

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE CAMPINAS
ESCOLA POLITÉCNICA
ENGENHARIA DE SOFTWARE

JEAN YUKI KIMURA (24008214)
JHENIFER LAÍS BARBOSA (24014979)
JOÃO PEDRO DUARTE GIATTI (24019083)
LAYRA ANGELI CAMPIOL (24015714)

TRABALHO DE MODELAGEM REFERENTE AO PROJETO INTEGRADOR

CAMPINAS
2024

Apresentação do Descritivo do Projeto

Este projeto consiste no desenvolvimento de uma plataforma digital de apostas, onde usuários poderão criar, participar e gerenciar apostas sobre eventos futuros em diversas áreas, como esportes, eleições e outros acontecimentos de interesse. A aplicação permitirá que os participantes criem eventos, façam apostas, gerenciem suas carteiras virtuais e acompanhem os resultados, tudo de maneira interativa e segura. A plataforma contará com funcionalidades de cadastro, login, aprovação de eventos por moderadores e a realização de transações financeiras simuladas para apostas.

Os principais objetivos do grupo com este trabalho são:

- Desenvolver uma plataforma funcional e segura que permita aos usuários interagirem de maneira eficiente, criando e participando de eventos de apostas.
- Garantir uma experiência de usuário intuitiva, com navegação simples e transparente, para facilitar a realização de apostas e o gerenciamento da carteira digital.
- Implementar um sistema de moderação eficaz, permitindo que os moderadores aprovem ou rejeitem eventos com base em critérios definidos.
- Assegurar a integridade das transações financeiras, realizando um gerenciamento eficiente das apostas e a distribuição dos prêmios de forma justa.

Esse trabalho visa oferecer uma aplicação completa e segura para apostas em eventos futuros, atendendo às necessidades de seus usuários com eficiência e transparência.

Liste os Requisitos Funcionais

[RF001] Cadastro de Usuários

Descrição: O sistema permitirá que novos usuários se cadastrem para criar e participar de eventos.

Entradas e Pré-Condições:

1. O usuário deverá informar os seguintes dados obrigatórios:

- Nome.
- E-mail.
- Senha.
- Data de nascimento.

Saídas e Pós-Condições:

- O sistema salvará os dados do usuário.
 - O acesso à conta será permitido após o login bem-sucedido.
-

[RF002] Login

Descrição: O sistema permitirá que usuários cadastrados realizem login para acessar suas contas.

Entradas e Pré-Condições:

1. O usuário deverá informar:

- E-mail.
- Senha válida.

Saídas e Pós-Condições:

- O sistema autenticará o usuário.
 - Um token de sessão será gerado, permitindo o acesso às funcionalidades da conta.
-

[RF003] Criação de Eventos

Descrição: Usuários autenticados poderão criar eventos para receber apostas.

Entradas e Pré-Condições:

1. O usuário deverá informar:

- Título do evento (máximo 50 caracteres).
- Descrição do evento (máximo 150 caracteres).
- Valor mínimo da cota (R\$ 1,00).
- Período de apostas (data e hora de início e fim).
- Data do evento (dia, mês, ano).
- O usuário deve estar autenticado.

Saídas e Pós-Condições:

- O evento será salvo no sistema.
 - O evento será enviado para aprovação de um moderador.
-

[RF004] Aprovação de Eventos

Descrição: Moderadores poderão aprovar ou rejeitar novos eventos cadastrados.

Entradas e Pré-Condições:

1. O moderador deverá:

- Selecionar o evento.
- Indicar a aprovação ou rejeição.
- Fornecer uma justificativa em caso de rejeição.

Saídas e Pós-Condições:

- O sistema notificará o criador do evento via e-mail informando o status da aprovação.
-

[RF005] Apostas em Eventos

Descrição: Usuários autenticados poderão apostar em eventos aprovados.

Entradas e Pré-Condições:

1. O usuário deverá:

- O usuário deverá:
- Selecionar o Evento
- Escolhe a opção ("Sim" ou "Não")

2. O usuário deve ter saldo suficiente na carteira.

Saídas e Pós-Condições:

- O valor apostado será debitado da carteira do usuário.\
-

RF006] Finalização de Eventos

Descrição: Moderadores encerrarão eventos indicando se o resultado foi alcançado ou não.

Entradas e Pré-Condições:

1. O moderador deverá:

- Indicar se o evento ocorreu ("Sim" ou "Não").

Saídas e Pós-Condições:

- O sistema distribuirá proporcionalmente os valores das apostas vencedoras.
-

[RF007] Gerenciamento de Carteira

Descrição: O sistema permitirá que usuários adicionem ou saquem fundos da carteira virtual.

Entradas e Pré-Condições:

1. Adicionar Fundos:

- Escolher o valor em reais
- Simular pagamento por cartão de crédito.

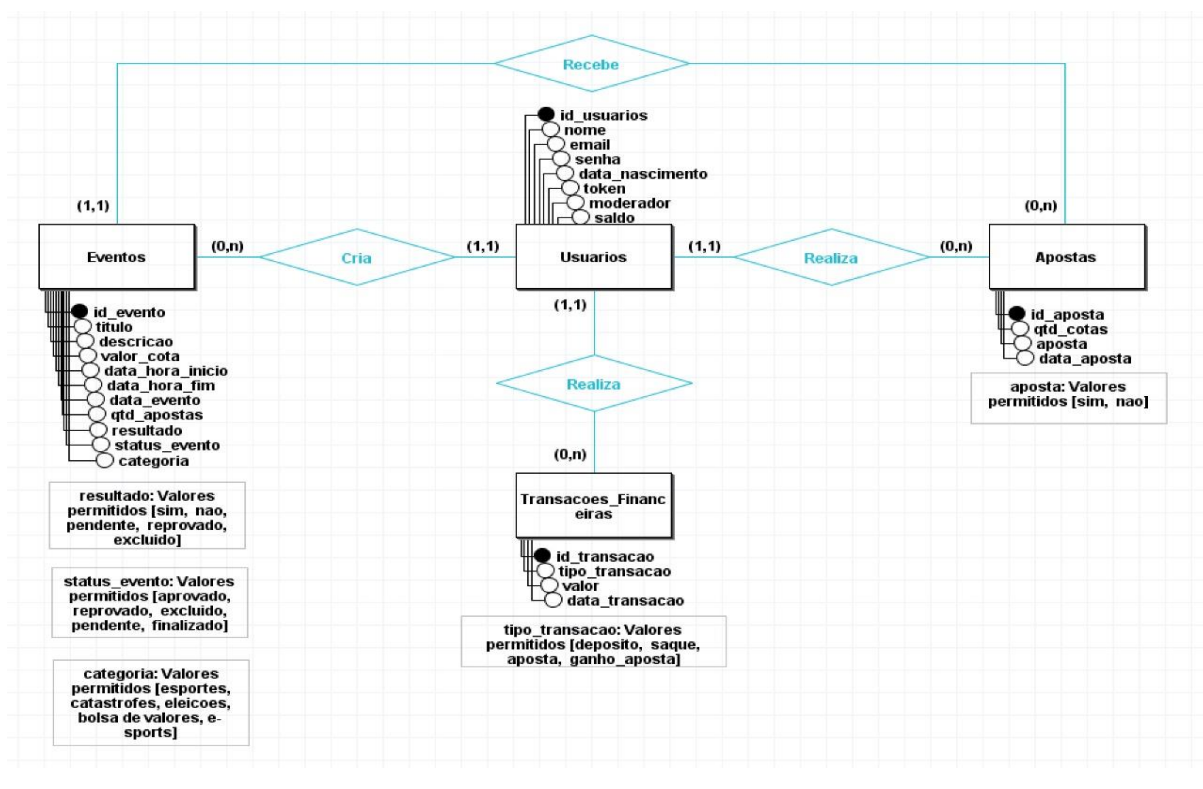
2. Sacar Fundos

- Informar os dados necessários:
- Banco.
- Agência.
- Conta corrente ou chave PIX.

Saídas e Pós-Condições:

- O sistema atualizará o saldo da carteira.
- Aplicará taxas de saque conforme tabela definida.

Modelagem do Sistema – Modelo Conceitual (MER)



Modelagem do Sistema - Modelo Lógico

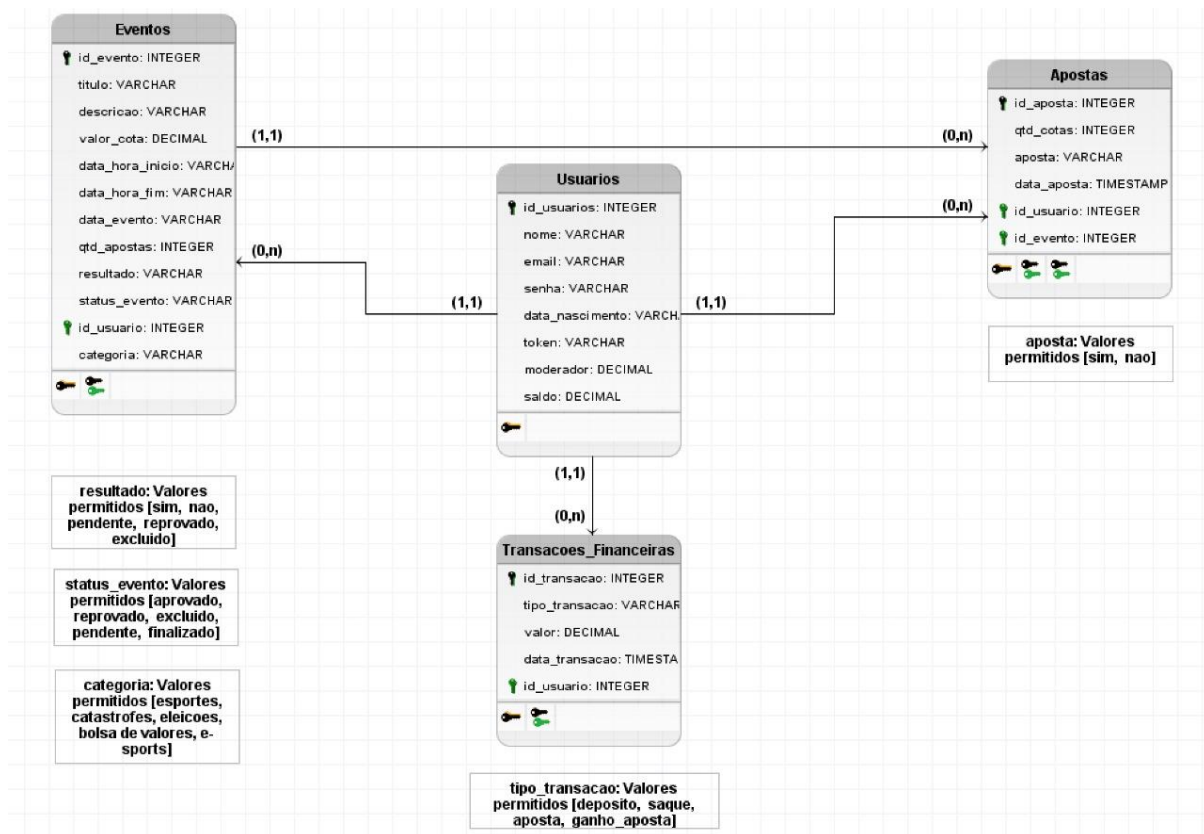


Tabela	Relacionamento	Nome do Relacionamento	Descrição
tbl_Dicionário de Dados usuarios	tbl_apostas	Realiza	Tabela para usuário realizar cadastro, criar eventos e realizar apostas
	tbl_transacoes_financeiras	Realiza	
	tbl_eventos	Cria	
tbl_transacoes_financeiras	tbl_usuarios	Realiza	Tabela para registrar transações financeiras dos usuários
tbl_eventos	tbl_apostas	Recebe	Tabela para registrar eventos, cada evento pode receber múltiplas apostas

Tabela	Nome	Tipos de	Comprime	Restriç	Valor	Descrição
--------	------	----------	----------	---------	-------	-----------

	da Coluna	Dados	nto	ões	Padrão	
tbl_usuari o	id_usuari o	inteiro	4 bytes	PK	N/D	Identificador único do usuário
	nome	caracteres	100 bytes	NOT NULL	N/D	Nome completo do usuário
	email	caracteres	150 bytes	UNIQUE, NOT NULL	N/D	Endereço de e- mail do usuário
	senha	caracteres	255 bytes	NOT NULL	N/D	Senha do usuário
	data_nas cimento	caracteres	11 bytes	NOT NULL	N/D	Data de nascimento do usuário
	token	caracteres	32 bytes	NOT NULL	N/D	Token único gerado automaticament e para cada usuário, com o intuito de guardar a sessão
	moderad or	number(1)	1 byte	DEFAU LT	0	Indica se o usuário for moderador (0 para usuário 1 para moderador)
	saldo	number	9 bytes	DEFAU LT	0.00	Saldo da conta do usuário

Tabela	Nome da Coluna	Tipos de Dados	Compriment o	Restriç ões	Valor Padrão	Descrição
tbl_event os	id_evento	inteiro	4 bytes	PK	N/D	Identificador único do evento
	titulo	caractere	50 bytes	NOT	N/D	Título do Evento

		s		NULL		
	descricao	caracteres	150 bytes	NOT NULL	N/D	Descrição do Evento
	valor_cota	number	5 bytes	NOT NULL	N/D	Valor da cota para apostas
	data_hora_inicio	caracteres	22 bytes	NOT NULL	N/D	Período inicial das apostas
	data_hora_fim	caracteres	22 bytes	NOT NULL	N/D	Período final das apostas
	data_evento	caracteres	22 bytes	NOT NULL	N/D	Data em que o evento ocorrerá
	qtd_apostas	inteiro	4 bytes	NOT NULL	0	Quantidade de apostas no evento
	resultado	caracteres	20 bytes	NOT NULL, CHECK	'pendente'	Resultado do evento (sim, nao, pendente, reprovado, excluído)
	status_evento	caracteres	20 bytes	NOT NULL, CHECK	'pendente'	Status atual do evento (aprovado, reprovado, excluido, pendente, finalizado)
	categoria	caracteres	20 bytes	NOT NULL, CHECK	N/D	Categorias do evento ('esportes', 'catastrofes', 'eleicoes', 'bolsa de valores', 'esportes')
	id_usuario	inteiro	4 bytes	FK, NOT NULL	N/D	Nº de identificação dos usuários

Tabela	Nome da Coluna	Tipos de Dados	Comprimento	Restrições	Valor Padrão	Descrição
tbl_transacoes_financeiras	id_transacao	inteiro	4 bytes	PK	N/D	Identificador único para transação
	tipo_transacao	caracteres	20 bytes	NOT NULL, CHECK	N/D	Tipos de transação ('deposito', 'saque', 'aposta', 'ganho_aposta', 'reembolso')
	valor	number		NOT NULL	0	Valor para as transações
	data_transacao	timestamp current timestamp	13 bytes	DEFAULT	N/D	Hora atual no momento da transação
	id_usuario	inteiro	4 bytes	FK, NOT NULL	N/D	Nº de identificação do usuário

Tabela	Nome da Coluna	Tipos de Dados	Comprimento	Restrições	Valor Padrão	Descrição
tbl_apostas	id_aposta	inteiro	4 bytes	PK	N/D	Identificador único para apostas
	qtd_cotas	inteiro	4 bytes	NOT NULL	N/D	Quantidade de cotas apostadas
	aposta	caracteres	3 bytes	NOT NULL, CHECK	N/D	Valores válidos ('sim', 'nao') indicando se o usuário aposta a favor ou contra
	data_aposta	timestamp current timestamp	13 bytes	DEFAULT	N/D	Hora atual no momento da inserção

	id_usuario	inteiro	4 bytes	NOT NULL	N/D	Nº de identificação do usuário
	id_evento	inteiro	4 bytes	NOT NULL	N/D	Nº de identificação do evento

Relacionamento	Tabela 1 - FK	Tabela 2 - PK	Descrição
“Cria”	tbl_eventos	tbl_usuarios	Relacionamento que define qual usuário cria determinado evento
“Realiza”	tbl_apostas	tbl_usuarios	Relacionamento que indica quais apostas são feitas pelos usuários
“Recebe”	tbl_transacoes_financeiras	tbl_usuarios	Relacionamento que conecta transações realizadas por um usuário
“Realiza”	tbl_apostas	tbl_eventos	Relacionamento que vincula as apostas aos eventos relacionados

Projeto de Banco de Dados – Modelo Físico

```

-- Tabela de USUARIOS
CREATE TABLE usuarios (
    id_usuario INT PRIMARY KEY,
    nome VARCHAR2(100) NOT NULL,
    email VARCHAR2(150) UNIQUE NOT NULL,
    senha VARCHAR2(255) NOT NULL,
    data_nascimento VARCHAR2(11) NOT NULL,
    token VARCHAR2(32) NOT NULL,
    moderador NUMBER(1) default 0, -- Valor inicial = 0 para usuario comum e 1 para moderador
    saldo NUMBER default 0 -- Valor inicial da carteira = 0
);

-- Criando sequência para USUARIOS
CREATE SEQUENCE SEQ_USUARIOS START WITH 1 INCREMENT BY 1;

-----

-- Tabela de TRANSACOES_FINANCEIRAS
CREATE TABLE transacoes_financeiras (
    id_transacao INT PRIMARY KEY,
    id_usuario INT NOT NULL, -- FK para a tabela USUARIOS
    tipo_transacao VARCHAR2(20) NOT NULL CHECK (tipo_transacao IN ('deposito', 'saque', 'aposta', 'ganho_aposta', 'reembolso')),
    valor NUMBER NOT NULL,
    data_transacao TIMESTAMP DEFAULT CURRENT_TIMESTAMP -- Hora atual no momento da inserção
);

-- Criando sequência para TRANSACOES_FINANCEIRAS
CREATE SEQUENCE SEQ_TRANSACOES_FINANCEIRAS START WITH 1 INCREMENT BY 1;

-----

-- Tabela de EVENTOS
CREATE TABLE eventos (
    id_evento INT PRIMARY KEY,
    id_usuario INT NOT NULL, -- FK para a tabela USUARIOS
    titulo VARCHAR2(50) NOT NULL,
    descricao VARCHAR2(150) NOT NULL,
    valor_cota NUMBER NOT NULL,
    data_hora_inicio VARCHAR2(22) NOT NULL,
    data_hora_fim VARCHAR2(22) NOT NULL,
    data_evento VARCHAR2(22) NOT NULL,
    qtd_apostas INT NOT NULL,
    resultado VARCHAR2(20) NOT NULL CHECK (resultado IN ('sim', 'nao', 'pendente', 'reprovado', 'excluido')),
    status_evento VARCHAR2(20) NOT NULL CHECK (status_evento IN ('aprovado', 'reprovado', 'excluido', 'pendente', 'finalizado')),
    categoria VARCHAR2(20) NOT NULL CHECK (categoria IN ('esportes', 'catastrofes', 'eleicoes', 'bolsa de valores', 'e-sports'))
);

-- Criando sequência para EVENTOS
CREATE SEQUENCE SEQ_EVENTOS START WITH 1 INCREMENT BY 1;

-----

-- Tabela de APOSTAS
CREATE TABLE apostas (
    id_aposta INT PRIMARY KEY,
    id_evento INT NOT NULL, -- FK para a tabela EVENTOS
    id_usuario INT NOT NULL, -- FK para a tabela USUARIOS
    qtd_cotas INT NOT NULL,
    aposta VARCHAR2(3) NOT NULL CHECK (aposta IN ('sim', 'nao')),
    data_aposta TIMESTAMP(6) DEFAULT CURRENT_TIMESTAMP -- Hora atual no momento da inserção
);

-- Criando sequência para APOSTAS
CREATE SEQUENCE SEQ_APOSTAS START WITH 1 INCREMENT BY 1;

-----

```

```

-- Definindo as chaves estrangeiras (FK)

-- FK para vincular TRANSACOES_FINANCEIRAS ao USUARIOS
ALTER TABLE TRANSACOES_FINANCEIRAS ADD CONSTRAINT fk_transacao_usuario FOREIGN KEY (id_usuario) REFERENCES USUARIOS(id_usuario);

-- FK para vincular EVENTOS ao USUARIOS
ALTER TABLE EVENTOS ADD CONSTRAINT fk_evento_usuario FOREIGN KEY (id_usuario) REFERENCES USUARIOS(id_usuario);

-- FK para vincular APOSTAS ao EVENTOS e ao USUARIOS
ALTER TABLE APOSTAS ADD CONSTRAINT fk_aposta_evento FOREIGN KEY (id_evento) REFERENCES EVENTOS(id_evento);
ALTER TABLE APOSTAS ADD CONSTRAINT fk_aposta_usuario FOREIGN KEY (id_usuario) REFERENCES USUARIOS(id_usuario);
-----

-- Procedure para finalizar o evento
create PROCEDURE finalizar_evento(idEvento IN NUMBER, veredito IN VARCHAR2) AS
    veredito_perdedor VARCHAR2(3);
    soma_vencedora NUMBER := 0;
    soma_perdedora NUMBER := 0;
    total_premio NUMBER := 0;
BEGIN
    -- Define o veredito perdedor com base no veredito fornecido
    IF veredito = 'sim' THEN veredito_perdedor := 'nao';
    ELSE
        veredito_perdedor := 'sim';
    END IF;

    -- Atualiza o status e o resultado do evento
    UPDATE eventos
    SET resultado = veredito, status_evento = 'finalizado'
    WHERE id_evento = idEvento;

    -- Calcula a soma das apostas vencedoras
    SELECT COALESCE(SUM(qtd_cotas * valor_cota), 0) INTO soma_vencedora
    FROM apostas a
    JOIN eventos e ON a.id_evento = e.id_evento
    WHERE a.id_evento = idEvento AND a.aposta = veredito;

    -- Calcula a soma das apostas perdedoras
    SELECT COALESCE(SUM(qtd_cotas * valor_cota), 0) INTO soma_perdedora
    FROM apostas a
    JOIN eventos e ON a.id_evento = e.id_evento
    WHERE a.id_evento = idEvento AND a.aposta = veredito_perdedor;

    -- Reembolsa todos os apostadores se não houver vencedores ou perdedores
    IF soma_vencedora = 0 OR soma_perdedora = 0 THEN
        FOR aposta IN (
            SELECT a.id_usuario, a.qtd_cotas * e.valor_cota AS valor_reembolso
            FROM apostas a
            JOIN eventos e ON a.id_evento = e.id_evento
            WHERE a.id_evento = idEvento) LOOP

            -- Atualiza o saldo dos usuários
            UPDATE usuarios
            SET saldo = saldo + aposta.valor_reembolso
            WHERE id_usuario = aposta.id_usuario;

            -- Insere transação de reembolso
            INSERT INTO transacoes_financeiras (id_transacao, id_usuario, tipo_transacao, valor, data_transacao)

```

```

-- Insere transação de reembolso
INSERT INTO transacoes_financeiras (id_transacao, id_usuario, tipo_transacao, valor, data_transacao)
VALUES ( id_transacao SEQ_TRANSACOES_FINANCEIRAS.NEXTVAL, aposta.id_usuario, tipo_transacao 'reembolso',
        valor aposta.valor_reembolso, data_transacao SYSDATE);

END LOOP;

RETURN;

END IF;

-- Calcula o prêmio total
total_premio := soma_vencedora + soma_perdedora;

-- Distribui os ganhos proporcionalmente entre os vencedores
FOR aposta IN (
    SELECT a.id_usuario, a.qtd_cotas * e.valor_cota AS valor_aposta
    FROM apostas a
        JOIN eventos e ON a.id_evento = e.id_evento
    WHERE a.id_evento = idEvento AND a.aposta = veredito) LOOP

    -- Calcula o valor ganho pelo usuário
    DECLARE
        valor_ganho NUMBER;
    BEGIN
        valor_ganho := (aposta.valor_aposta / soma_vencedora) * total_premio;

        -- Atualiza o saldo do vencedor
        UPDATE usuarios
        SET saldo = saldo + valor_ganho
        WHERE id_usuario = aposta.id_usuario;

        -- Insere transação de ganho com a sequência SEQ_TRANSACOES_FINANCEIRAS
        INSERT INTO transacoes_financeiras (id_transacao, id_usuario, tipo_transacao, valor, data_transacao)
        VALUES ( id_transacao SEQ_TRANSACOES_FINANCEIRAS.NEXTVAL, aposta.id_usuario, tipo_transacao 'ganho_aposta',
                valor valor_ganho, data_transacao SYSDATE);

    END;
END LOOP;

END finalizar_evento;
/

```

```

-- Procedure para realizar uma aposta
create PROCEDURE realizar_aposta (
    idEvento IN NUMBER,
    idUsuario IN NUMBER,
    qtdCotas IN NUMBER,
    apostaSN IN VARCHAR2
) AS
BEGIN

    -- 1. Debita o saldo do usuário pelo valor da aposta
    UPDATE usuarios
    SET saldo = saldo - (qtdCotas * (SELECT valor_cota FROM eventos WHERE id_evento = idEvento))
    WHERE id_usuario = idUsuario;

    -- 2. Aumenta a quantidade de apostas no evento
    UPDATE eventos
    SET qtd_apostas = qtd_apostas + 1
    WHERE id_evento = idEvento;

    -- 3. Registra a aposta na tabela APOSTAS
    INSERT INTO apostas (id_aposta, id_evento, id_usuario, qtd_cotas, aposta)
    VALUES ( id_aposta SEQ_APOSTAS.NEXTVAL, id_evento idEvento, id_usuario idUsuario, qtd_cotas qtdCotas, aposta apostaSN);

    -- 4. Insere a transação financeira para registrar a aposta
    INSERT INTO transacoes_financeiras (id_transacao, id_usuario, tipo_transacao, valor)
    VALUES ( id_transacao SEQ_TRANSACOES_FINANCEIRAS.NEXTVAL, id_usuario idUsuario, tipo_transacao 'aposta',
            valor qtdCotas * (SELECT valor valor_cota FROM eventos WHERE id_evento = idEvento));

    COMMIT;
END realizar_aposta;
/

```

Considerações Finais

Síntese do Trabalho Realizado e Resultados Alcançados

O grupo trabalhou no desenvolvimento de uma plataforma de apostas online, seguindo os requisitos estabelecidos. Foram implementadas funcionalidades como cadastro e login de usuários, criação e moderação de eventos, apostas em eventos aprovados, e gerenciamento de carteiras virtuais. A integração entre o frontend e o backend foi realizada, resultando em um sistema funcional e acessível, com fluxo seguro para transações financeiras e apostas.

Os principais resultados alcançados incluem:

- Implementação Completa do Backend: Desenvolvimento de APIs para cadastro, login, criação de eventos, apostas, e gerenciamento de saldo na carteira.
- Sistema de Aprovação de Eventos: Moderadores podem aprovar ou rejeitar eventos, garantindo a qualidade e conformidade da plataforma.
- Gerenciamento de Carteira: Adição e saque de fundos, com cálculo automático de taxas, proporcionando uma simulação realista de operações financeiras.
- Interface Intuitiva: O frontend foi desenvolvido para garantir uma navegação amigável e uma boa experiência do usuário.

O sistema foi testado em diferentes cenários, e os principais fluxos foram validados com sucesso.

Principais Dificuldades Enfrentadas

O grupo enfrentou diversos desafios ao longo do projeto, que foram superados com trabalho em equipe e busca constante de soluções:

- Integração Backend e Frontend: Ajustar os serviços implementados no backend para atender às necessidades do frontend foi mais trabalhoso do que o esperado, devido a ajustes em parâmetros e respostas das APIs.
- Gestão de Sessões e Segurança: Garantir a segurança dos dados dos usuários e gerenciar tokens de sessão foi uma tarefa técnica que exigiu pesquisas e aprimoramento constante.

- Gerenciamento de Carteira e Transações: A criação de um sistema que simulasse transações financeiras, incluindo taxas e limites, apresentou desafios técnicos e matemáticos.
- Controle de Versionamento e Comunicação: A integração do trabalho de diferentes membros da equipe no mesmo repositório às vezes gerou conflitos de código, demandando coordenação e revisão frequentes.

Apesar das dificuldades, o grupo conseguiu desenvolver uma solução funcional, atendendo aos objetivos principais do projeto e entregando uma plataforma que pode ser aprimorada futuramente.