

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).

ESCOLHA UM DOS SEGUINTE SISTEMAS PARA DESENVOLVIMENTO

SISTEMA 1

O sistema proverá um serviço de *blog* simplificado. A cada *blog* é associado o nome do *blog* e o nome do autor do *blog*. A cada *blog* são associadas *postagens* (*posts*). Cada usuário pode submeter, no máximo, até cinco comentários para uma mesma postagem. A cada postagem podem ser associados, caso autorizados, comentários. A cada postagem, é associada uma avaliação (nota de 1 a 5) dos usuários, que é a média das avaliações até o momento. A avaliação de cada postagem pode ser vista por qualquer usuário. Para usar certos serviços, o usuário precisa se autenticar no sistema. Para se autenticar, o usuário precisa informar email e senha. Para se cadastrar no sistema, o usuário deve informar: nome, senha e e-mail. Sem se autenticar, cada usuário tem acesso aos seguintes serviços: listar nomes dos *blogs*; listar postagens associadas a qualquer *blog*; listar comentários associados a qualquer postagem. Uma vez autenticado, além de ter acesso aos serviços anteriormente relacionados, o usuário também tem acesso aos seguintes serviços: alterar a sua senha; excluir a sua conta; criar ou excluir *blog*; criar, remover ou editar postagens no seu *blog*; criar comentários associados a postagens em qualquer *blog* (desde que comentários estejam autorizados); remover postagens de sua autoria; remover comentários associados às suas postagens; avaliar postagens de outros usuários; encerrar sua conta (ao ser encerrada uma conta, são removidos *blog*, postagens e comentários associados à conta).

DOMÍNIOS

NOME	20 CARACTERES (LETRAS A-Z E ESPAÇO EM BRANCO)
SENHA	5 CARACTERES QUAISQUER (SEM REPETIÇÃO DE CHARACTER)
EMAIL	FORMATO L@L.L (ONDE L SÃO LETRAS)
AValiação	1, 2, 3, 4, 5
TEXTO (POSTAGEM OU COMENTÁRIO)	50 CARACTERES

POSSÍVEIS ENTIDADES

USUÁRIO, BLOG, POSTAGEM, COMENTÁRIO

SISTEMA 2

O sistema possibilita o desenvolvimento de vocabulários controlados. A cada vocabulário é associado o nome do vocabulário e o nome do autor do vocabulário. Cada vocabulário controlado é composto por termos. Em um vocabulário controlado podem existir até 100 termos. A cada termo podem ser associadas até cinco definições. A cada definição, é associada uma avaliação (nota de 1 a 5) dos usuários, que é a média das avaliações até o momento. A avaliação de cada definição pode ser vista por qualquer usuário. Para usar certos serviços, o usuário precisa se autenticar no sistema. Para se autenticar, o usuário precisa informar email e senha. Para se cadastrar no sistema, o usuário deve informar: nome, senha e e-mail. Sem se autenticar, cada usuário tem acesso aos seguintes serviços: listar nomes dos vocabulários; listar termos associados a qualquer vocabulário; listar definições associadas a qualquer termo. Uma vez autenticado, além de ter acesso aos serviços anteriormente relacionados, o usuário também tem acesso aos seguintes serviços: alterar a sua senha; excluir a sua conta; criar ou excluir vocabulário controlado; criar, remover ou editar termos no seu vocabulário; criar definições associadas a termos no seu vocabulário; avaliar

definições de outros usuários; encerrar sua conta (ao ser encerrada uma conta, são removidos vocabulário, termos e definições associadas à conta).

DOMÍNIOS

NOME	20 CARACTERES (LETRAS A-Z E ESPAÇO EM BRANCO)
SENHA	5 CARACTERES QUAISQUER (SEM REPETIÇÃO DE CHARACTER)
EMAIL	FORMATO L@L.L (ONDE L SÃO LETRAS)
AValiação	1, 2, 3, 4, 5
TEXTO (TERMO OU DEFINIÇÃO)	50 CARACTERES

POSSÍVEIS ENTIDADES

USUÁRIO, VOCABULÁRIO, TERMO, DEFINIÇÃO

REQUISITOS NÃO FUNCIONAIS

1. A arquitetura do software é composta por camadas de apresentação, negócio e persistência.
2. A camada de apresentação é responsável pela interface com o usuário e pela validação de domínios.
3. A camada de negócio é responsável pela lógica de negócio (imposição de regras).
4. A camada de persistência é responsável pelo armazenamento em memória secundária.
5. A arquitetura do software é composta por módulos.
6. Os relacionamentos entre módulos ocorrem por meio de interfaces.
7. As interfaces devem ser definidas por meio de classes abstratas.
8. 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 e para cada classe de entidade.
4. Produzir documentação para domínios e entidades.

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 classe de entidade não possibilita armazenamento de atributo com formato inválido. | <input type="checkbox"/> 0 <input type="checkbox"/> 5 <input type="checkbox"/> 10 |
| 5. | Cada teste de unidade consiste de uma classe. | <input type="checkbox"/> 0 <input type="checkbox"/> 5 <input type="checkbox"/> 10 |
| 6. | Cada teste de unidade contém diferentes métodos para diferentes casos de teste. | <input type="checkbox"/> 0 <input type="checkbox"/> 5 <input type="checkbox"/> 10 |
| 7. | Cada teste de unidade contém um caso de teste com argumento válido e outro com inválido. | <input type="checkbox"/> 0 <input type="checkbox"/> 5 <input type="checkbox"/> 10 |
| 8. | Classes de domínio funcionam corretamente segundo os testes de unidade. | <input type="checkbox"/> 0 <input type="checkbox"/> 5 <input type="checkbox"/> 10 |
| 9. | Classes de entidade funcionam corretamente segundo os testes de unidade. | <input type="checkbox"/> 0 <input type="checkbox"/> 5 <input type="checkbox"/> 10 |
| 10. | Fornecida documentação das classes de domínio e das entidades. | <input type="checkbox"/> 0 <input type="checkbox"/> 5 <input type="checkbox"/> 10 |

QUESTÃO	COMENTÁRIO
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	

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 classes da camada de apresentação.
3. Codificar classes da camada de negócio.
4. Codificar classes da camada de persistência.
5. Codificar interfaces entre camadas.
6. Codificar classes stub para a camada de negócio.
7. Produzir documentação para as classes das camadas de apresentação, negócio e persistência.

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 classes da camada de negócio. | <input type="checkbox"/> 0 <input type="checkbox"/> 5 <input type="checkbox"/> 10 |
| 5. | Codificação das classes da camada de persistência. | <input type="checkbox"/> 0 <input type="checkbox"/> 5 <input type="checkbox"/> 10 |
| 6. | Codificação das interfaces entre camadas. | <input type="checkbox"/> 0 <input type="checkbox"/> 5 <input type="checkbox"/> 10 |
| 7. | Codificação das classes stub da camada de negócio. | <input type="checkbox"/> 0 <input type="checkbox"/> 5 <input type="checkbox"/> 10 |
| 8. | Camada de apresentação funciona corretamente com <i>stubs</i> de negócio. | <input type="checkbox"/> 0 <input type="checkbox"/> 5 <input type="checkbox"/> 10 |
| 9. | Código integrado funciona corretamente. | <input type="checkbox"/> 0 <input type="checkbox"/> 5 <input type="checkbox"/> 10 |
| 10. | Fornecida documentação das classes das camadas de apresentação, negócio e persistência. | <input type="checkbox"/> 0 <input type="checkbox"/> 5 <input type="checkbox"/> 10 |

QUESTÃO	COMENTÁRIO
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	