

Projeto Final

Relatório

SCC-541 Laboratório de Bases de Dados

Leonardo Gonçalves Chahud - 5266649
Murilo Franchi - 9790760
Rafael Dantas - 12563686

Prof. Dr Caetano Traina Jr.
PAE: Igor Alberte R. Eleutério

Sumário

1. Descrição do projeto	2
1. Administrar usuários:	2
2. Telas de ferramentas:	2
3. Ações possíveis do usuário:	3
4. Definição da tela de overview:	3
5. Relatórios:	4
2. Tabelas criadas	5
3. Funções criadas	6
4. Ferramentas utilizadas	7
5. Segurança	8

1. Descrição do projeto

O projeto tem como objetivo integrar o conhecimento adquirido na disciplina SCC-541 Laboratório de Bases de Dados, com a criação de um protótipo de uma ferramenta que seja capaz de manipular os dados e gerar relatórios de uma forma amigável.

O projeto é dividido em 5 partes:

1. Administrar usuários:

O acesso a ferramenta somente pode ser feito a partir do login que cada usuário deve efetuar para ter acesso à base de dados. Cada usuário com acesso deve ser de um dos seguintes tipos:

- Admin: Pode acessar quaisquer informações da base;
 - Login: admin;
 - Senha: admin;
- Escuderia: Pode acessar apenas informações relativas à sua Escuderia e aos pilotos que correm por ela;
 - Login: (constructorref)_c
 - Senha: constructorref
 - Exemplo: constructorref = mclaren,
 - Login: mclaren_c
 - Senha: mclaren
- Piloto: Pode acessar apenas informações relativas ao seu desempenho.
 - Login: (driverref)_p
 - Senha: driverref
 - Exemplo: driverref = hamilton,
 - Login: hamilton_p
 - Senha: hamilton

2. Telas de ferramentas:

A ferramenta deve estar centrada em três telas principais, cada uma correspondente a um modo de operação da ferramenta:

- Tela 1: tela de login;
- Tela 2: tela de overview;
- Tela 3: tela de relatórios.

3. Ações possíveis do usuário:

- Admin:
 - Cadastrar escuderias;
 - Cadastrar pilotos;
 - Visualizar overview;
 - Visualizar relatório.
- Escuderia:
 - Consultar piloto;
 - Visualizar overview;
 - Visualizar relatório.
- Piloto:
 - Visualizar overview;
 - Visualizar relatório.

4. Definição da tela de overview:

Para cada tipo de usuário, a tela de overview deve apresentar informações diferentes, da seguinte maneira:

- Admin:
 - Quantidade de pilotos cadastrados;
 - Quantidade de escuderias cadastradas;
 - Quantidade de corridas cadastradas;
 - Quantidade de temporadas (seasons) cadastradas.
- Escuderia:
 - Quantidade de vitórias da escuderia;
 - Quantidade de pilotos diferentes que já correram pela escuderia;
 - Primeiro e último ano em que há dados da escuderia na base.
- Piloto:
 - Quantidade de vitórias do piloto;
 - Primeiro e último ano em que há dados do piloto na base.

5. Relatórios:

Apresente os relatórios considerando que eles devem ser compreendidos por um usuário do respectivo tipo. É interessante, por exemplo, alguma ordenação que faça sentido em cada relatório.

- Admin:
 - Relatório 1: indica a quantidade de resultados por cada status, apresentando o nome do status e sua contagem;
 - Relatório 2: receber o nome de uma cidade e, para cada cidade que tenha esse nome, apresenta todos os aeroportos brasileiros que estejam a, no máximo, 100km das respectivas cidades e que sejam dos tipos 'medium airport' ou 'large airport'.
- Escuderia:
 - Relatório 3: lista os pilotos da escuderia, bem como a quantidade de vezes em que cada um deles alcançou a primeira posição em uma corrida ;
 - Relatório 4: liste a quantidade de resultados por cada status, apresentando o status e sua contagem, limitados ao escopo de sua escuderia.
- Piloto:
 - Relatório 5: consultar a quantidade de vitórias obtidas, apresentando o ano e a corrida onde cada vitória foi alcançada;
 - Relatório 6: liste a quantidade de resultados por cada status, apresentando o status e sua contagem, limitado ao piloto logado.

2. Tabelas criadas

Para o projeto final, foram criadas duas novas tabelas:

- users: armazena os usuários do sistema (admin, escuderia ou piloto),

```
create table if not exists users(  
    userid smallint not null,  
    login text not null,  
    password text not null,  
    tipo text not null,  
    idoriginal smallint not null,  
    constraint pk_users primary key(userid),  
    constraint uk_users unique(login),  
    constraint ck_users_tipo check(tipo = 'admin' or tipo = 'escuderia' or  
tipo = 'piloto')  
);
```

- log_table: armazena a movimentação da base de dados (conectar, desconectar e commit).

```
create table if not exists log_table(  
    logid serial not null,  
    userid int not null,  
    data date default current_date,  
    hora time default localtime(5),  
    tipo text not null,  
    constraint pk_log primary key(logid),  
    constraint fk_log_table foreign key(userid) references users(userid),  
    constraint ck_log_table_tipo check(tipo = 'connect' or tipo =  
'disconnect' or tipo = 'commit')  
);
```

3. Funções criadas

Para a criação de novos pilotos e escuderias, foram criadas as seguintes funções:

- `new_driver(driverref_ text, number_ integer, code_ text, forename_ text, surname_ text, dob_ date, nationality_ text);`
- `new_constructor(constructorref_ text, name_ text, nationality_ text);`

Para inserção e atualização de pilotos e escuderias na tabela de users, foram criadas as seguintes funções e triggers:

- `insert_driver();`
- `update_driver();`
- `tr_insert_driver;`
- `tr_update_driver;`
- `insert_constructors();`
- `update_constructors();`
- `tr_insert_constructors;`
- `tr_update_constructors.`

Para inserção dos dados já existentes de pilotos e escuderias na tabela de users, foram criadas as seguintes funções:

- `add_driver_to_users();`
- `add_constructor_to_users().`

Para inserção de logs na tabela `log_table`, foi criado a seguinte função:

- `new_log(userid_ int, tipo_ text).`

Em relação ao overview do admin, foram criadas as seguintes funções:

- `total_drivers();`
- `total_constructors();`
- `total_races();`
- `total_seasons();`

Em relação ao overview da escuderia, foram criadas as seguintes funções:

- `consultar_piloto(escuderia text, piloto text);`
- `total_wins_constructors(nameref text);`
- `total_drivers_constructors(nameref text);`
- `first_last_year_constructors(nameref text).`

Em relação ao overview do piloto, foram criadas as seguintes funções:

- `total_wins_drivers(nameref text);`
- `first_last_year_drivers(nameref text).`

Em relação ao relatório do admin, foram criadas as seguintes funções:

- `report1();`
- `report2().`

Em relação ao relatório da escuderia, foram criadas as seguintes funções:

- `report3();`
- `report4();`

Em relação ao relatório do piloto, foram criadas as seguintes funções:

- `report5();`
- `report6().`

4. Ferramentas utilizadas

A interface do projeto foi desenvolvida usando a linguagem de programação [python](#), utilizando as bibliotecas [psycopg2](#) (para conectar com a base de dados) e a biblioteca [prettytable](#) (para melhor visualização dos dados).

A base de dados utilizada foi o [PostgreSQL](#). Os SGBDs utilizados durante o desenvolvimento foram o [pgAdmin](#) e o [DBeaver](#).

5. Segurança

Esta seção apresenta alguns comentários relativos à segurança do projeto desenvolvido. Aponta, não necessariamente o que foi aplicado, mas sim o que talvez seja considerado o ideal. Dessa forma, é demonstrado o entendimento e a conscientização sobre o tema.

- 1) Credenciais: As credenciais utilizadas na(s) conta(s) administrativa(s) são ilustrativas visto que este projeto não será implementado em um ambiente real.
- 2) Hash das senhas dos usuários: Foi utilizado o MD5 como foi pedido no enunciado do projeto. Contudo, o grupo entende que o MD5 já está desatualizado e que um algoritmo de hash ideal seria, por exemplo, SHA512 e outros tipos de hash cujo tamanho seja maior (mais bits utilizados).
- 3) SQL Injections: As consultas que apresentaram como parâmetro algum input do usuário foram filtradas através do uso adequado da biblioteca `psycpg2`.
- 4) Papéis adequados para os usuários: Foram atribuídas permissões condizentes com o perfil de cada usuário no banco.