|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | **Código: EC-E-10504**  **Disciplina: Linguagem de Programação I** N1 | 2º bimestre| Curso: EC | Turma: 522/05/2024 - 19h15 Prof.(ª): Eduardo R. Marcelino | Coord.: Rodrigo Fontes | |  |
| Aluno (a): ­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­ | Nº: | | RA: | Nota: |
| Marcos Felipe C. Soares  Kauê de Souza Silva |  | | 081220020/081220003 |  |
| Orientações: Respostas a lápis não são passíveis de revisão. A interpretação faz parte da prova. SEM CONSULTA. Indique quando não resolver no local predeterminado. A identação do código será avaliada. | | | |  |
| *Data da vista da prova: 27* / 05 / 2024 *- horário: 19* h 15.  Declaro estar de acordo com a nota e com a revisão e correção realizada pelo professor.  O não comparecimento do aluno na vista dos instrumentos avaliativos, na data prevista, significa a concordância tácita com as notas atribuídas, desobrigando a instituição de deferir eventuais pedidos de revisão de notas (após a divulgação de notas no portal do aluno, dentro do semestre letivo ou em situações futuras). | | | | Rubrica do aluno: |

TRABALHO

∟ Entregar este documento preenchido via moodle

∟ O código fonte do projeto e o Scripts SQL para criar a(s) tabela(s) no banco de dados , tudo pelo moodle.

Tema livre: o aluno pode escolher qualquer tipo de sistema desde que atenda aos requisitos abaixo:

O trabalho deve possuir – **obrigatoriamente** - as seguintes características:

* Pelo menos 3 CRUDs, sendo que todos deverão possuir relacionamento com outra tabela. As listagens de tabelas que possuam relacionamento devem fazer o “join” e exibir o campo relacionado.
* Utilização de herança nas classes, como explicado em aula.
* Ao menos 1 dos cadastros deverá manipular imagem (ex: foto do funcionário, do produto).
* Área “Sobre”, exibindo informações acerca do sistema, além do nome e o RA dos alunos.
* Uma tela com a funcionalidade principal do sistema. Esta é geralmente a tela mestre-detalhe, onde há um cadastro que envolve, no mínimo, três entidades do banco de dados. Exemplos:
  + Se for um sistema de controle de padaria, deverá haver a tela de venda, onde o cliente deverá informar a data da venda, código do cliente (quando houver), produtos adquiridos, etc.
  + Se for um sistema de digitação de notas, esta tela seria a tela onde as notas seriam informadas, e teria também uma funcionalidade para calcular as médias. Neste caso, deveria ser possível consultar as notas por avaliações (N1, N2, etc), por data da avaliação, por professor, disciplina, aluno, etc.
  + Se for um sistema de álbum de figurinha, essa funcionalidade principal poderia ser a tela onde o usuário iria montar o álbum com base em figurinhas que ele ganhou, comprou ou trocou.
  + Qualquer sistema que contenha pedido e item do pedido atendera este requisito.
  + A tabela “detalhe” deve ser composta por mais de um campo além de uma possível FK. Exemplo: Se for ProdutosDaVenda, deve possuir além do código do produto algum outro campo, como por exemplo a quantidade de produtos vendidos.
  + Utilize controle de transação
* Duas telas de consulta, utilizando AJAX que permitam alguma diversidade de filtros (mínimo 2 filtros por tela) e exibam os dados em tabelas (Dados gerenciais). **Exemplos**:
  + Consulta de produtos vendidos com opção de filtro por data, produto, cliente, fornecedor, etc..
  + Consulta de clientes e logradouros com opção de filtro por estado, cidade e bairro, etc...
  + Top 10 produtos mais vendidos, com opção de filtro por período, categoria, etc...
* Controle de acesso com login e senha, com opção para que o usuário se cadastre e com opção para cadastrar usuários administradores do sistema.
* Área administrativa e área do cliente. Exiba as telas de acordo com o perfil do usuário logado.
* O sistema deve possuir uma página inicial (Home).
* Todos os conceitos e boas práticas de programação que já foram tratadas em outras disciplinas devem ser aplicadas aqui, como identação do código, controle de exceção, orientação a objetos, complexidade ciclomática, etc.

* **Entregar o projeto via Moodle**
* Deverá ser entregue o código fonte do trabalho e o script de criação do banco de dados.
* Descreva Abaixo as atividades que cada aluno desenvolveu:

|  |  |
| --- | --- |
| Aluno | Conteúdo(s) desenvolvidos |
| Marcos Felipe Correa Soares | Implementação dos CRUDs de Produto e Review, Consultas Avançadas, Página Sobre, Uso de AJAX para Consultas Avançadas e consumo de APIs. |
| Kauê de Souza Silva | Modelagem do banco de dados, Implementação do CRUD de Usuário, implementação de pedidos, Tela de Login, HTML e CSS, separação entre admin e cliente |
|  |  |

**Formas de Avaliação/Desenvolvimento do Trabalho**

Serão avaliados, além dos pré-requisitos já mencionados anteriormente, os seguintes aspectos:

* Organização do código;
* Utilização dos recursos oferecidos pela linguagem com o objetivo de diminuir o código, facilidade de manutenção, utilização de memória, eficiência do código, etc.;
* Facilidade de utilização (usabilidade);
* Desenvolvimento das funcionalidades pré-definidas neste documento.
* Validações e controle de exceção.
* Modelo de dados e instruções SQL.
* Interface gráfica e usabilidade do sistema.

Dúvidas deverão ser sanadas diretamente com os professores das disciplinas.

**Trabalhos iguais serão anulados.**

|  |  |  |
| --- | --- | --- |
| **Tópico** | **Nota máxima** | **NOTA** |
| **Home e sobre** | 0,5 |  |
| **Telas de cadastro, utilizando herança** | 0,5 |  |
| **Utilização de FK nas telas de cadastro, com Join nas listagens** | 0,5 |  |
| **CSS, HTML 5 e Javascript** | 0,5 |  |
| **Utilização de imagem (salvar/recuperar/exibir)** | 0,5 |  |
| **Utilização de Ajax em 2 locais** | 2,0 |  |
| **Telas de consulta avançada com filtros e joins** | 2,0 |  |
| **Mestre detalhe** | 1,0 |  |
| **Controle de transação** | 0,5 |  |
| **Login com opção de cadastro** | 1,0 |  |
| **Separação da área do cliente e área administrativa** | 1,0 |  |