**Squad 3 Sonserina**:



**Nome do projeto:** Serviço de Streaming - Maryflix

**Objetivos do projeto:**

* Criar um banco de dados relacional contendo as informações de usuários(clientes) com os contratos, planos e pagamentos relacionados a esse usuário.
* Incluir as informações/histórico de avaliações e visualizações de cada usuário relacionadas a um título específico, que também estão incluídos na base de dados.

**Integrantes:**  
Gregorio Leite Gaio [gaiogreg@msu.edu](mailto:gaiogreg@msu.edu)

Guilherme César de Lima Nascimento [guilherme.cl.nascimento@gmail.com](mailto:guilherme.cl.nascimento@gmail.com)

João Vitor Dias Ferraz [diasferraz94@gmail.com](mailto:diasferraz94@gmail.com)

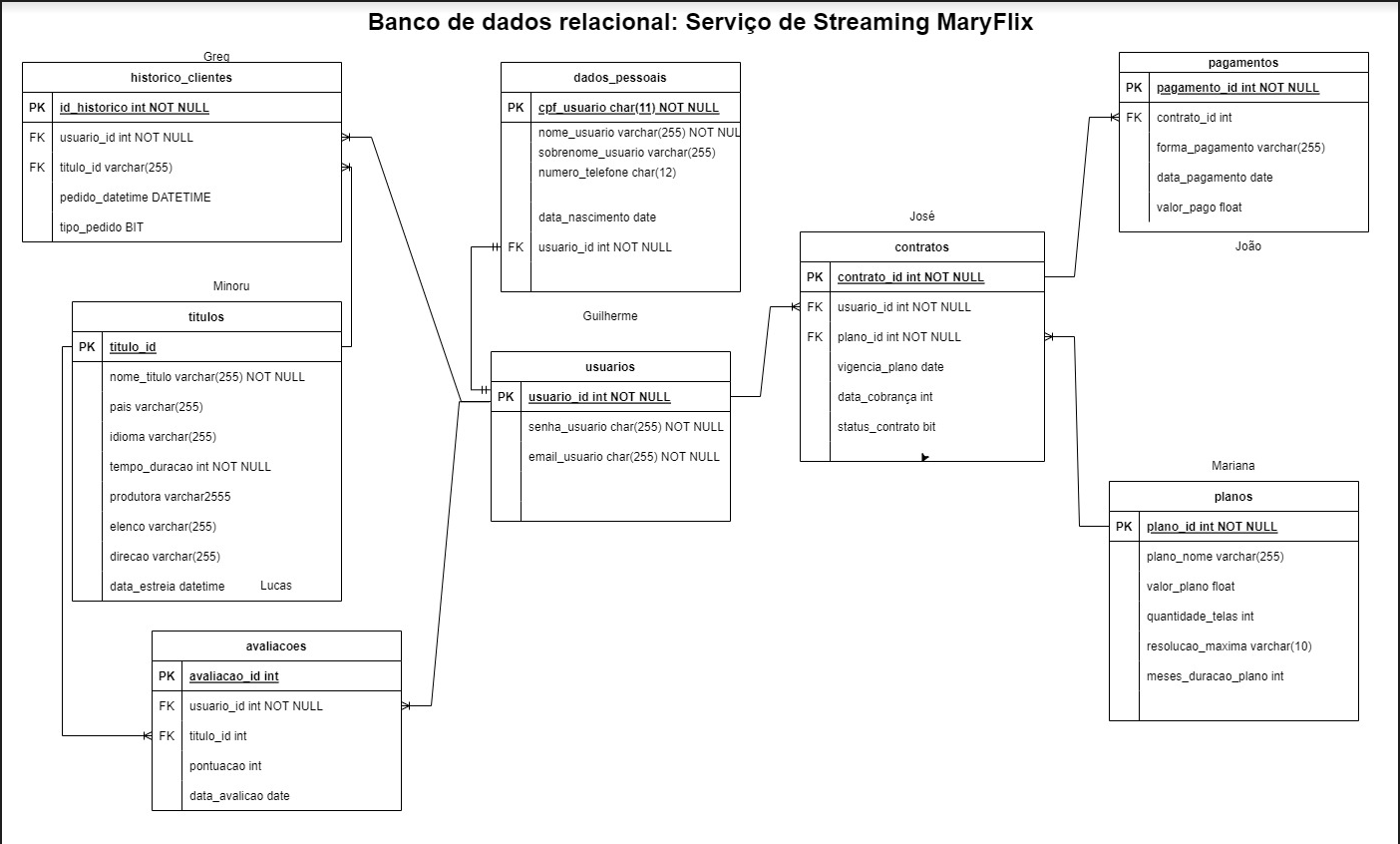
José Rafael Souza do Nascimento [jorafaelnascimento@gmail.com](mailto:jorafaelnascimento@gmail.com)

Lucas Medeiros da Silva [lucasmedesilva@gmail.com](mailto:lucasmedesilva@gmail.com)

Mariana Moutinho Nomura [mariana.moutinho13@gmail.com](mailto:mariana.moutinho13@gmail.com)

Rodrigo Minoru Hayakawa Tanaka [rodrigohayakawa@gmail.com](mailto:rodrigohayakawa@gmail.com)

Diagrama: [DB MARYFLIX](https://drive.google.com/file/d/1nUDGP9MYgNvgEJdwH2xGsGNhJz45elgs/view?usp=sharing)



**CRIAÇÃO DAS TABELAS DO BANCO DE DADOS**

CREATE DATABASE maryflixdb

USE maryflixdb

CREATE TABLE usuarios (

usuario\_id INT IDENTITY (1,1) PRIMARY KEY,

senha\_usuario VARCHAR(50) NOT NULL,

email\_usuario VARCHAR(50) NOT NULL,

);

CREATE TABLE planos(

plano\_id INT IDENTITY (1,1) PRIMARY KEY,

plano\_nome VARCHAR(15) NOT NULL,

valor\_plano FLOAT NOT NULL,

quantidade\_telas INT NOT NULL,

resolucao\_maxima VARCHAR(10) NOT NULL,

meses\_duracao\_plano INT NOT NULL

);

CREATE TABLE contratos(

contrato\_id INT IDENTITY (1,1) PRIMARY KEY,

usuario\_id INT FOREIGN KEY REFERENCES usuarios(usuario\_id),

plano\_id INT FOREIGN KEY REFERENCES planos(plano\_id),

vigencia\_plano DATE,

data\_cobranca INT,

status\_contrato BIT

);

CREATE TABLE pagamentos(

pagamento\_id INT IDENTITY (1,1) PRIMARY KEY,

contrato\_id INT FOREIGN KEY REFERENCES contratos(contrato\_id),

forma\_pagamento VARCHAR(50) NOT NULL,

data\_pagamento DATETIME NOT NULL,

valor\_pago FLOAT NOT NULL

);

CREATE TABLE titulos(

titulo\_id INT IDENTITY (1,1) PRIMARY KEY,

nome\_titulo VARCHAR(255) NOT NULL,

tempo\_duracao INT,

elenco VARCHAR(255),

direcao VARCHAR(255),

pais VARCHAR(255),

produtora VARCHAR(255),

idioma VARCHAR(255),

data\_estreia DATE

);

CREATE TABLE historico\_clientes (

id\_historico INT IDENTITY (1,1) PRIMARY KEY,

usuario\_id INT FOREIGN KEY REFERENCES usuarios(usuario\_id),

titulo\_id INT FOREIGN KEY REFERENCES titulos(titulo\_id),

pedido\_datetime DATETIME,

tipo\_pedido BIT

);

CREATE TABLE avaliacoes(

avaliacao\_id INT IDENTITY (1,1) PRIMARY KEY,

usuario\_id INT FOREIGN KEY REFERENCES usuarios(usuario\_id),

titulo\_id INT FOREIGN KEY REFERENCES titulos(titulo\_id),

pontuacao INT,

data\_avaliacao DATETIME

);

CREATE TABLE dados\_pessoais (

cpf\_usuario CHAR(11) PRIMARY KEY,

nome\_usuario VARCHAR(50) NOT NULL,

sobrenome\_usuario VARCHAR(50) NOT NULL,

numero\_telefone CHAR(12),

data\_nascimento DATE,

usuario\_id INT FOREIGN KEY REFERENCES usuarios(usuario\_id)

);

**-- POPULANDO O BANCO DE DADOS**

INSERT INTO usuarios (senha\_usuario, email\_usuario)

VALUES

('banana123', 'mariana\_banana@gmail.com'),

('abacaxi321', 'greg\_abacaxi@gmail.com'),

('arrozcomfeijao123', 'joao\_pe\_de\_feijao@gmail.com'),

('joaquina123', 'maria\_joaquina@gmail.com'),

('avassalador321', 'avassaladores@gmail.com'),

('guigojunior123', 'guigo\_junior@gmail.com'),

('ademar456', 'ademar\_silva@hotmail..com'),

('rogerio654', 'rogerio\_carlos@outlook.com'),

('andressa123', 'andressa\_mor@outlook.com'),

('marquinhos789', 'marcos\_2@outlook.com'),

('larissa321', 'lari.amorzinho@outlook.com'),

('roger654', 'roger@outlook.com'),

('maria!123', 'maria23@gmail.com'),

('senhaaa', 'felipe\_assis@gmail.com'),

('danilosoares15', 'danilo@gmail.com');

INSERT INTO dados\_pessoais(cpf\_usuario, nome\_usuario, sobrenome\_usuario, numero\_telefone, data\_nascimento, usuario\_id)

VALUES (12345678901, 'Mariana', 'Moutinho', '011970707070', '1994-05-20', 1),

(23456789012, 'Greg', 'Gaio', '019970707070', '1997-04-09', 2),

(98765432109, 'Joao', 'Ferraz', '011987654321', '1994-06-26', 3),

(55522233388, 'Maria', 'Joaquina', '011965778924', '1990-10-27', 4),

(23456787777, 'Ava', 'Salador', '019966332244', '1999-04-09', 5),

(98765432559, 'Guilherme', 'Santos', '011987658888', '1993-07-16', 6),

(15258147931, 'Ademar', 'Guia', '032854784736', '1987-02-24', 7),

(13579792837, 'Rogerio', 'Miranda', '022237685746', '1985-07-11', 8),

(66665432109, 'Andressa', 'Lira', '011987645622', '1990-12-01', 9),

(48574629381, 'Marcos', 'Lima', '051987485746', '1981-02-23', 10),

(85730485731, 'Larissa', 'Manoela', '031976475648', '1975-12-05', 11),

(74296049872, 'Roger', 'Guedes', '081912385736', '1981-09-23', 12),

(15384756931, 'Maria', 'Lucia', '021943702270', '1992-03-11', 13),

(58999999837, 'Felipe', 'Assis', '041943723070', '1990-02-23', 14),

(13748577772, 'Danilo', 'Soares', '018985543321', '1996-11-10', 15);

INSERT INTO titulos(nome\_titulo, tempo\_duracao, elenco, direcao, pais, produtora, idioma, data\_estreia)

VALUES

('Harry Potter and the Philosopher''s Stone', 152, 'Daniel Radcliffe, Rupert Grint, Emma Watson, John Cleese, Robbie Coltrane, Warwick Davis', 'Chris Columbus', 'United Kingdom, United States', 'Warner Bros. Pictures', 'English', '2001-11-04'),

('Harry Potter and the Chamber of Secrets', 161, 'Daniel Radcliffe, Rupert Grint, Emma Watson, Kenneth Branagh, John Cleese, Robbie Coltrane', 'Chris Columbus', 'United Kingdom, United States', 'Warner Bros. Pictures', 'English', '2002-11-03'),

('Harry Potter and the Prisoner of Azkaban', 142, 'Daniel Radcliffe, Rupert Grint, Emma Watson, Robbie Coltrane, Michael Gambon, Richard Griffiths', 'Alfonso Cuarón', 'United Kingdom, United States', 'Warner Bros. Pictures', 'English', '2004-05-23'),

('Harry Potter and the Goblet of Fire', 157, 'Daniel Radcliffe, Rupert Grint, Emma Watson, Robbie Coltrane, Ralph Fiennes, Michael Gambon', 'Mike Newell', 'United Kingdom, United States', 'Warner Bros. Pictures', 'English', '2005-11-06'),

('Harry Potter and the Order of the Phoenix', 138, 'Daniel Radcliffe, Rupert Grint, Emma Watson, Helena Bonham Carter, Robbie Coltrane, Warwick Davis', 'David Yates', 'United Kingdom, United States', 'Warner Bros. Pictures', 'English', '2007-06-28'),

('Harry Potter and the Half-Blood Prince', 153, 'Daniel Radcliffe, Rupert Grint, Emma Watson, Jim Broadbent, Helena Bonham Carter, Robbie Coltrane', 'David Yates', 'United Kingdom, United States', 'Warner Bros. Pictures', 'English', '2009-07-07'),

('Harry Potter and the Deathly Hallows – Part 1', 146, 'Daniel Radcliffe, Rupert Grint, Emma Watson, Helena Bonham Carter, Robbie Coltrane, Warwick Davis', 'David Yates', 'United Kingdom, United States', 'Warner Bros. Pictures', 'English', '2010-11-11'),

('Harry Potter and the Deathly Hallows – Part 2', 130, 'Daniel Radcliffe, Rupert Grint, Emma Watson, Helena Bonham Carter, Robbie Coltrane, Warwick Davis', 'David Yates', 'United Kingdom, United States', 'Warner Bros. Pictures', 'English', '2011-07-07');

INSERT INTO avaliacoes(usuario\_id, titulo\_id, pontuacao, data\_avaliacao)

VALUES

(1, 1, 2, '2022-08-16 17:23:15')

,(1, 3, 4, '2022-08-16 20:14:11')

,(2, 1, 3, '2022-04-16 10:30:23')

,(2, 2, 4, '2022-02-22 14:10:50')

,(3, 1, 4, '2021-09-10 12:40:25')

,(3, 2, 4, '2020-10-22 14:55:08')

;

INSERT INTO historico\_clientes (usuario\_id, titulo\_id, pedido\_datetime, tipo\_pedido)

VALUES

(1, 1, '2022-08-16 09:25:02', 1),

(1, 1, '2022-08-16 12:17:01', 0),

(2, 2, '2022-02-22 07:10:10', 1),

(2, 2, '2022-02-22 09:20:02', 0),

(3, 1, '2021-09-10 10:00:02', 1),

(3, 1, '2021-09-10 12:25:10', 0);

INSERT INTO planos(plano\_nome, valor\_plano, quantidade\_telas, resolucao\_maxima, meses\_duracao\_plano)

VALUES

('BASICO MENSAL', 15.99, 1, 'FHD', 1),

('BASICO ANUAL', 149.99, 1, 'FHD', 12),

('PREMIUM MENSAL', 39.99, 4, '4K', 1),

('PREMIUM ANUAL', 349.99, 4, '4K', 12);

INSERT INTO contratos(usuario\_id, plano\_id, vigencia\_plano, data\_cobranca, status\_contrato)

VALUES

(1, 1, '2022-09-16', 5, 1),

(2, 3, '2022-09-01', 15, 1),

(3, 4, '2023-02-01', 10, 1),

(3, 1, '2022-01-29', 10, 0),

(4, 3, '2022-12-25', 15, 1),

(5, 2, '2023-08-10', 12, 1),

(6, 1, '2022-09-10', 12, 1),

(7, 4, '2023-07-10', 7, 1),

(8, 3,'2022-12-10', 10, 1),

(9, 3,'2023-07-13' , 05, 1),

(10, 4,'2022-08-22' , 25, 1),

(11, 2,'2023-08-11' , 20, 1),

(12, 3, '2022-09-12', 10, 1),

(13, 4, '2023-05-23', 15, 1),

(14, 2, '2022-09-06', 10, 1),

(15, 4, '2023-04-11', 05, 1);

INSERT INTO pagamentos(contrato\_id, forma\_pagamento, data\_pagamento, valor\_pago)

VALUES

(1, 'BOLETO', '2022-08-05', 15.99),

(2, 'DEBITO AUTOMATICO','2022-07-15', 39.99),

(3, 'BOLETO','2022-02-13', 366.66),

(4, 'PIX', '2022-01-10', 15.99),

(5, 'DEBITO AUTOMATICO', '2022-06-12', 15.99),

(6, 'PIX','2023-09-03', 50.00),

(7, 'BOLETO','2022-02-13', 367.66),

(8, 'PIX', '2022-07-07', 349.99),

(9, 'BOLETO','2022-11-10', 39.99),

(10, 'BOLETO','2022-08-05', 39.99),

(11, 'BOLETO', '2022-05-08', 15.99),

(12, 'CARTAO DE CREDITO', '2022-12-08', 149.99),

(13, 'DEBITO AUTOMATICO', '2022-10-10', 39.99),

(14, 'DEBITO AUTOMATICO', '2022-08-25', 349.99),

(15, 'PIX','2022-06-11', 150.00),

(16, 'BOLETO','2022-05-13', 360.00);

**POPULAÇÃO EXTRA**

**Avaliações:** [avaliacoes.sql](https://drive.google.com/file/d/1e-f6tZ8V7bxieWA3NeEEMiAqAqvIAGHf/view?usp=sharing)

**Lista de filmes:** [lista\_filmes\_MTVMA.sql](https://drive.google.com/file/d/18dxXICAfn5GOsGJsSuMlhJ1deqv_rCLJ/view?usp=sharing)

**Visualizações:** [visualizacoes.sql](https://drive.google.com/file/d/10IHxypPVXVwIYBoOT6vYCf6ZhLq_QA7C/view?usp=sharing)

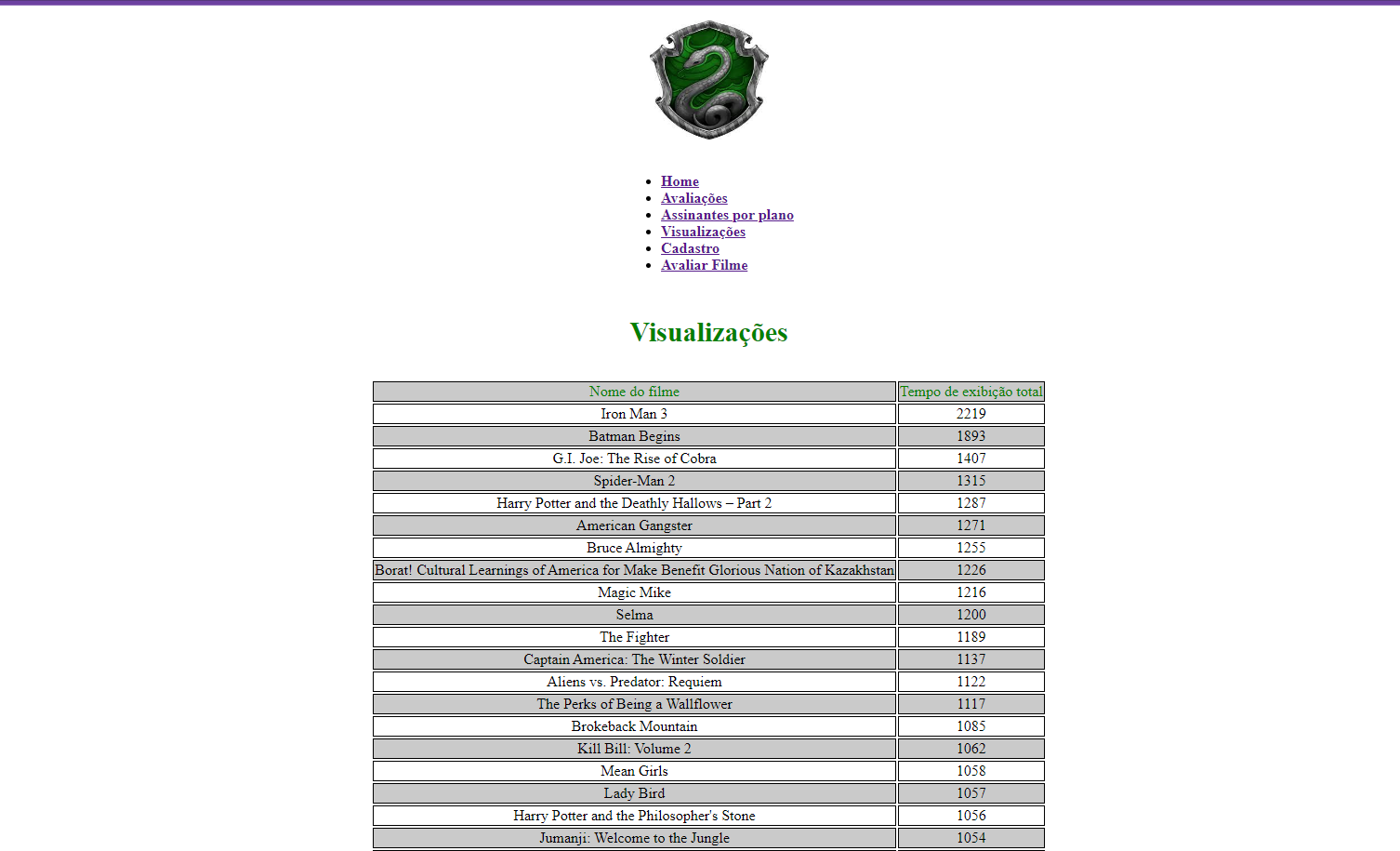
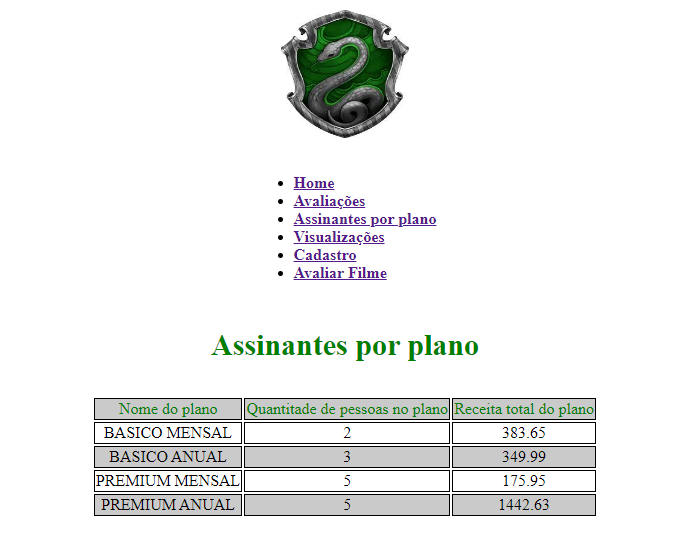
**APLICAÇÃO**

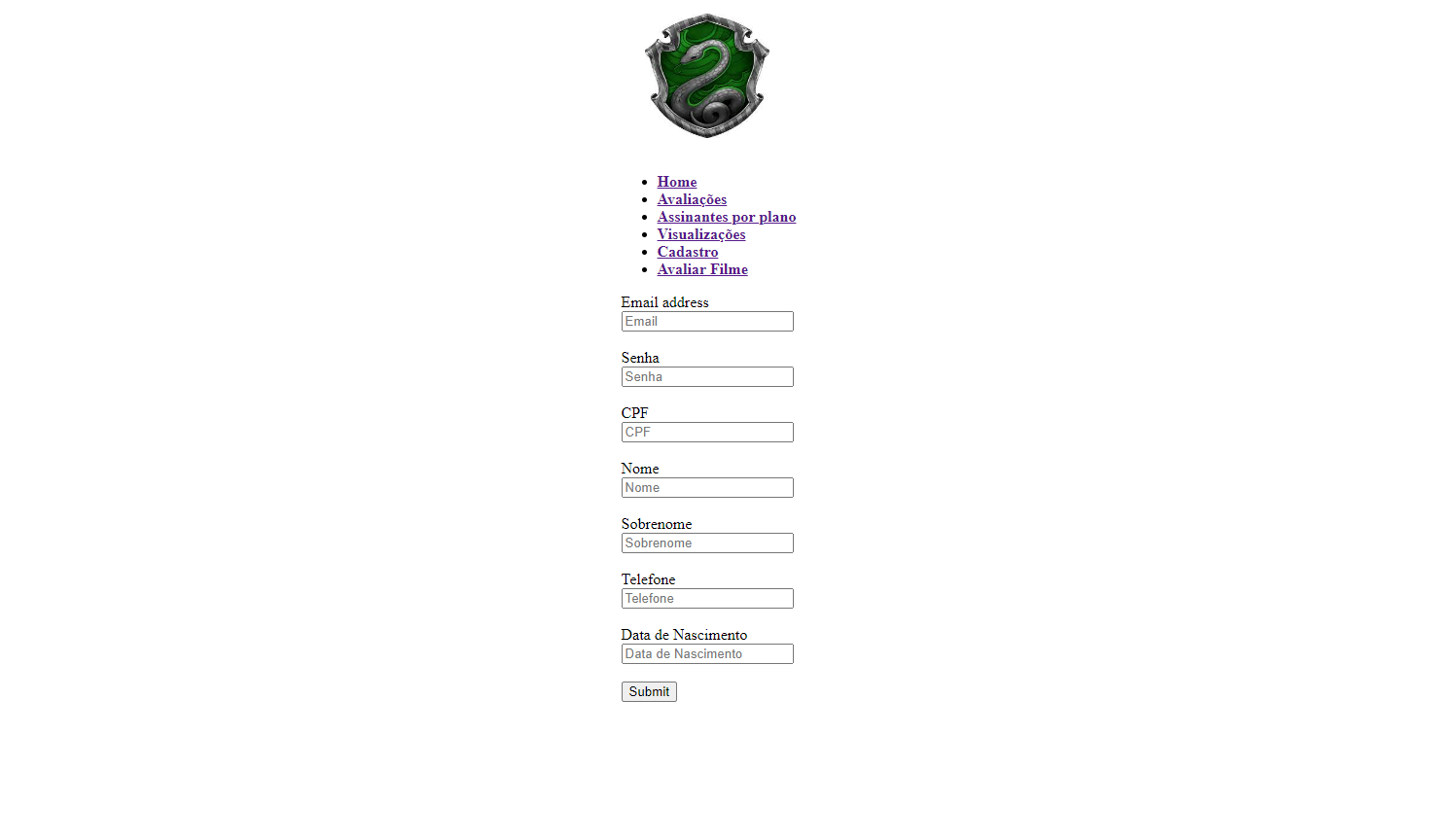
CREATE PROCEDURE cadastra\_usuario @email VARCHAR(50), @senha VARCHAR(50), @cpf\_usuario CHAR(11), @nome\_usuario VARCHAR(50), @sobrenome\_usuario VARCHAR(50), @numero\_telefone CHAR(12), @data\_nascimento DATE, @usuario\_id INT AS INSERT INTO usuarios(senha\_usuario,email\_usuario) VALUES(@senha,@email);

SET @usuario\_id = SCOPE\_IDENTITY(); INSERT INTO dados\_pessoais(cpf\_usuario,nome\_usuario,sobrenome\_usuario,numero\_telefone,data\_nascimento,usuario\_id) VALUES(@cpf\_usuario,@nome\_usuario,@sobrenome\_usuario,@numero\_telefone,@data\_nascimento,@usuario\_id);

GO

Código da aplicação: <https://github.com/joaovdferraz/Projeto5>





**PESQUISAS REALIZADAS**

**Objetivo do exercício:** exemplos de queries que fazem sentido para o negócio.

* **Título e nota média das avaliações em ordem decrescente.**

SELECT nome\_titulo, CAST(AVG(CAST(pontuacao AS DECIMAL(10,2)))AS DECIMAL(10,2)) as media\_avaliacao

FROM titulos

INNER JOIN avaliacoes

ON titulos.titulo\_id = avaliacoes.titulo\_id

GROUP BY nome\_titulo

ORDER BY media\_avaliacao DESC

* **Diretor, título do filme e avaliação média do filme mais bem avaliado.**

SELECT TOP(1) direcao, nome\_titulo, AVG(pontuacao) as media\_avaliacao

FROM titulos

INNER JOIN avaliacoes

ON titulos.titulo\_id = avaliacoes.titulo\_id

GROUP BY direcao , nome\_titulo

* **Usuarios que pagaram multa por atraso.**

SELECT usuario\_id

FROM contratos

INNER JOIN pagamentos

ON pagamentos.contrato\_id = contratos.contrato id

WHERE valor\_pago NOT IN (15.99,149.99,39.99,349.99)

* **Usuário que fez o pagamento mais recente.**

SELECT TOP 1 data\_pagamento, dados\_pessoais.nome\_usuario,dados\_pessoais.sobrenome\_usuario,

usuarios.email\_usuario from pagamentos

INNER JOIN contratos

ON pagamentos.contrato\_id = contratos.contrato\_id

INNER JOIN usuarios

ON contratos.usuario\_id = usuarios.usuario\_id

INNER JOIN dados\_pessoais

ON usuarios.usuario\_id = dados\_pessoais.usuario\_id

ORDER BY data\_pagamento DESC

* **Usuário ativo com mais tempo de contrato com a Maryflix.**

SELECT top 1 usuario\_id, sum(meses\_duracao\_plano) as tempo\_contratado

FROM contratos inner join planos on contratos.plano\_id = planos.plano\_id

WHERE status\_contrato = 1 group by usuario\_id order by tempo\_contratado desc

* **Usuário com maior número de renovações de contratos mensais.**

SELECT email\_usuario, COUNT(email\_usuario) AS quant\_contratos FROM usuarios

INNER JOIN contratos ON usuarios.usuario\_id = contratos.usuario\_id

WHERE status\_contrato = 1

GROUP BY email\_usuario

ORDER BY quant\_contratos DESC;

* **Quais/quantos usuários do plano básico avaliaram o filme Harry Potter e a Pedra Filosofal (“Harry Potter and the Philosopher's Stone”).**

SELECT u.usuario\_id, t.nome\_titulo, p.plano\_nome, a.avaliacao\_id

FROM usuarios as u

INNER JOIN contratos as c

ON u.usuario\_id = c.usuario\_id

INNER JOIN planos as p

ON c.plano\_id = p.plano\_id

INNER JOIN avaliacoes as a

ON a.usuario\_id = u.usuario\_id

INNER JOIN titulos as t

ON a.titulo\_id = t.titulo\_id

WHERE p.plano\_id = 1 AND t.nome\_titulo = 'Harry Potter and the Philosopher''s Stone'

* **Títulos ordenados por tempo de exibição.**

SELECT t.titulo\_id, t.nome\_titulo, SUM(i.intervalo) AS tempo\_exibicao

FROM (SELECT h1.id\_historico, h1.usuario\_id, h1.titulo\_id, h1.pedido\_datetime AS hora\_inicio, h2.pedido\_datetime AS hora\_fim, DATEDIFF(MINUTE, h1.pedido\_datetime, h2.pedido\_datetime) AS intervalo

FROM historico\_clientes AS h1

INNER JOIN historico\_clientes AS h2

ON h1.usuario\_id = h2.usuario\_id AND h1.titulo\_id = h2.titulo\_id

WHERE h1.tipo\_pedido = 1 AND h2.tipo\_pedido = 0) AS i

JOIN titulos AS t

ON i.titulo\_id = t.titulo\_id

GROUP BY t.titulo\_id, t.nome\_titulo

ORDER BY tempo\_exibicao DESC

* **Quantos assinantes ativos temos por plano e a respectiva receita.**

SELECT plano\_nome, COUNT(DISTINCT (contratos.contrato\_id)) AS assinantes\_por\_plano,

SUM(valor\_pago) AS receita\_total\_por\_plano

FROM contratos

INNER JOIN planos

ON contratos.plano\_id = planos.plano\_id

INNER JOIN pagamentos

ON contratos.contrato\_id = pagamentos.contrato\_id

WHERE status\_contrato = 1

GROUP BY planos.plano\_nome

**CASE/VIEW**

**Objetivo do exercício:** usar um select case e subselects para criar uma view para o projeto.

CREATE VIEW consulta\_multa AS

SELECT pagamento\_id, valor\_pago, valor\_plano, CASE

WHEN valor\_pago > valor\_plano

THEN 'ATRASADO'

ELSE 'EM DIA'

END AS tipo\_pagamento

FROM pagamentos

INNER JOIN contratos

ON pagamentos.contrato\_id = contratos.contrato\_id

INNER JOIN planos

ON planos.plano\_id = contratos.plano\_id;

SELECT \* FROM consulta\_multa;

**HIGHLIGHTS DO PROJETO**

* Introdução: Diagrama (MaryFlix) - Guilherme
* A população por *scraping* (*pulchra pulmenti*) - Rodrigo Minoru
* Nossa aplicação/HTML - João