

# //GameLearning

```
list<string> Time = new list<string>();
```

```
Time.Add("Gustavo Fuzinelli");
```

```
Time.Add("Paulo Belucci");
```

```
Time.Add("Silvio da Col");
```



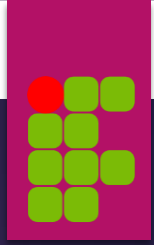
# //Objetivo

```
Console.WriteLine("Viabilizar o ensino da  
programação através de uma plataforma que visa  
facilitar a interação professor/aluno e  
auxiliar através da gamificação o processo de  
aprendizagem.");
```



# //Justificativa

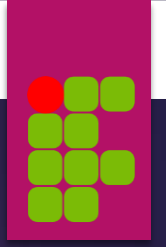
Tendo em vista o alto índice de evasão dos cursos de tecnologia por conta dos conteúdos teóricos, enxergamos a possibilidade de, através do GameLearning(plataforma) e conteúdos lúdicos(jogos), fomentar o interesse dos alunos pelo conteúdo aplicado e assim diminuir o índice de evasão mencionado.



# //A plataforma

`Console.WriteLine("Denominada GameLearning  
consiste em um ambiente interativo onde  
professores poderão criar exercícios para os  
alunos, como apoio ao material didático  
disponível.");`

# //Engenharia Software



- ❑ Levantamento Requisitos

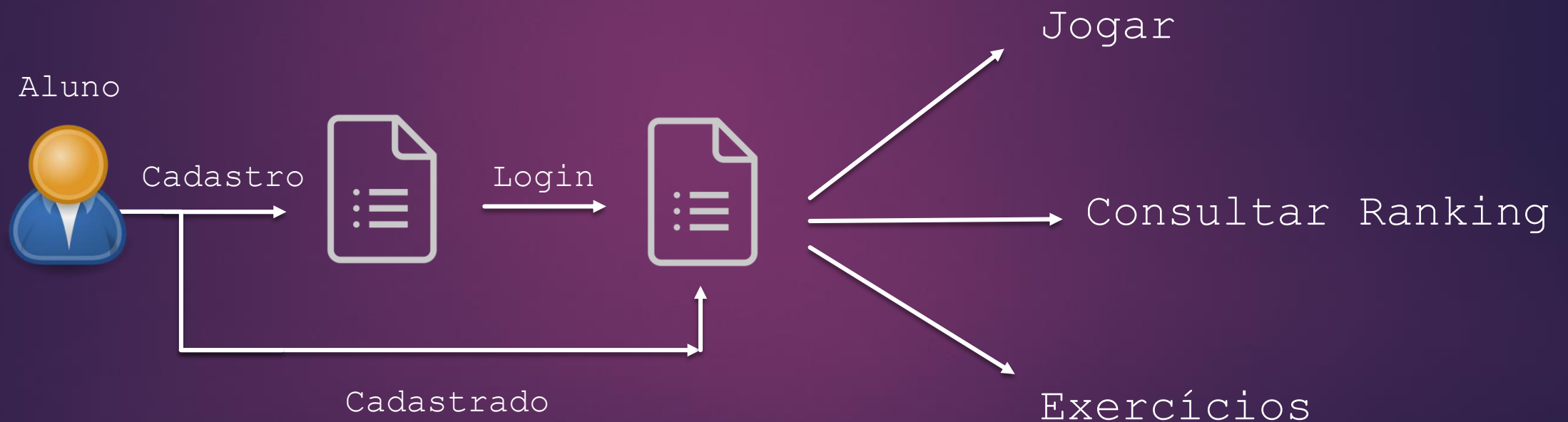
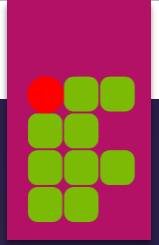
- ❑ Etnografia
  - ❑ Prototipagem
  - ❑ Brainstorming

- ❑ Metodologia

- ❑ SCRUM
  - ❑ Trello (ferramenta organização)

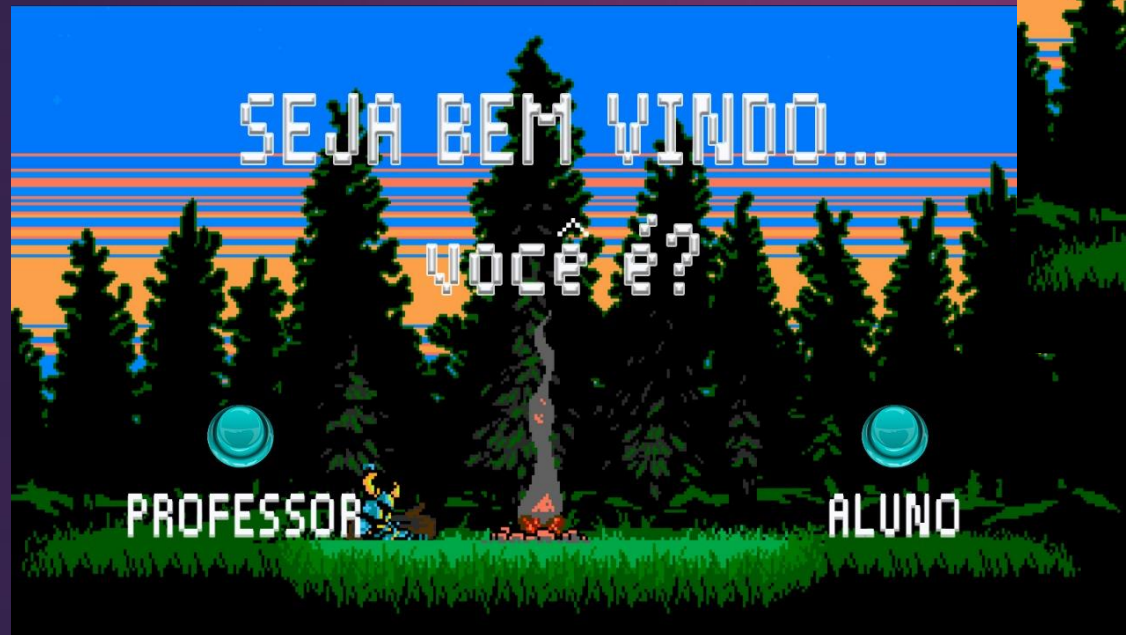
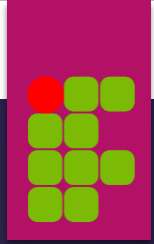
# //Engenharia Software

## Diagrama de Caso de Uso

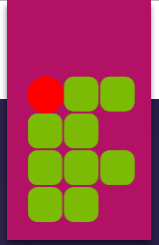


# //Engenharia Software

PROTOTIPAÇÃO – Versão\_1



# //Engenharia Software

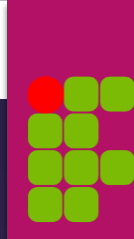


PROTOTIPAÇÃO – Versão\_Final

- Design Clean
- Visual Retrô
- Baseado em jogos anos 90
- Pixelização



# //Engenharia Software



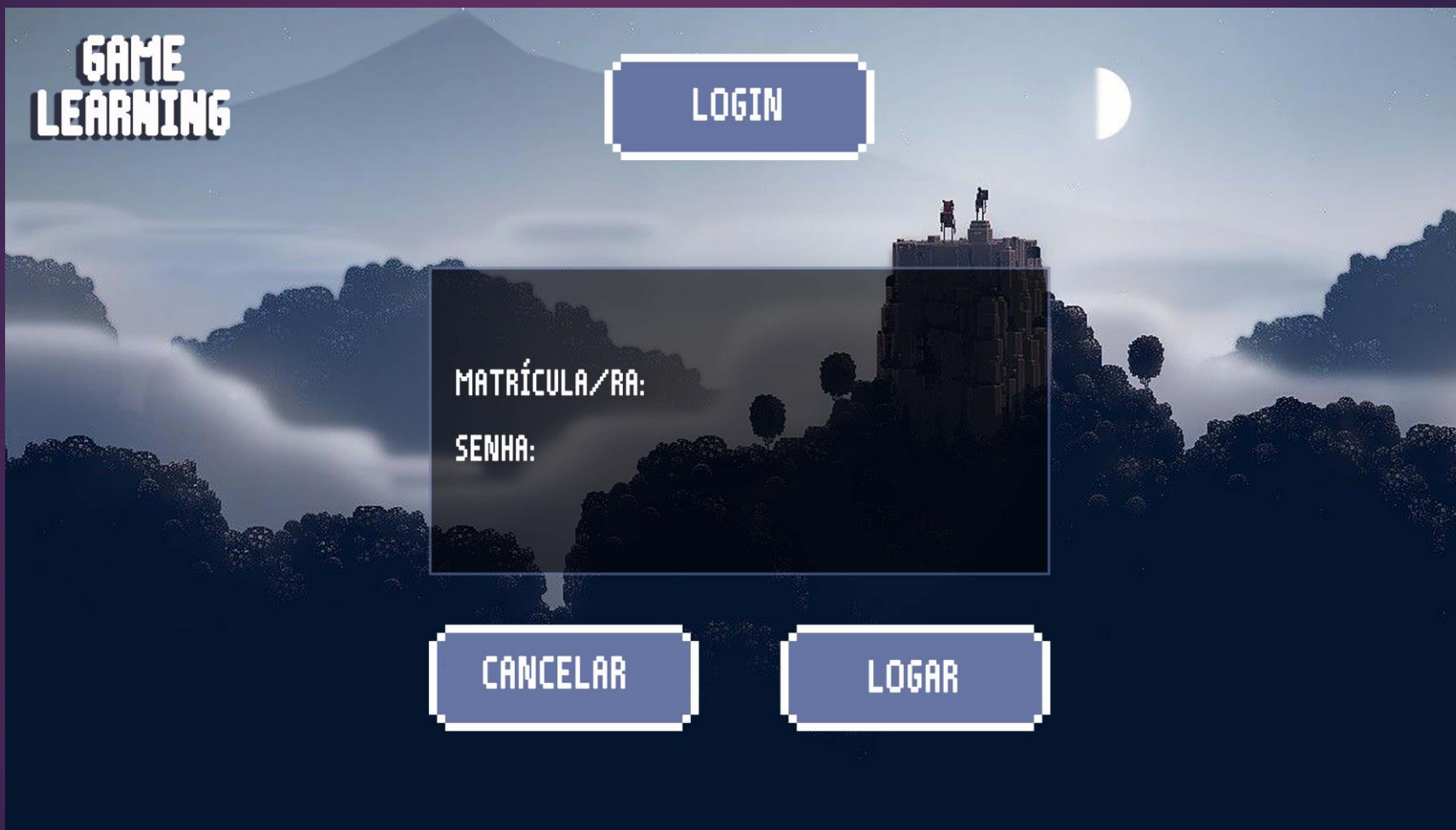
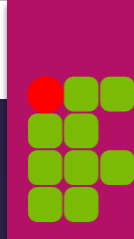
## GAME LEARNING

CADASTRE-SE

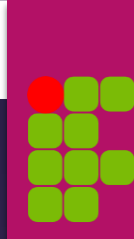
LOGIN

SOBRE

# //Engenharia Software



# //Engenharia Software



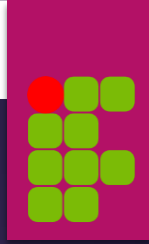
GAME  
LEARNING

SOBRE

TEXTO EXPLICANDO SOBRE O PROGRAMA

VOLTAR

# // PROGRAMAÇÃO



## Classes

- Aluno.cs
- App.config
- Conexao.cs
- Exercicio.cs
- Jogo.cs
- packages.config
- Professor.cs
- Program.cs
- Turma.cs
- Usuario.cs

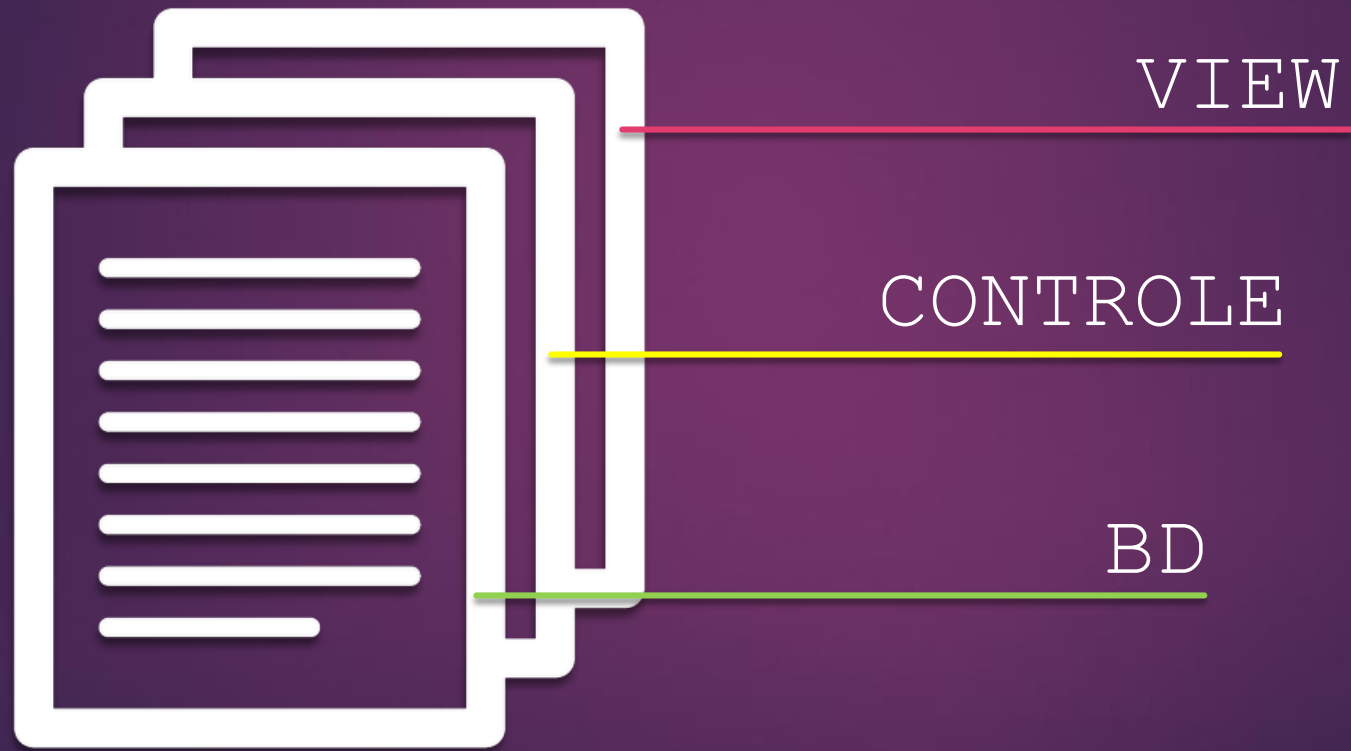
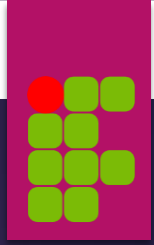
## Resources

- Resources
  - algoRace.jpg
  - algoRacelcon.png
  - AlgoRaceRanking.jpg
  - btn\_login.png
  - btnCadastro.png
  - btnCancelar.png
  - btnCriar.png
  - btnEnviar.png
  - btnExcluir.png
  - btnExerc.png
  - btnInversoCadastro.png
  - btnInversoCancelar.png
  - btnInversoEnviar.png
  - btnInversoLogin.png

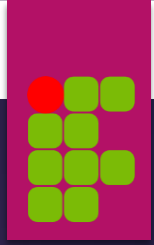
## Telas

- telaAlunosTurma.cs
- telaCadastro.cs
- telaCriarExercicio.cs
- telaInicial.cs
- telaJogos.cs
- telaLogadoAluno.cs
- telaLogadoProfessor.cs
- telaLogin.cs
- telaManterExercicios.cs
- TelaManterTurma.cs
- telaPontuacao.cs
- telaPrevialogo.cs
- telaSobre.cs

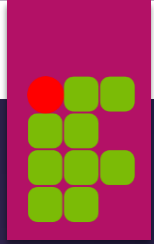
# // PROGRAMAÇÃO



# // PROGRAMAÇÃO



# //BANCO DE DADOS



EXEMPLO DENTRO DO VISUAL STUDIO

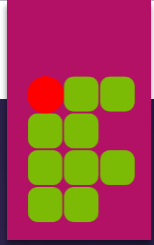
```
public string getIDAluno(string cod_user)
{
    CRUD = "select cod_aluno from aluno where usuario = '" + cod_user + "'";
    resposta = conectarConsultarDesconectar();
    return resposta;
}

public void deletaAluno(string cod_user)
{
    CRUD = "delete FROM aluno where usuario = '" + cod_user + "'";
    resposta = conectarInserirDesconectar();
}

#endregion
```



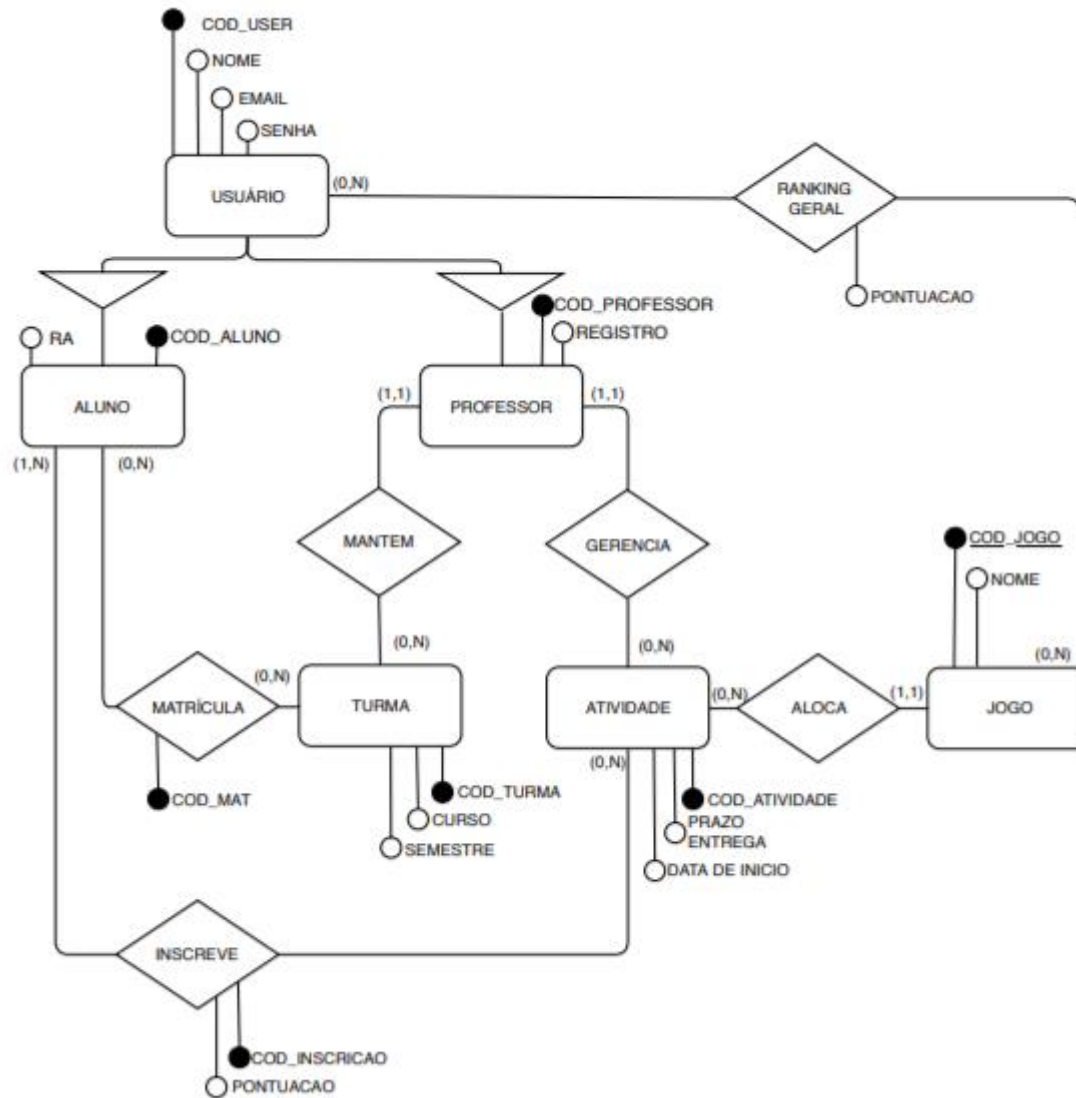
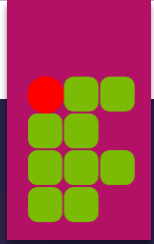
# // BANCO DE DADOS



EXEMPLO DE CRIAÇÃO DE TABELA

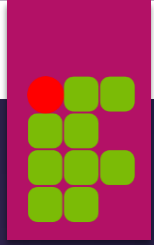
```
57 CREATE TABLE atividade(  
58     cod_atividade serial primary key,  
59     prazo_atividade timestamp without time zone not null,  
60     inicio_atividade timestamp without time zone not null,  
61     Prof integer not null,  
62     Jogo integer not null,  
63     CONSTRAINT ch_estr_de_professor_em_atividade FOREIGN KEY(Prof)  
64     REFERENCES professor(cod_professor) MATCH SIMPLE,  
65     CONSTRAINT ch_estr_de_jogo_em_atividade FOREIGN KEY(Jogo)  
66     REFERENCES jogo(cod_jogo) MATCH SIMPLE  
67 );
```





# // BANCO DE DADOS

DER



//DEMO