

## Especificação de Trabalho Acadêmico: Programação Orientada a Objetos em Java

### Tema: Simulação de uma Casa de Apostas

#### Objetivo

Desenvolver um sistema que simule uma casa de apostas utilizando a linguagem de programação Java, aplicando conceitos de Programação Orientada a Objetos (POO). O sistema deve incluir uma interface gráfica para interação com o usuário, classes de negócio para a lógica da aplicação e classes de acesso a dados utilizando JDBC para comunicação com um banco de dados PostgreSQL.

#### Requisitos Funcionais

##### 1. \*\*Cadastro de Usuários\*\*:

- O sistema deve permitir o cadastro de novos usuários com informações básicas (nome, e-mail, senha).
- Deve ser possível editar e excluir os dados de um usuário existente.

##### 2. \*\*Gerenciamento de Apostas\*\*:

- O usuário deve poder visualizar eventos disponíveis para apostas (ex: jogos de futebol, corridas, etc.).
- O sistema deve permitir que o usuário realize apostas em eventos disponíveis.
- As apostas devem incluir informações como valor apostado, odds, e possível retorno. (Odd é a expressão numérica da probabilidade de um resultado acontecer em um evento esportivo, indicando também o valor que se pode ganhar ao apostar nesse resultado).

##### 3. \*\*Histórico de Apostas\*\*:

- O usuário deve poder visualizar um histórico de suas apostas realizadas, incluindo o status (ganha, perdida, pendente, saldo).

##### 4. \*\*Administração de Eventos\*\*:

- Deve haver uma interface para administradores cadastrarem, editarem e excluírem eventos.
- Os eventos devem ter informações como nome, data, descrição, e odds associadas.

##### 5. \*\*Autenticação e Autorização\*\*:

- Implementação de login e logout para usuários e administradores.
- Diferenciação de permissões entre usuários comuns e administradores.

#### #### Requisitos Não Funcionais

##### 1. \*\*Usabilidade\*\*:

- A interface gráfica deve ser intuitiva e fácil de usar.
- A navegação entre as telas deve ser clara e objetiva.

##### 2. \*\*Segurança\*\*:

- Senhas devem ser armazenadas de forma segura (ex: hash com salt).

##### 3. \*\*Desempenho\*\*:

- O sistema deve ser eficiente, com tempo de resposta aceitável para as operações.

#### #### Estrutura do Projeto

1. **Interface Gráfica**:
  - Utilizar Swing para a criação da interface gráfica.
  - Telas obrigatórias: Login, Cadastro de Usuário, Tela Principal, Tela de Apostas, Tela de Histórico, Tela de Administração de Eventos.
2. **Classes de Negócio**: Ex.: Usuario, Apostas, Evento, etc.
3. **Classes de Acesso a Dados**:
  - Utilizar JDBC para a comunicação com o banco de dados PostgreSQL.
  - Classes como por exemplo **UsuarioDAO**, **ApostasDAO**, **EventoDAO** para as operações CRUD (Create, Read, Update, Delete).
4. **Banco de Dados**:
  - Criar um banco de dados PostgreSQL com as tabelas necessárias para armazenar usuários, apostas, eventos, etc.
  - Scripts SQL para criação e povoamento inicial das tabelas.

#### #### Entregáveis

1. **Código Fonte**:
  - Arquivos Java, scripts SQL, e outros recursos necessários para a execução do projeto.
2. **Documentação**:
  - Documento descrevendo a arquitetura do sistema, diagramas de classes, e instruções para configuração e execução do projeto.
3. **Apresentação**:
  - Slides de apresentação do projeto destacando os principais pontos e funcionalidades implementadas.

#### #### Avaliação

1. **Funcionalidades Implementadas**: 40%
2. **Qualidade do Código**: 20%
3. **Interface Gráfica e Usabilidade**: 20%
4. **Documentação e Apresentação**: 20%

#### ### Cronograma

Apresentação em Sala (lab 3) com data definida em durante a apresentação desta especificação.