

TÉCNICAS DE PROGRAMAÇÃO 1

TRABALHOS

INTRODUÇÃO

1. Desenvolva o software seguindo os requisitos especificados.
2. Atente para as instruções quanto à arquitetura e as responsabilidades dos módulos.
3. Use a linguagem C++.
4. Preencha os documentos com clareza e tome cuidado com a ortografia.
5. Adote uma convenção de codificação (identifique a convenção adotada no seu trabalho).
6. Forneça os códigos em formatos fonte e executável.
7. Forneça textos e diagramas em formato PDF.
8. Forneça os artefatos produzidos em um CD adequadamente organizado (diretórios).
9. Inclua na raiz do CD um arquivo LEIAME.PDF onde define o objetivo de cada diretório.
10. Identifique o CD com as matrículas dos membros da equipe (no arquivo LEIAME.PDF e escrito na capa do CD).

REQUISITOS FUNCIONAIS

O sistema a ser desenvolvido implementará estantes virtuais para os seus usuários (vide exemplo dessa classe de sistema de informação em <https://www.skoob.com.br/>). Para acessar o sistema, o usuário precisará estar cadastrado. Para se cadastrar, o usuário precisará informar os seguintes dados: nome, apelido, telefone e senha. Para acessar o sistema, o usuário deverá informar apelido e senha. Cada usuário terá uma estante virtual onde poderá armazenar até 10 (dez) exemplares de livros. Na estante virtual de um usuário, não pode haver mais de um exemplar de um livro. Acerca de cada livro, serão armazenados os seguintes dados: título, nome do autor, data de publicação, código e gênero literário. Os dados acerca de um livro serão informados quando um exemplar do livro for incluído pela primeira vez em uma estante virtual, e serão excluídos quando removidos todos os exemplares do livro. Uma vez autenticado, o leitor terá acesso aos seguintes serviços: consultar dados de livro, incluir exemplar em estante, remover exemplar de estante, criar resenha, informar se deseja trocar exemplar e procurar usuário. A consulta a dados de livro será por título e resultará na seguinte informação: título, autor, data de publicação, gênero literário e resenhas acerca do livro. A procura por troca será por título e resultará em lista de usuários com exemplares e que desejam a troca. A procura por usuário será por apelido e resultará na seguinte informação: nome, apelido e telefone do usuário.

DOMÍNIOS

NOME	15 CARACTERES (LETRA, ESPAÇO EM BRANCO, PONTO)
APELIDO (CHAVE)	5 LETRAS
TELEFONE	(XX) – XXXXXXXXXX (X É ALGARISMO DE 0 A 9)
SENHA	4 LETRAS (NÃO PODE OCORRER REPETIÇÃO DE LETRA)
TÍTULO	20 CARACTERES (NÃO PODE HAVER ESPAÇO EM BRANCO DUPLICADO)
DATA	DD/MM/AA (DD É VALOR ENTRE 00 E 31, MM ENTRE 01 E 12, AA ENTRE 00 E 99)
CÓDIGO (CHAVE)	XXXXXX (X É ALGARISMOS DE 0 A 9)
TEXTO	40 CARACTERES
GÊNERO LITERÁRIO	EPOPEIA, NOVELA, CONTO, ENSAIO, ROMANCE

ENTIDADES

USUÁRIO, LIVRO, RESENHA

REQUISITOS NÃO FUNCIONAIS

1. A arquitetura do software é composta por camadas de apresentação e serviço.
2. A camada de apresentação é responsável pela interface com o usuário e pela validação de domínios.
3. A camada de serviço é responsável pela lógica de negócio (imposição de regras) e armazenamento.
4. A arquitetura do software é composta por módulos.
5. Os relacionamentos entre módulos ocorrem por meio de interfaces.
6. As interfaces devem ser definidas por meio de classes abstratas.
7. A interface com o usuário pode ser orientada a texto (TUI) ou gráfica (GUI).

TÉCNICAS DE PROGRAMAÇÃO 1

TRABALHO 1

MATRÍCULAS:

NOTA:

ATIVIDADES

1. Codificar as classes que representam os domínios.
2. Codificar as classes que representam as entidades.
3. Codificar um teste de unidade para cada classe de domínio (um cenário de sucesso e um de falha).
4. Codificar um teste de unidade para cada classe de entidade (apenas cenário de sucesso).
5. Produzir documentação para as classes implementadas usando Doxygen.

QUESTÕES

- | | | |
|----|---|---|
| 1. | Cada classe de domínio armazena os seus atributos. | <input type="checkbox"/> 0 <input type="checkbox"/> 5 <input type="checkbox"/> 10 |
| 2. | Cada classe de domínio valida os seus atributos e lança exceção em caso de falha. | <input type="checkbox"/> 0 <input type="checkbox"/> 5 <input type="checkbox"/> 10 |
| 3. | Cada classe de entidade armazena os seus atributos. | <input type="checkbox"/> 0 <input type="checkbox"/> 5 <input type="checkbox"/> 10 |
| 4. | Cada teste de unidade consiste de uma classe. | <input type="checkbox"/> 0 <input type="checkbox"/> 5 <input type="checkbox"/> 10 |
| 5. | Cada teste de unidade contém diferentes métodos para diferentes casos de teste. | <input type="checkbox"/> 0 <input type="checkbox"/> 5 <input type="checkbox"/> 10 |
| 6. | Classes de domínio funcionam corretamente segundo os testes de unidade. | <input type="checkbox"/> 0 <input type="checkbox"/> 5 <input type="checkbox"/> 10 |
| 7. | Classes de entidade funcionam corretamente segundo os testes de unidade. | <input type="checkbox"/> 0 <input type="checkbox"/> 5 <input type="checkbox"/> 10 |
| 8. | Fornecida documentação das classes implementadas usando Doxygen. | <input type="checkbox"/> 0 <input type="checkbox"/> 5 <input type="checkbox"/> 10 |

QUESTÃO	COMENTÁRIO
1	
2	
3	
4	
5	
6	
7	
8	

TÉCNICAS DE PROGRAMAÇÃO 1

TRABALHO 2

MATRÍCULAS :

NOTA:

ATIVIDADES

1. Construir modelo composto por diagrama de arquitetura do software no nível de módulos.
2. Codificar camada de apresentação.
3. Codificar interfaces entre camada de apresentação e camada de serviço.
4. Codificar classes stub para a camada de serviço.
5. Produzir documentação para as classes implementadas usando Doxygen.

QUESTÕES

- | | |
|--|---|
| 1. Diagrama da arquitetura do software no nível de módulos de software. | <input type="checkbox"/> 0 <input type="checkbox"/> 5 <input type="checkbox"/> 10 |
| 2. Descrição de cada módulo de software e de cada interface entre módulos de software. | <input type="checkbox"/> 0 <input type="checkbox"/> 5 <input type="checkbox"/> 10 |
| 3. Codificação das classes da camada de apresentação. | <input type="checkbox"/> 0 <input type="checkbox"/> 5 <input type="checkbox"/> 10 |
| 4. Codificação das interfaces entre as camadas de apresentação e de serviço. | <input type="checkbox"/> 0 <input type="checkbox"/> 5 <input type="checkbox"/> 10 |
| 5. Codificação das classes stub da camada de serviço. | <input type="checkbox"/> 0 <input type="checkbox"/> 5 <input type="checkbox"/> 10 |
| 6. Camada de apresentação funciona corretamente com <i>stubs</i> de serviço. | <input type="checkbox"/> 0 <input type="checkbox"/> 5 <input type="checkbox"/> 10 |
| 7. Fornecida documentação das classes implementadas usando Doxygen. | <input type="checkbox"/> 0 <input type="checkbox"/> 5 <input type="checkbox"/> 10 |

QUESTÃO	COMENTÁRIO
1	
2	
3	
4	
5	
6	
7	

TÉCNICAS DE PROGRAMAÇÃO 1

TRABALHO 3

MATRÍCULAS :

NOTA:

ATIVIDADES

1. Codificar classes da camada de serviço (implementação de armazenamento persistente é opcional).
2. Codificar interfaces entre as camadas de apresentação e de serviço.
3. Integrar as camadas do sistema.
4. Elaborar *smoke test*.
5. Produzir documentação para as classes implementadas usando Doxygen.

QUESTÕES

- | | | |
|----|---|---|
| 1. | Codificação das classes da camada de serviço. | <input type="checkbox"/> 0 <input type="checkbox"/> 5 <input type="checkbox"/> 10 |
| 2. | Codificação das interfaces entre as camadas de apresentação e de serviço. | <input type="checkbox"/> 0 <input type="checkbox"/> 5 <input type="checkbox"/> 10 |
| 3. | Fornecido <i>smoke test</i> . | <input type="checkbox"/> 0 <input type="checkbox"/> 5 <input type="checkbox"/> 10 |
| 4. | Código integrado funciona corretamente. | <input type="checkbox"/> 0 <input type="checkbox"/> 5 <input type="checkbox"/> 10 |
| 5. | Fornecida documentação das classes implementadas usando Doxygen. | <input type="checkbox"/> 0 <input type="checkbox"/> 5 <input type="checkbox"/> 10 |

QUESTÃO	COMENTÁRIO
1	
2	
3	
4	
5	