|  |
| --- |
|  |
|  |
| **Projeto Interdisciplinar**  **4° Período** |
|  |
|  |
|  |
|  |
| **<Nome do Sistema>** |
|  |
|  |
|  |
|  |
|  |
| **Aluno 01**  **Aluno 02** |
| **Aluno 03** |
| **Aluno 04** |
|  |

|  |
| --- |
|  |

Março de 2024

Sumário

[1 Introdução 1](#__RefHeading___Toc65847972)

[1.1 Objetivo 1](#__RefHeading___Toc65847973)

[1.2 Justificativa 1](#__RefHeading___Toc65847974)

[2 Análise do Sistema Proposto 2](#__RefHeading___Toc65847975)

[2.1 Escopo do Sistema 2](#__RefHeading___Toc65847976)

[2.2 Não Escopo do Sistema 2](#__RefHeading___Toc65847977)

[2.3 Requisitos do Sistema 2](#__RefHeading___Toc65847978)

[2.4 Glossário 3](#__RefHeading___Toc65847979)

[2.5 Mapeamento de Processos 3](#__RefHeading___Toc65847980)

[2.6 Diagrama de Caso de Uso 4](#__RefHeading___Toc65847981)

[2.7 Caso de Uso Expandido 5](#__RefHeading___Toc65847982)

[2.8 Diagrama de Classe 6](#__RefHeading___Toc65847983)

[2.9 Testes do Software/Aplicativo/Solução 7](#__RefHeading___Toc65847984)

[3 Projeto de Interface Gráfica (Modelo não Funcional) 8](#__RefHeading___Toc65847985)

[3.1 Modelo Não Funcional 8](#__RefHeading___Toc65847986)

[3.2 Internacionalização do software 9](#__RefHeading___Toc65847987)

[4 Projeto de Banco de Dados 10](#__RefHeading___Toc65847988)

[4.1 Diagrama de Entidade e Relacionamento (DER) 10](#__RefHeading___Toc65847989)

[4.2 Diagrama de Banco de Dados 11](#__RefHeading___Toc65847990)

[4.3 Quadro Resumo dos Objetos de Banco de Dados 11](#__RefHeading___Toc65847991)

[4.4 Mapeamento para o Modelo Lógico (Modelo Relacional) 12](#__RefHeading___Toc65847992)

[4.5 Dicionário de Dados 12](#__RefHeading___Toc65847993)

[4.6 Script de Banco de Dados 12](#__RefHeading___Toc65847994)

[4.7 Inserção no banco de dados 13](#__RefHeading___Toc65847995)

[4.8 Views 14](#__RefHeading___Toc65847996)

[4.9 Procedures 14](#__RefHeading___Toc65847997)

[5 Considerações Finais 15](#__RefHeading___Toc65847998)

[5.1 Dificuldades Encontradas 15](#__RefHeading___Toc65847999)

[5.2 Sugestões de Melhoria 15](#__RefHeading___Toc65848000)

[6 Referências 16](#__RefHeading___Toc65848001)

# Introdução

|  |
| --- |
| **Título do Projeto:** |

|  |
| --- |
| **Nome do Grupo:** <um nome pelo qual o grupo será conhecido> |

|  |  |  |
| --- | --- | --- |
| **Nr.** | **Aluno(s) envolvido(s)** | **E-mail(s)** |
| 01 | <coloque aqui o nome dos alunos> | <email> |
| 02 |  |  |
| 03 |  |  |
| 04 |  |  |

|  |
| --- |
| **Objetivo** |
| No objetivo deve aparecer **para** que o sistema será desenvolvido. Por exemplo:  Construir uma aplicação web de controle de notas e faltas para ?????? |

|  |
| --- |
| **Justificativa** |
| Na justificativa deve aparecer o porquê da elaboração do projeto. Por exemplo:  O presente projeto justifica-se pela ausência ou necessidade, etc...... |

# Análise do Sistema Proposto

|  |
| --- |
| **Escopo do Sistema** |
| (detalhamento textual sobre o sistema a ser desenvolvido).  O Sistema de Emissão de Certificados para Treinamento em Segurança do Trabalho é uma plataforma desenvolvida para facilitar o processo de registro, gerenciamento e emissão de certificados para funcionários que participam de treinamentos em segurança do trabalho.  Com este sistema, as empresas serão cadastradas facilmente, fornecendo algumas informações relevantes como nome, CNPJ, endereço e contato. Da mesma forma, os funcionários serão cadastrados, em ambos os casos caso já não estejam registrados.  Uma vez cadastrados, os funcionários serão registrados para participar de treinamentos específicos, com a definição da data de início do curso. O sistema conta com os cadastrados dos instrutores permitindo o CRUD deles, aqui é coletada a assinatura deles uma informação importante que irá ser utilizada na geração do certificado. Além disso, o sistema pode coletar a assinatura do funcionário para pré-preencher o certificado, agilizando o processo de emissão.  Para garantir a qualidade do treinamento, o sistema conta com um banco de questões, categorizadas por tipo de curso, que são utilizadas para gerar provas customizadas. Após a realização do treinamento e da prova, o sistema automaticamente gera o certificado para os funcionários aprovados. Este certificado pode ser impresso fisicamente ou enviado por e-mail para a empresa do funcionário.  O sistema também oferece recursos de controle e monitoramento, permitindo o acompanhamento do histórico de treinamentos e certificações de cada funcionário. Além disso, relatórios e estatísticas podem ser gerados para análise do desempenho dos treinamentos, taxas de aprovação e outras métricas relevantes. |

## **Não Escopo do Sistema**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Detalhamento textual sobre o que o sistema não vai fazer, o que não será desenvolvido).

- O sistema não fornece o treinamento, é apenas um gerenciamento do processo de emissão de certificado

- O sistema não realiza provas remotamente, a princípio, somente serão geradas para impressão

- O sistema não valida legalmente os certificados. Fica a cargo dos envolvidos.

|  |
| --- |
| **Requisitos do Sistema** |
| |  |  | | --- | --- | | **Nr.** | **Requisito** | | 01 | O sistema deve permitir o cadastro (inclusão, alteração e exclusão) das empresas clientes, . | | 02 | O sistema deve permitir o registro de pedidos de produtos solicitados pelos clientes. | | 03 | O sistema deve emitir relatório de vendas por período. | | ... | ... | |

|  |
| --- |
| **Glossário** |
| |  |  | | --- | --- | | **Termo** | **Descrição** | | Diária | Valor a ser pago pelo cliente referente a um dia completo de aluguel do carro. | | Taxa de serviço | Percentual fixo cobrado sobre os serviços adicionais, que deve ser pago pelo cliente que faz uma retirada de carro. | | ... | ... | |

|  |
| --- |
| **Mapeamento de Processos** |
| Será desenvolvido na presente documentação, o mapeamento do processo do projeto Interdisciplinar, no qual o mesmo fará uso da notação BPMN para elaboração dos fluxos.    \*imagem meramente ilustrativa |

|  |
| --- |
| **Diagrama de Caso de Uso** |
|  |

|  |
| --- |
| **Caso de Uso Expandido** |
|  |

|  |
| --- |
| **Diagrama de Classe** |
|  |

|  |
| --- |
| **Testes do Software/Aplicativo/Solução** |
| 1. **Plano de Teste**    1. **Caso de Teste** |

# Projeto de Interface Gráfica (Modelo não Funcional)

|  |
| --- |
| **Modelo Não Funcional** |
| **Interface 01:**  **Descrição:** responsável pela interação para autenticação de usuários no sistema.  **Nome:** FrmAutenticação  **Case de Uso:** <colocar aqui o nome do caso de uso, se houver. Por exemplo: Cadastrar Cliente>    **Interface 02**  **Descrição:** responsável pela interação do cadastro de clientes.  **Nome:** FrmCadastrarCliente  **Case de Uso:** <colocar aqui o nome do caso de uso, se houver. Por exemplo: Cadastrar Cliente>    <fazer o mesmo para todas as interfaces, inclusive relatórios> |

|  |
| --- |
| **Internacionalização do software** |
| <coloque aqui a correlação de todos os termos utilizados na interface gráfica nos idiomas Português e Inglês, conforme abaixo:>   |  |  | | --- | --- | | **Termos utilizados para a Internacionalização** | | | **Português** | **Inglês** | | Cadastrar Cliente | Register Customer | | Você deseja continuar? | Do you want to continue? | | ... | … | | ... | … | | ... | … | |

# Projeto de Banco de Dados

|  |
| --- |
| **Diagrama de Entidade e Relacionamento (DER)** |
|  |

|  |
| --- |
| **Diagrama de Banco de Dados** |
|  |

|  |
| --- |
| **Quadro Resumo dos Objetos de Banco de Dados** |
| |  |  | | --- | --- | | **Objeto de Banco de Dados** | **Quantidade** | | **Tabelas** | <preencher> | | **Views** | <preencher> | | **Procedures** | <preencher> |   <Aqui vc precisa colocar qual é a quantidade que foi feita pelo grupo para cada objeto acima. |

|  |
| --- |
| **Mapeamento para o Modelo Lógico (Modelo Relacional)** |
| Pessoas (id, nome, cpf, status).  Clientes (#pessoa\_id, renda, credito).  Vendedores (#pessoa\_id, salario).  Pedidos (nr, data, valor, status, #vendedor\_id, #cliente\_id).  Produtos (codigo, descricao, estoque, valor, status).  Itens\_pedidos (#pedido\_nr, #produto\_codigo, qtd, valor).  <Colocar todas as tabelas do banco de dados> |

|  |
| --- |
| **Dicionário de Dados** |
| |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Tabela** | **Funcionarios** | | | | | | | | | | | **Colunas** | **Tipo** | **Tamanho** | **Precisão** | **Escala** | **Null** | **PK** | **FK** | **UK** | **Referência** | **Check** | | id | int | 4 | 10 | 0 | Não | Sim | Não | Sim | -o- | -o- | | nome | varchar | 50 | -o- | -o- | Não | Não | Não | Não | -o- | -o- | | salario | decimal | 5 | 10 | 2 | Sim | Não | Não | Não | -o- | > 500 | | cpf | varchar | 12 | -o- | -o- | Não | Não | Não | Sim | -o- | -o- | | status | int | 4 | 10 | 0 | Sim | Não | Não | Não | -o- | (1,2,3,4) | | dep\_id | int | 4 | 10 | 0 | Não | Não | Sim | Não | Departamentos | -o- |   <fazer para cada tabela do banco de dados> |

|  |
| --- |
| **Script de Banco de Dados** |
| **create** **database** vendas  go  **use** vendas  go  **create** **table** pessoas  (  id int not null primary key identity,  nome varchar(50) not null,  cpf varchar(12) not null unique,  status int check(status in (1,2,3,4))  )  go  **create** **table** clientes  (  pessoa\_id int not null primary key,  renda decimal(7,2),  credito decimal(7,2),  foreign key(pessoa\_id) references pessoas(id)  )  go  **create** **table** vendedores  (  pessoa\_id int not null primary key,  salario decimal(7,2) check(salario > 0),  foreign key(pessoa\_id) references pessoas(id),  )  go  **create** **table** pedidos  (  nr int not null primary key identity,  data datetime,  valor decimal(7,2) check(valor > 0),  status int check(status in (1,2,3,4)),  vendedor\_id int not null,  cliente\_id int not null,  foreign key(cliente\_id) references clientes(pessoa\_id),  foreign key(vendedor\_id) references  vendedores(pessoa\_id)  )  go  **create** **table** produtos  (  id int not null primary key identity,  descricao varchar(50) not null,  estoque int,  valor decimal(7,2),  status int check(status in (1,2,3,4))  )  go  **create** **table** itens\_pedidos  (  pedido\_nr int not null,  produto\_codigo int not null,  qtd int not null,  valor decimal(7,2) not null,  primary key(pedido\_nr, produto\_codigo),  foreign key(pedido\_nr) references pedidos(nr),  foreign key(produto\_codigo) references produtos(codigo)  )  go |

|  |
| --- |
| **Inserção no banco de dados** |
| Colocar aqui os comandos de inserts (no mínimo três) por tabela do banco de dados. Por exemplo:  -- Tabela Produtos –  Insert into produtos values ('CD', 1000, 1.2, 1);  Insert into produtos values ('DVD', 1000, 1.5, 1);  Insert into produtos values ('Blu-ray', 1000, 2.0, 1);  Pode-se utilizar as procedures para demonstrar a inserção no banco de dados ao invés do comando de insert. |

|  |
| --- |
| **Views** |
| **Colocar aqui as view considerando no mínimo:**  **- Uma view para cada tabela do banco de dados. Em uma herança faça as views somente das entidades filhas com junção com a entidade pai. Ou seja, não deve-se fazer uma view somente para a entidade pai.**  **- 7 views com junção (join). Considere as junções mais importantes a serem utilizadas na aplicação.**  **- Não se esqueça de colocar o teste de execução.**  **- Numere as views conforme abaixo.**  Faça conforme o exemplo abaixo:  1. Visualização da tabela produtos.  create view v\_produtos  as  select prd\_codigo Codigo,  prd\_descricao Descricao,  prd\_qtd Estoque,  prd\_valor Preco\_Venda,  prd\_status Status  from produtos  -- Teste de execução –  Select \* from v\_produtos |

|  |
| --- |
| **Procedures** |
| Colocar aqui as procedures considerando no mínimo:  - uma procedure para inserção (insert) para cada tabela do banco de dados.  - uma procedure para atualização (update) para cada tabela do banco de dados.  - Não se esqueça de colocar o teste de execução.  - A procedure para a entidade pai na herança não deve ser feita.  - Faça sempre na ordem. Uma para inserir e outra para atualizar a mesma tabela.  - Não esqueça de numerar conforme abaixo.  Faça conforme o exemplo abaixo:  1. Procedure para cadastrar um produto na tabela produtos.  create procedure prodAdd  (  @codigo int,  @descricao varchar(50),  @qtd int,  @valor int,  @status int = 1 -- o valor 1 foi indicado como padrão.  )  as  begin  insert into produtos values (@descricao, @qtd, @valor, @status)  end  -- Teste de execução –  prodAdd 'CD', 1000, 1.2, 1  2. Procedure para atualizar um produto na tabela produtos.  create procedure prodAlt  (  @codigo int,  @descricao varchar(50),  @qtd int,  @valor int,  @status int = 1 -- o valor 1 foi indicado como padrão.  )  as  begin  update produtos set  prd\_descricao = @descricao,  prd\_qtd = @qtd,  prd\_valor = @valor,  prd\_status = @status  where prd\_codigo = @codigo  end  -- Teste de execução –  prodAlt 'DVD', 100, 1.5, 1 |

# Considerações Finais

|  |
| --- |
| **Dificuldades Encontradas** |
| <Faça aqui uma conclusão do trabalho expondo as **dificuldades e problemas enfrentados>** |

|  |
| --- |
| **Sugestões de Melhoria** |
| <Descreva aqui alguma sugestão para melhoria do projeto interdisciplinar no sentido de proporcionarmos a melhoria continua de todo o processo> |

# Referências

|  |
| --- |
|  |
| <colocar referências no padrão ABNT sobre C# (Dot Net), Banco de dados, Engenharia de Software, dentre outras. |