cada vitória dentro dos jogos. A propriedade “ativo” será responsável por identificar se o jogador está ativo ou não no sistema, uma vez que não poderá haver dois jogadores logados na mesma coisa simultaneamente. Também íamos utilizar ela para indicar aos amigos desse jogador que ele está ativo para jogar, porém essa última acabamos não desenvolvimento devido o prazo.

Entidade Salas

A entidade sala, será relativamente simples, apenas armazenando chave-valor, ou seja, ID e descrição da sala. O ID será um numérico auto incremento alimentado pela sequence ROOM_ID. O campo descrição irá ser do tipo texto com capacidade máxima de 20 caracteres, sendo único na tabela, ou seja, não poderá haver salas com id’s diferentes com o mesmo nome (Sala teste, por exemplo).

Entidade Jogadores em uma sala

A entidade jogadores em uma sala será responsável por armazenar todos jogadores que estão em um asala, onde o ID da sala será uma PK e FK com a tabela de salas. O campo de jogador também será PK e FK com tabela jogadores. Há também um campo responsável por armazenar a data que cada jogador entrou na sala, uma vez que o jogador mais antigo será o administrador dessa sala.

Entidade Tabuleiro com a jogada de cada jogador

A entidade tabuleiro com a jogada de cada jogador será responsável por armazenar a posição que cada jogador escolheu, dessa forma, também servirá para dizer quem está na vez de jogar

Entidade Pontuação

A entidade de pontuação será responsável por dizer qual é o range de cada nível, uma vez que cada jogador terá sua própria pontuação e sua classificação “mundial” dentro do game. O ID será auto incremento, alimentado pela sequence SCORE_ID. A pontuação inicial irá dizer o valor inicial de uma determinada classificação. A pontuação final irá dizer qual é o valor final para aquela classificação. A descrição será um campo texto com capacidade para 30 caracteres e irá ser o descritivo da classificação (Iniciante III, por exemplo).

Entidade Histórico de Partidas

A entidade histórico de partidas será responsável por armazenar os jogos que cada jogador realizou. Nela irá conter o jogador em questão, o oponente, quando houve o jogo e se teve vitória. Aqui irá haver uma chave composta tripla entre jogador, oponente e data de jogo.

Etapa 2 - Projeto da Base de Dados

Na figura 1 é possível visualizar o MER do nosso projeto, criado no MySql Workbench (apesar de ser Oracle) e editado no paint para adição de informações como pk's, fk's e afins.

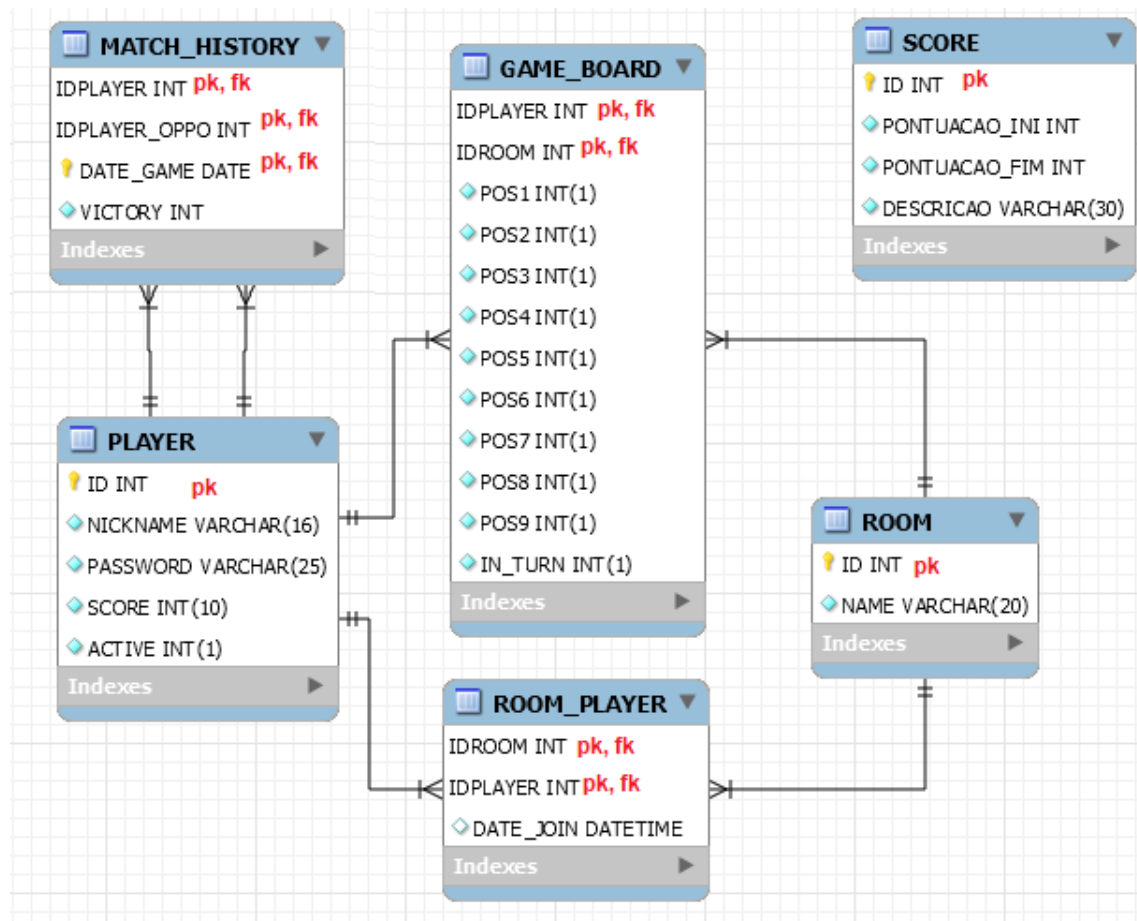


Figura 1. MER do projeto

Etapa 3 – Prototipação do SI sobre a Base de Dados

A seguinte view (VW_ICON), irá mostrar todos os ícones da base de dados:

```
CREATE OR REPLACE FORCE VIEW VW_ICON AS
SELECT ID,
       NAME,
       ADDRESS
FROM ICON
```

A seguinte view (VW_GAME_BOARD), irá mostrar todos os tabuleiros da base de dados:

```
CREATE OR REPLACE FORCE VIEW VW_GAME_BOARD AS
SELECT GM.IDROOM,
       GM.IDPLAYER,
       GM.POS1,
       GM.POS2,
       GM.POS3,
       GM.POS4,
       GM.POS5,
       GM.POS6,
       GM.POS7,
       GM.POS8,
       GM.POS9,
       GM.IN_TURN
FROM GAME_BOARD GM
```

A seguinte view (VW_USER), irá mostrar todos os jogadores da base de dados, sem filtro:

```
CREATE OR REPLACE FORCE VIEW VW_USER AS
SELECT ID,
       NICKNAME,
       PASSWORD,
       SCORE,
       ACTIVE,
       ICON_GAME
FROM PLAYER;
```

Obs.: Optamos por criptografar a coluna “Password” com md5, fornecido pelo próprio Oracle.

A seguinte função (md5) é responsável por criptografar a senha dos jogadores:

```
CREATE OR REPLACE FUNCTION md5 (valor VARCHAR) RETURN VARCHAR2 IS
    v_input VARCHAR2(2000) := valor;
    hexkey VARCHAR2(32) := NULL;
BEGIN
    hexkey := RAWTOHEX(DBMS_OBFUSCATION_TOOLKIT.md5(input =>
    UTL_RAW.cast_to_raw(v_input)));
    RETURN NVL(hexkey, '');
END;
```

Fonte: <https://www.oraclehome.com.br/2011/08/30/criptografando-em-md5-no-oracle/>

A seguinte view (VW_TOP_PLAYERS), irá mostrar os melhores jogadores da base de dados, ordenando de forma decrescente a pontuação dos jogadores a fim de mostrar os jogadores com maior pontuação de cima para baixo ↓:

```
CREATE OR REPLACE FORCE VIEW VW_TOP_PLAYERS AS
SELECT PL.ID,
       PL.NICKNAME,
       PL.SCORE,
       (SELECT TRIM( SC.DESCRICAO )
        FROM SCORE SC
        WHERE PL.SCORE >= SC.PONTUACAO_INI
        AND PL.SCORE <= SC.PONTUACAO_FIM) DESCR_SCORE
```

```
FROM PLAYER PL
ORDER BY PL.SCORE DESC,
        PL.NICKNAME
```

A seguinte view (VW_PLAYER_BY_LEVEL), irá mostrar a quantidade de jogadores por nível:

```
CREATE OR REPLACE FORCE VIEW VW_PLAYER_BY_LEVEL AS
SELECT SC.DESCRICAO NIVEL,
       COUNT(PL.ID) PLAYERS
FROM   SCORE SC,
       PLAYER PL
WHERE  PL.SCORE >= SC.PONTUACAO_INI
       AND PL.SCORE <= SC.PONTUACAO_FIM
GROUP BY SC.DESCRICAO
```

A seguinte view (VW_MATCH_HISTORY), irá mostrar o histórico de partidas de um determinado jogador (PLAYER). Nela será possível descobrir quem foi seu oponente (OPPONENT), a data do jogo (DATE_GAME) e se houve vitória ou derrota (VICTORY):

```
CREATE OR REPLACE FORCE VIEW VW_MATCH_HISTORY AS
SELECT PL1.NICKNAME PLAYER,
       PL2.NICKNAME OPPONENT,
       MH.DATE_GAME,
       MH.VICTORY
FROM   MATCH_HISTORY MH,
       PLAYER PL1,
       PLAYER PL2
WHERE  MH.IDPLAYER = PL1.ID
       AND MH.IDPLAYER_OPPO = PL2.ID
```