

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 possibilitará a construção de vocabulários controlados (vide exemplo dessa classe de sistema em <http://vocbench.uniroma2.it/>). Por meio do sistema, será possível construir vocabulários controlados estruturados como listas de termos. Para acessar o sistema, o usuário precisa criar uma conta. Existem os seguintes tipos de conta: leitor, desenvolvedor e administrador. Uma vez criada uma conta, para acessar o sistema, o usuário deve informar o seu endereço de correio eletrônico e a sua senha. Para criar uma conta de leitor, o usuário deve informar nome, sobrenome, endereço de correio eletrônico e senha. Os seguintes serviços estão disponíveis aos leitores: apresentar dados do usuário, editar dados do usuário, excluir conta, listar nomes dos vocabulários controlados, apresentar dados de vocabulário controlado, consultar termo, consultar definição de termo. Para criar uma conta de desenvolvedor, o usuário deve informar nome, sobrenome, data de nascimento, endereço de correio eletrônico e senha. Além dos serviços disponíveis aos leitores, os seguintes serviços estão disponíveis aos desenvolvedores: cadastrar como desenvolvedor de vocabulário controlado; criar, excluir, editar termo e definição de termo. Cada desenvolvedor pode se cadastrar como desenvolvedor de quantos vocabulários controlados queira. Cada vocabulário controlado pode ter de zero a dez desenvolvedores. Cada vocabulário controlado pode conter zero ou vários termos. Cada termo pertence a um vocabulário controlado. Quando um termo é criado, o desenvolvedor deve informar nome do termo, classe do termo e data. Cada termo pode ter de zero a cinco definições. Quando de uma definição, o desenvolvedor deve informar o texto da definição e data. Cada termo pode ter de zero a cinco definições. Cada definição está associada a um ou a vários termos. Quando só um termo está associado a uma definição, esse é o termo preferido (*preferred term*). Quando mais de um termo está associado a uma definição, um é o preferido e os outros são não preferidos (*non preferred term*). Para criar uma conta de administrador, o usuário deve informar: nome, sobrenome, telefone, data de nascimento, endereço, endereço de correio eletrônico e senha. Um administrador pode criar quantos vocabulários controlados queira. Quando um vocabulário controlado é criado, o administrador deve informar: nome do vocabulário controlado, texto de definição, idioma e data. Ao ser excluída a conta de um administrador, são excluídos os dados dos vocabulários administrados, os termos do vocabulário controlado e as definições apropriadas. Além dos serviços anteriormente listados, os seguintes serviços estão disponíveis para os administradores: editar definição de vocabulário controlado, alterar idioma de vocabulário controlado, excluir vocabulário controlado.

DOMÍNIOS

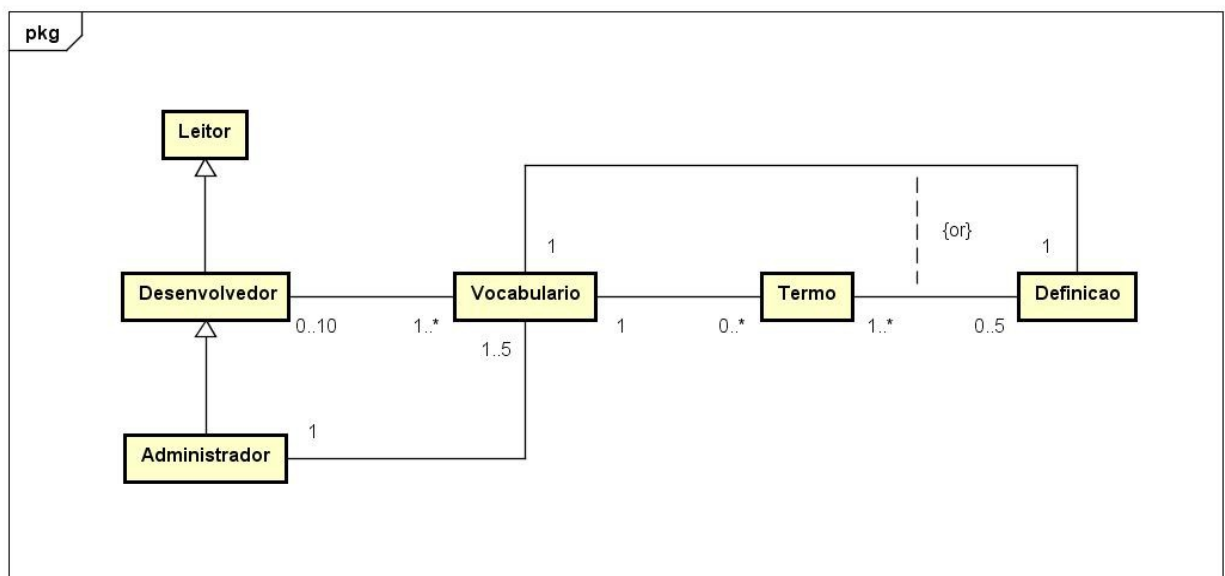
NOME	Contém de uma a vinte letras (A – Z a – z), sendo a primeira maiúscula.
SOBRENOME	Contém de uma a vinte letras (A – Z a – z), sendo a primeira maiúscula.
TELEFONE	Formato AA NNNNN–NNNN onde A e N são dígitos (0 – 9).
ENDEREÇO	Contém de uma a vinte letras (A – Z a – z). Pode conter espaços em branco. O primeiro e o último caracter não podem ser espaços em branco. Não podem haver espaços em branco em sequência (dois ou mais).
DATA	Formato DD/MM/AAAA onde DD consiste de número de 01 e 31,

	MM consiste de número de 01 e 12, AAAA consiste de número de 1900 a 2099. As datas devem considerar os anos que sejam bissextos.
ENDEREÇO DE CORREIO ELETRÔNICO	Composto por parte local e domínio separados por @. A parte local pode conter letras (A – Z a – z), dígitos (0 – 9) e caracteres (! # \$ % & ' * + - / = ? ^ _ ` { } ~ ;). Pode conter ponto (.) desde que não seja o primeiro ou o último caracter. O domínio pode conter letras (A – Z a – z); dígitos (0 – 9), desde que o domínio não seja só numérico; – (hífen), desde que não seja o primeiro ou o último caracter.
SENHA	Contém oito caracteres, que podem ser letras (A – Z a – z) ou dígitos (0 a 9). Cada senha precisa conter, pelo menos, uma letra maiúscula, uma letra minúscula e um dígito. Não pode conter o nome do seu usuário.
TEXTO DE DEFINIÇÃO	Contém de um a trinta caracteres quaisquer.
IDIOMA	ENG (inglês), FRA (francês), GER (alemão), ITA (italiano), POR (português) e SPA (espanhol).
CLASSE DE TERMO	PT (<i>preferred term</i>) e NP (<i>non preferred term</i>)

ENTIDADES

LEITOR	NOME SOBRENOME ENDEREÇO DE CORREIO ELETRÔNICO SENHA
DESENVOLVEDOR	NOME SOBRENOME DATA DE NASCIMENTO ENDEREÇO DE CORREIO ELETRÔNICO SENHA
ADMINISTRADOR	NOME SOBRENOME TELEFONE DATA DE NASCIMENTO ENDEREÇO ENDEREÇO DE CORREIO ELETRÔNICO SENHA
VOCABULÁRIO CONTROLADO	NOME IDIOMA DATA
TERMO	NOME CLASSE DE TERMO DATA
DEFINIÇÃO	TEXTO DE DEFINIÇÃO DATA

RELAÇÕES ENTRE ENTIDADES



powered by Astah

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 e lança exceções em caso de falhas. | <input type="checkbox"/> 0 <input type="checkbox"/> 5 <input type="checkbox"/> 10 |
| 2. | Cada classe de entidade armazena os seus atributos e esses são de classes de domínio. | <input type="checkbox"/> 0 <input type="checkbox"/> 5 <input type="checkbox"/> 10 |
| 3. | Cada teste de unidade é uma classe e contém diferentes métodos para diferentes casos de teste. | <input type="checkbox"/> 0 <input type="checkbox"/> 5 <input type="checkbox"/> 10 |
| 4. | Classes de domínio funcionam corretamente segundo os testes de unidade. | <input type="checkbox"/> 0 <input type="checkbox"/> 5 <input type="checkbox"/> 10 |
| 5. | Classes de entidade funcionam corretamente segundo os testes de unidade. | <input type="checkbox"/> 0 <input type="checkbox"/> 5 <input type="checkbox"/> 10 |
| 6. | 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	

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. Descrição da arquitetura no nível de módulos (diagrama, descrições de módulos e interfaces). | <input type="checkbox"/> 0 <input type="checkbox"/> 5 <input type="checkbox"/> 10 |
| 2. Codificação das classes da camada de apresentação. | <input type="checkbox"/> 0 <input type="checkbox"/> 5 <input type="checkbox"/> 10 |
| 3. Codificação das interfaces entre as camadas de apresentação e de serviço. | <input type="checkbox"/> 0 <input type="checkbox"/> 5 <input type="checkbox"/> 10 |
| 4. Codificação das classes stub da camada de serviço. | <input type="checkbox"/> 0 <input type="checkbox"/> 5 <input type="checkbox"/> 10 |
| 5. Camada de apresentação funciona corretamente com <i>stubs</i> de serviço. | <input type="checkbox"/> 0 <input type="checkbox"/> 5 <input type="checkbox"/> 10 |
| 6. 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	

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 segundo o <i>smoke test</i> . | <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	