

# **Projeto de Desenvolvimento de uma Aplicação Simulando um Sistema Real de Interface entre Uma Linguagem de Programação e o Banco de Dados Oracle**

Disciplina: **Banco de Dados**

Professor: **Howard Roatti**

Última Atualização: **09/09/2024**

- **Incluído(s)**: Tabela com Sugestão de Temas.

- **Itens Atualizados**: Nenhum.

## Sumário

Edital: .....	1
ESQUEMA DO PROJETO NO GITHUB .....	7
FAQ .....	8
ROTEIRO PARA AUXÍLIO NA EXECUÇÃO DO TRABALHO.....	9
Orientações para Desenvolvimento do Projeto: .....	10

## Edital:

**0** - Formem um grupo de **5 pessoas**, envie os nomes dos componentes para o professor **até o dia 16/09/2024** no endereço: **howard.cruz@faesa.br**

**1** - Escolha um tema para a aplicação, exceto a aplicação do exemplo

Sugestão de Temas (Não precisam ficar limitados a eles)

<b>Sistema</b>	<b>Descrição</b>
<i>Sistema de Gerenciamento de Livraria</i>	Controle de livros, autores e pedidos, com relatórios de vendas por gênero e autor.
<i>Sistema de Gestão de Estudantes</i>	Controle de estudantes, cursos e notas, com relatórios de média por turma e desempenho por curso.
<i>Sistema de Controle de Estoque</i>	Controle de produtos e fornecedores, com relatórios de estoque baixo e quantidade comprada.
<i>Sistema de Reservas de Hotel</i>	Controle de reservas e hóspedes, com relatórios de ocupação e número de reservas por mês.
<i>Sistema de Gerenciamento de Tarefas</i>	Controle de tarefas e usuários, com relatórios de tarefas concluídas e tempo gasto.
<i>Sistema de Controle de Consultas Médicas</i>	Controle de médicos, pacientes e consultas, com relatórios de consultas por médico e especialidade.
<i>Sistema de Vendas Online</i>	Controle de produtos, clientes e pedidos, com relatórios de vendas por categoria e cliente.
<i>Sistema de Locação de Veículos</i>	Controle de veículos e locações, com relatórios de locações por modelo e duração da locação.
<i>Sistema de Biblioteca Escolar</i>	Controle de livros, empréstimos e alunos, com relatórios de livros emprestados e atrasos.
<i>Sistema de Gerenciamento de Projetos</i>	Controle de projetos e tarefas, com relatórios de progresso por projeto e desempenho por equipe.
<i>Sistema de Controle de Ponto</i>	Controle de funcionários e horários, com relatórios de horas trabalhadas e atrasos.
<i>Sistema de Academia de Ginástica</i>	Controle de alunos, treinos e instrutores, com relatórios de frequência por aluno e por horário.
<i>Sistema de Doações para ONG</i>	Controle de doadores e doações, com relatórios de doações por campanha e valor total arrecadado.
<i>Sistema de Registro de Eventos</i>	Controle de eventos e participantes, com relatórios de número de participantes por evento.
<i>Sistema de Agendamento de Salas</i>	Controle de salas, reservas e usuários, com relatórios de uso por sala e horário.
<i>Sistema de Atendimento ao Cliente</i>	Controle de atendimentos, clientes e funcionários, com relatórios de tempo médio de atendimento.
<i>Sistema de Manutenção de Equipamentos</i>	Controle de equipamentos, manutenções e fornecedores, com relatórios de frequência de manutenção e custos.
<i>Sistema de Controle de Transporte Escolar</i>	Controle de alunos, rotas e veículos, com relatórios de rotas mais utilizadas e número de alunos transportados.
<i>Sistema de Gerenciamento de Cursos Online</i>	Controle de cursos, alunos e inscrições, com relatórios de inscrições por curso e progresso dos alunos.

2 - A aplicação deve possuir no mínimo 2 (duas) tabelas com relacionamentos entre elas

3 - Desenvolvam o diagrama relacional, especificando os atributos, chaves, obrigаторiedades e relacionamentos

4 - Criem um script para gerar as tabelas e os relacionamentos no banco de dados relacional de sua escolha: Oracle, MySQL ou PostgreSQL

5 - Criem um programa na linguagem de sua preferência que:

a) Possua um menu com as seguintes opções:

i) relatórios

ii) inserir registros

iii) remover registros

iv) atualizar registros

v) sair

Obs.: O menu pode ser personalizado, desde que atenda aos requisitos listados acima;

b) Ao iniciar, seu programa deverá exibir o nome da aplicação e os nomes dos componentes do grupo (*Splash Screen*)

c) Ao iniciar, seu programa deverá informar se as tabelas possuem registros, para isso execute uma consulta para tabela que seja capaz de contar o número de registros (**select** count(1) total\_{nome\_da\_entidade} **from** {nome\_da\_entidade})

d) Exemplo de tela de inicialização:

```

#####
#                               #
#           SISTEMA DE VENDAS           #
#                               #
#                               #
#   TOTAL DE REGISTROS EXISTENTES   #
#   1 - FORNECEDORES: 50           #
#   2 - CLIENTES: 10               #
#   3 - PRODUTOS: 180              #
#   4 - PEDIDOS: 1349              #
#   5 - ITENS: 5819                #
#                               #
#                               #
#   CRIADO POR: FULANO             #
#                   BELTRANO        #
#                   CICLANO         #
#                               #
#                               #
#   DISCIPLINA: BANCO DE DADOS     #
#                   2022/2          #
#   PROFESSOR: HOWARD ROATTI       #
#####

```

e) O menu deverá ser exibido constantemente, de acordo com o fluxo de execução projeto

f) Ao criar submenus, durante a utilização deles, deverão ser exibidos constantemente de acordo com o fluxo de execução

6 - Descrição das **funcionalidades** dos itens de **menu**:

a) **relatórios**:

(1) deve haver ao menos duas opções de relatórios:

(i) consulta realizando a sumarização (agrupamento) de alguma tabela, por exemplo: o valor total dos pedidos agrupados por fornecedores.

(ii) uma consulta com junção de tabelas, por exemplo: relatório de itens de pedido.

**b) inserir registros:**

- (1) deverá exibir um menu com as opções das entidades existentes
- (2) o usuário deverá selecionar uma das entidades
- (3) exiba cada atributo da entidade selecionada e permita que o usuário informe os dados delas
- (4) realize a inserção do registro
- (5) pergunte ao usuário se deseja inserir mais algum registro, caso a resposta seja "Não", volte ao menu principal. Caso a resposta seja "Sim", volte ao menu de opções de entidades existentes ou permita que continue realizando a ação na entidade previamente selecionada.

**c) remover registros:**

- (1) deverá exibir um menu com as opções das entidades existentes
- (2) o usuário deverá selecionar uma das entidades
- (3) deverá listar as tuplas (registros) da entidade (tabela) selecionada, no mínimo a chave primária e um campo descritivo
- (4) o usuário irá selecionar a tupla que deseja remover através de uma opção (chave primária, listagem numérica fixa, por exemplo)
- (5) confirme com o usuário se deseja realmente remover o registro, caso a resposta seja "Não", volte ao menu principal. Caso a resposta seja "Sim":
  - (i) verifique se o registro que o usuário deseja alterar é uma FK (Foreign Key) em outra tabela, caso seja: informe que aquele registro não poderá ser excluído ou pergunte se é desejo dele remover o registro da tabela filha, excluindo caso a resposta seja sim.
- (6) pergunte ao usuário se deseja excluir mais algum registro, caso a resposta seja "Não", volte ao menu principal. Caso a resposta seja "Sim", volte ao menu de

opções de entidades existentes ou permita que continue realizando a ação na entidade previamente selecionada.

**d) atualizar registros:**

- (1) deverá exibir um menu com as opções das entidades existentes
- (2) o usuário deverá selecionar uma das entidades
- (3) deverá listar as tuplas (registros) da entidade (tabela) selecionada, no mínimo a chave primária e um campo descritivo
- (4) o usuário irá selecionar a tupla que deseja atualizar através de uma opção (chave primária, listagem numérica fixa, por exemplo)
- (5) solicite que ele informe todos os atributos atômicos, exceto a chave primária ou permita que ele escolha qual atributo deseja alterar
- (6) realize a atualização da tupla com os dados informados pelo usuário
- (7) exiba o registro atualizado
- (8) pergunte ao usuário se deseja atualizar mais algum registro, caso a resposta seja "Não", volte ao menu principal. Caso a resposta seja "Sim", volte ao menu de opções de entidades existentes ou permita que continue realizando a ação na entidade previamente selecionada.

**7 - Entregáveis:**

- (a) Todos os scripts deverão estar disponíveis no **Github do grupo**
  - (i) script de criação das tabelas e relacionamentos
  - (ii) código fonte do programa desenvolvido, devidamente organizado em diretórios (não compactado)
  - (iii) diagrama relacional (pdf, png ou jpeg)
  - (iv) video demonstrativo das funcionalidades implementadas, explicando o passo a passo da aplicação

(v) README.MD explicando como executar o projeto no ambiente proposto (Linux)

(b) Deverá ser postado no AVA apenas o link para o Github (caso o AVA solicite o envio de um arquivo, envie um documento de texto (txt) com o link do Github no corpo do documento)

(c) Somente será corrigido se seguir as orientações do edital, caso contrário a avaliação será desconsiderada e nota automaticamente zerada!

## 8 - Distribuição da Pontuação:

Nota	Item	Item Específico
1,5	Relatórios	1.0 <6.a.1.i> e 0.5 <6.a.1.ii>
1,5	Inserir Registros	1.0 <6.b.[1-4]> e 0.5 <6.b.5>
1,5	Remover Registros	1.0 <6.c.[1-4 e 6]> e 0.5 <6.c.5.i>
1,5	Atualizar Registros	1.0 <6.d.[1-6]> e 0.5 <6.d.[7-8]>
0,5	Diagrama Relacional, corretamente desenvolvido.	
0,5	Interface Amigável.	
	Obs.: não é obrigatória a criação de telas ou páginas, mesmo no console é possível ter uma interface amigável.	
0,5	Vídeo Demonstrativo no YouTube explicando o objetivo do trabalho e a utilização do sistema passo a passo em todos os menus. Não há necessidade alguma de explicar o código, apenas o funcionamento do sistema. Todos os menus devem ser exibidos no vídeo em funcionamento.	
0,5	Documentação explicativa de utilização do sistema no Linux no README.MD do repositório no GitHub.	
8,0	Total	

### Observações Importantes:

(1) todos os componentes do grupo deverão participar de sua criação, caso seja informado pelos membros ativos que houve algum componente que não participou de sua criação, esse terá sua avaliação zerada automaticamente.

(2) lembre-se que trabalho não possui avaliação substitutiva.

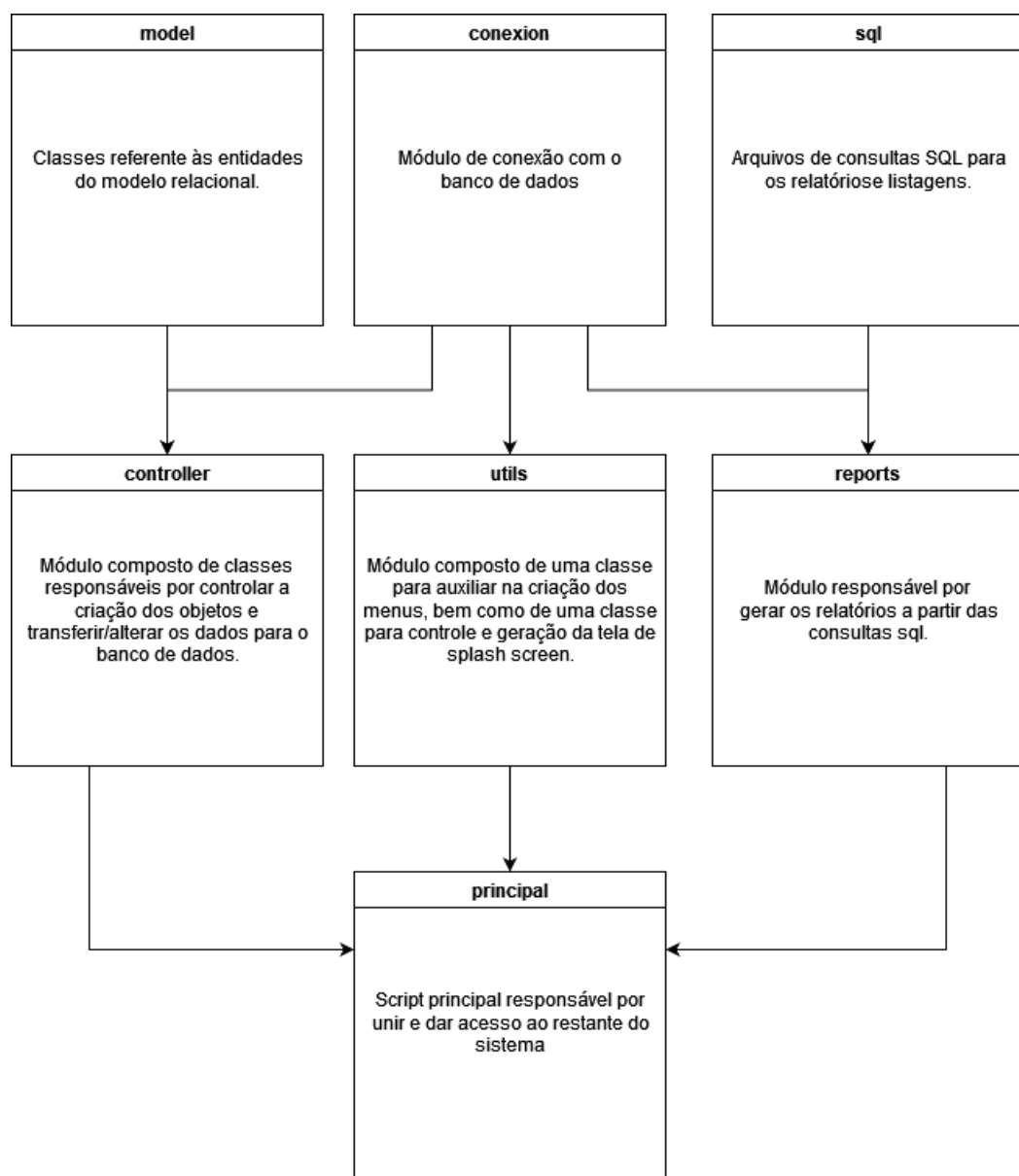
(3) atrasos serão tolerados, porém, perderão 0,5 pontos a cada dia de atraso.

(4) o banco de dados deverá ser: Oracle, MySQL ou PostgreSQL.

## 9 - Projeto de Exemplo: [https://github.com/howardroatti/example\\_crud\\_oracle](https://github.com/howardroatti/example_crud_oracle)

- (a) façam um fork (cópia) do projeto e analise o projeto antes de iniciar o de vocês
- (b) não deixem para fazer em cima da hora, o projeto é extenso e vocês podem não ter tempo hábil de realizá-lo
- (c) não serão aceitos projetos idênticos ou cópias parciais

### ESQUEMA DO PROJETO NO GITHUB





## **FAQ**

### **Podemos criar interface gráfica?**

Sim

### **Podemos manter o programa apenas no console sem interface gráfica?**

Sim, desde que tenha uma interface amigável e seja de fácil utilização.

### **Podemos utilizar uma linguagem diferente?**

Sim, a linguagem que vocês se sentirem mais confortáveis, porém, seu projeto deve ser capaz de ser executado em um ambiente Linux

### **Podemos usar algum componente, biblioteca ou funcionalidade que facilite a criação de consultas e alterações das tabelas do banco de dados?**

Não, você deverá criar as consultas e atualizações utilizando uma variável texto e concatenar com os atributos das entidades. Não será possível a utilização de um ORM.

### **Podemos realizar o trabalho sem grupos?**

Prioritariamente, o trabalho deverá ser desenvolvido em grupo. É importante que aprendam a trabalhar em equipe e designar tarefas. Entretanto, é possível a realização do trabalho individualmente, desde que o aluno se responsabilize por desenvolver o projeto e entregar em tempo hábil para sua correção, não deixando para informar na semana de entrega a necessidade de realizar a atividade solo.

### **É possível trocar de grupo?**

A troca de grupo será permitida até duas semanas antes da data de entrega, para que o aluno seja capaz de contribuir com o desenvolvimento da atividade proposta. Caso seja interesse do aluno, esse deverá procurar um grupo disponível que aceite sua entrada.

## **ROTEIRO PARA AUXÍLIO NA EXECUÇÃO DO TRABALHO**

- 1 - Crie o modelo pelo Sql Power Architect
- 2 - Gere um PDF com o modelo proposto
- 3 - Gere o script de criação das tabelas e relacionamentos
- 4 - Criem os objetos relacionados às entidades do modelo proposto
- 5 - Criem as classes controladoras para cada objeto do modelo proposto
- 6 - Criem os relatórios solicitados no edital
- 7 - Altere o splash\_screen
- 8 - Altere a classe principal

## Orientações para Desenvolvimento do Projeto:

### **Etapa 1:** Modelagem e Criação de Tabelas

- Defina quais serão as entidades e seus atributos;
- Lembre-se de definir corretamente os nomes dos campos (nome único por tabela);
- Lembre-se de definir corretamente o tipo de cada campo (inteiro, real, texto, data, etc);
- Lembre-se de definir os campos determinantes (PK e FK);
- Crie o diagrama relacional com as tabelas, campos, tipos, chave primária, chave estrangeira e relacionamentos com as devidas cardinalidades;
- Gere o script para criação das tabelas e relacionamentos;

### **Etapa 2:** Criação das Classes Principais

- Desenvolva as classes que representam as tabelas do tema que estão trabalhando, utilize os exemplos em src/model;
- Lembre-se que sempre que um objeto possui uma associação com outro, a classe deverá ter como atributo uma instância do outro objeto, não apenas seu atributo. Olhe os exemplos: src/model/itens\_pedido.py e src/model/pedidos.py;
- Lembre-se de criar os getters e setters para cada atributo;
- Lembre-se de criar um método to\_string para cada classe;

### **Etapa 3:** Criação das Classes Controladoras

- Desenvolva as classes que irão realizar a interface entre os objetos da **Etapa 2** e as tabelas da **Etapa 1**;
- Lembre-se de criar um método de verificação de existência ou não do registro que está trabalhando. Sendo um novo (Inserção), ele não deve existir, para que possa

ser criado e então mantemos a integridade de entidade. Sendo um registro existente (Atualização ou Exclusão), ele deve existir, para que possa realizar o procedimento;

- Lembre-se de, em cada interface, realizar a conexão antes de efetuar a execução dos scripts SQL;

- Lembre-se de realizar a concatenação dos atributos com a query;

- Veja os exemplos existentes, para criar o seu. Eles se encontram em:  
src/controller;

- Caso utilize o objeto SEQUENCE do Oracle para controlar as chaves primárias (PK), veja o exemplo de utilização que há em:  
src/controller/controller\_item\_pedido.py e src/controller/controller\_pedido.py;

- Realize os testes de cada query SQL que irá criar, para saber se está criando corretamente. É importante respeitar os tipos de dados e o teste através do SQL Developer ou outra IDE que queira utilizar é muito importante;

#### **Etapa 4:** Criação dos Relatórios

- Junto do seu grupo, definam os relatórios que irão criar para atender ao edital. Lembre-se que, obrigatoriamente, deverão ter dois relatórios: um com funções agregadas e cláusula group by (sumarização) e outro com joins (junção de tabelas);

- Caso seja necessário exibir informações das tabelas no formato de listagem, como no exemplo da controladora de Itens de Pedidos e da controladora de Pedidos, crie as consultas no formato de relatório no repositório src/sql. Lá você pode verificar alguns exemplos existentes;

- Altere a classe src/reports/relatorios.py para que ela seja capaz de ler os relatórios que criou e executar as consultas de cada relatório;

#### **Etapa 5:** Splash Screen e Menus

- Altere e/ou inclua no arquivo src/utils/config.py os menus de relatórios, entidades e algum outro que queira inserir;

- Altere os atributos da classe `src/utils/splash_screen.py` para que apareça o nome do sistema proposto, a quantidade de registros em cada tabela, os nomes dos componentes do grupo e preserve o nome do professor, disciplina e semestre;

#### **Etapa 6: Principal**

- Altere as interfaces existentes (métodos inserir, atualizar, excluir e reports) para atender ao sistema proposto;
- Lembre-se de que no sistema proposto não há a necessidade de ter mais do que duas entidades/tabelas;
- Lembre-se de verificar se todos os imports estão adequados ao que criou;
- Lembre-se de alterar os fluxos de menus para atender ao edital, no exemplo existem partes que não foram implementadas que podem afetar o resultado do trabalho;

#### **Etapa Final:**

- Estude o projeto de exemplo, é importante que tirem todas as dúvidas durante o desenvolvimento do trabalho;
- Respeite a Orientação a Objetos e a utilize corretamente;
- Respeite a criação e utilização dos comandos SQL. Use adequadamente;
- Realize diversos testes: peça cada componente do grupo que teste, peça pessoas de fora do grupo para testar;
- Estabeleçam metas semanais para execução do trabalho. Lembre-se que o tempo é curto;
- Lembre-se de gravar o vídeo demonstrando a utilização do sistema proposto;
- Lembre-se de manter o GitHub atualizado até o dia da entrega;