Resumo Prova 1 - Engenharia de Software II

Vitor Lúcio de Oliveira

Baixo Acoplamento (Low Coupling)

- Refere-se a minimizar as dependências entre módulos ou componentes de um sistema
- Torna o código mais flexível e fácil de modificar
- Reduz o impacto de mudanças e testes, pois menos partes do sistema serão afetadas.
- Pode gerar excesso de indireção (muitas interfaces e abstrações desnecessárias).

Alta Coesão (High Cohesion)

- Refere-se a manter os elementos dentro de um módulo fortemente relacionados entre si, ou seja, um componente deve ter uma única responsabilidade bem definida.
- Facilita a manutenção e reuso do código.
- Pode levar a módulos muito pequenos e fragmentados.

Arquitetura Lógica (Conceitual e Organizacional)
Foca na organização do software, descrevendo os módulos, camadas e fluxos de dados independentemente da infraestrutura física.

Arquitetura Física (Infraestrutura e Implementação)
Foca na distribuição dos componentes em hardware e redes, considerando servidores, banco de dados. Depende da infraestrutura física ou virtual utilizada

Projeto Preliminar (Conceitual e Exploratório)
Definição inicial do sistema, focando na visão geral e nas principais decisões arquiteturais
Define os requisitos principais do sistema, e pode incluir prototipagem.

Projeto Detalhado (Técnica e Implementação)
Especificação completa do sistema, incluindo todos os detalhes necessários para o desenvolvimento e implementação. Contém diagramas detalhado.

Projeto de persistência

- Define como os dados serão armazenados, recuperados e gerenciados em um sistema. Ele especifica a estrutura do banco de dados, modelos de dados, estratégias de acesso e otimizações para garantir eficiência e segurança na manipulação das informações.
- O objetivo principal é garantir que os dados do sistema sejam armazenados de forma eficiente, consistente e segura, permitindo acesso rápido e confiável quando necessário.
- Etapas: Análise dos Requisitos de Dados, Escolha do Tipo de Banco de Dados, Modelagem de Dados, Implementação do Banco de Dados, Segurança e Controle de Acesso

DESIGN SYSTEM
É uma abordagem de projeto de interface e interação que padroniza, para os fornecedores, clientes e colaboradores de uma determinada organização, características como Layouts, componentes padrão, cores e etc. E desta forma todos os produtos web ou mobile daquela organização estarão seguindo os mesmos princípios em interfaces

Design pattern
A abordagem de reúso que utiliza abstrações genéricas, não incluindo detalhes de implementação, que mostram objetos abstratos e concretos e interações

Arquitetura de software

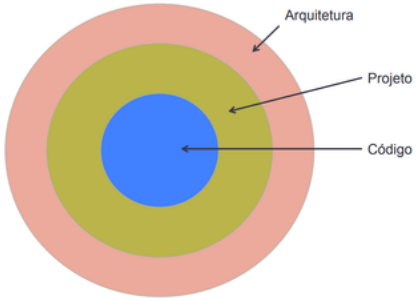
Sistemas grandes são decompostos em subsistemas que fornecem um conjunto de serviços relacionados.

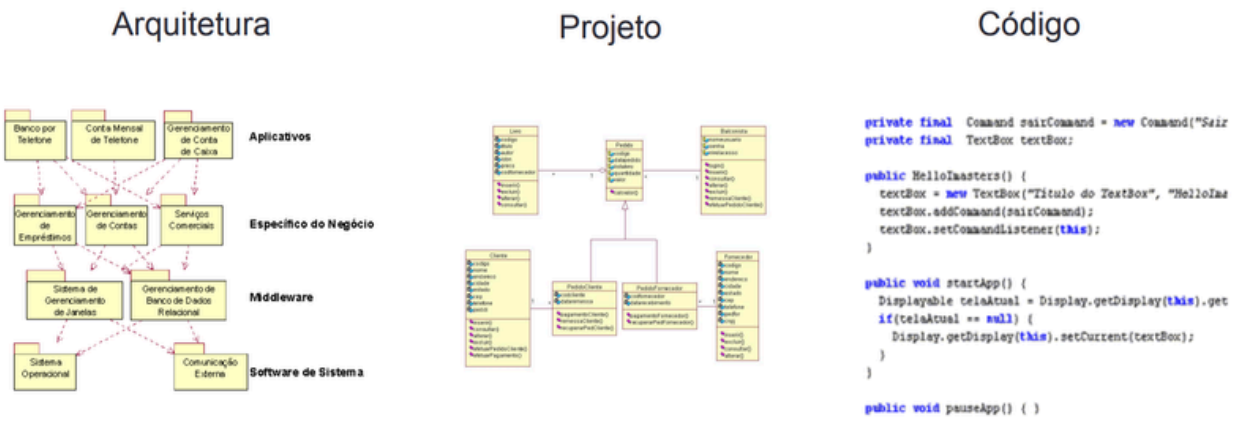
Uma **arquitetura de sistema** é a estrutura que engloba os subsistemas e componentes do software, definindo as suas propriedades visíveis externamente, o relacionamento entre eles e os mecanismos de controle

Além da estrutura tambem envolve a decisões dos engenheiros para atender aos requisitos, protocolos utilizados e funcionalidades dos componentes.

Foca em requisitos não-funcionais:

- Performance
- Confiabilidade
- Escalabilidade
- Testabilidade
- Manutenibilidade





Vantagens de projetar e documentar a arquitetura:

- Facilidade comunicação com stackholders
- Promover analise de requisitos
- Promover reutilização de software de outro sistema semelhante

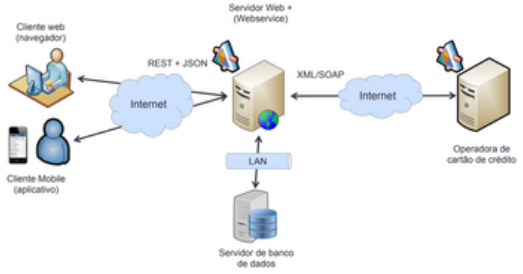
Impactos da escolha da arquitetura:

- Desempenho - estruturas menos granulares
- Manutenção - estruturas mais granulares
- Disponibilidade - estruturas que é possível substituir componentes sem para o sistema

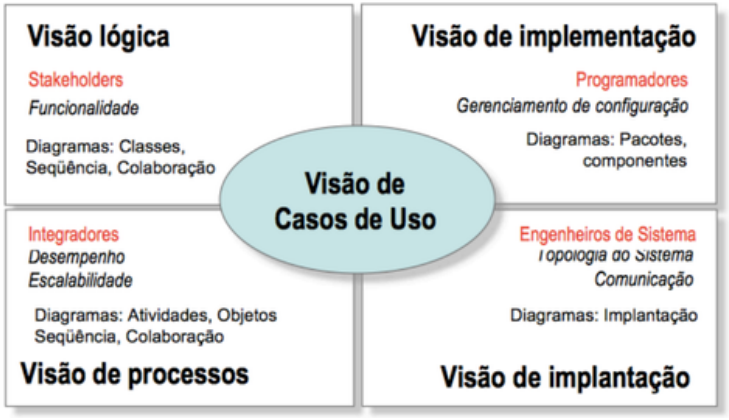
Documento de arquitetura:

- Stakeholders
- Representações gráficas da arquitetura
- Conexão entre os componentes
- Decisões tomadas(restrições)

Exemplo de representação gráfica



Visão de Casos de Uso

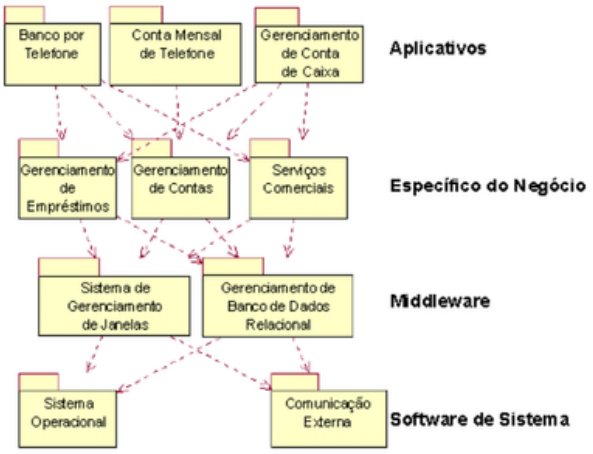


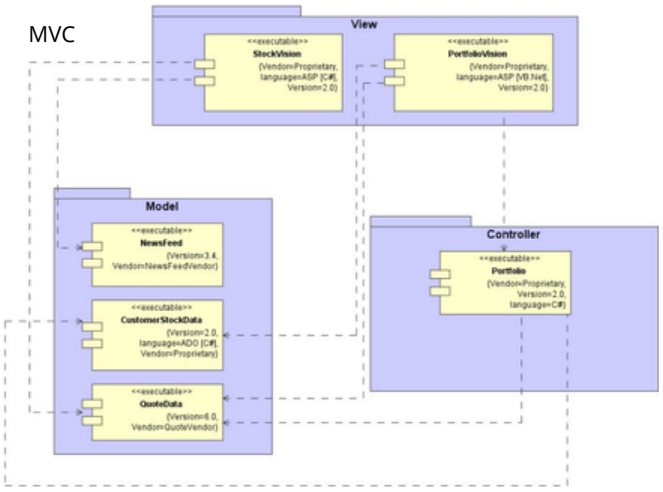
Visão logica:

- Descreve requisitos comportamentais e a decomposição do sistema em um conjunto de abstrações
- Diagramas de classes, sequência e colaboração

Visão de implementação:

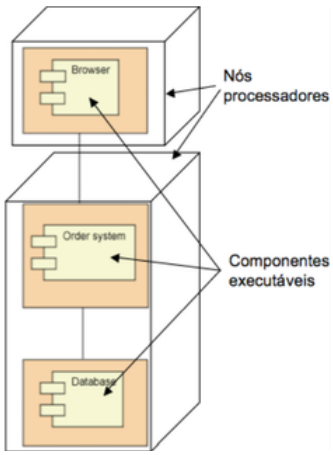
- Descreve os módulos do sistema
- Módulos são mais abstratos que classes e objetos





Visão de implantação (fisica):

- Descreve como a aplicação é instalada e como executa em uma rede de computadores
- Visão usada para avaliar requisitos nãofuncionais

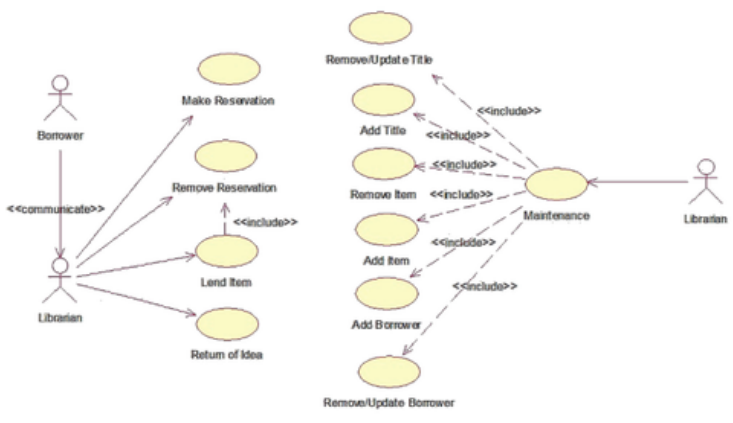


Visão de processo:

- Descreve os processos do sistema e como eles se comunicam
- Útil quando se tem múltiplos processos ou threads concorrentes
- Diagramas de atividades podem ser usados

Visão de casos de uso:

- Descreve a funcionalidade do sistema em termos de casos de uso

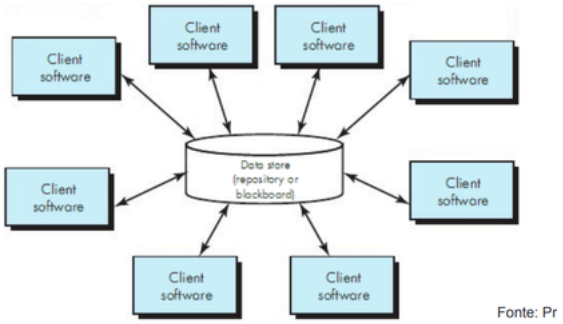


Estilos Arquiteturais

Problemas se repetem na engenharia de software. Então documentou-se grupos de estilos capazes de tratar problemas recorrentes.

Arquiteturas centrada em dados:

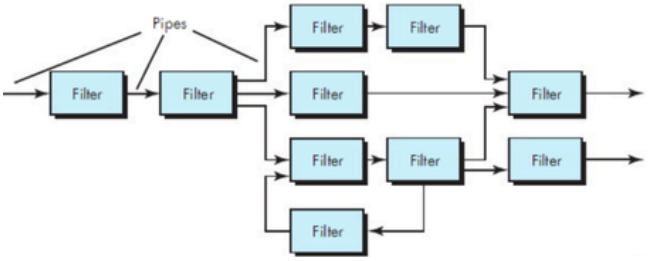
- Repositório de dados é o centro da arquitetura.
- Clientes (subsistemas, sistemas) acessam o repositório para inserir, recuperar, alterar, remover e também trocar dados entre eles.
- Promove a integração entre componentes e sistemas.
- Componentes e sistemas podem mudar, ou serem adicionados novos, sem impactar nos demais .



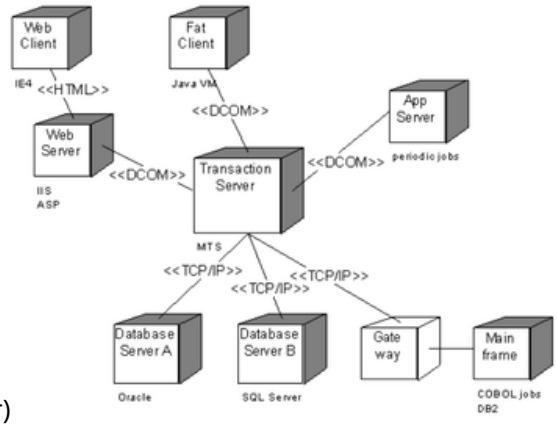
Fonte: Pr

Arquitetura de fluxo de dados:

- Dados de entrada são transformados em dados de saída através de uma série de cálculos e transformações.
- As entradas e saídas têm formato pré-definido.
- Um componente não precisa saber como os outros funcionam.



- **Sistema distribuído:**
- Módulos distintos cooperam
- Podem estar no mesmo nó ou em nós separados.
- Comunicação por protocolos abertos ou proprietários.
- Aplicação em sistemas tolerante a falhas.



Vantagens

- Distribuição de carga
- Reuso.
- Especialização

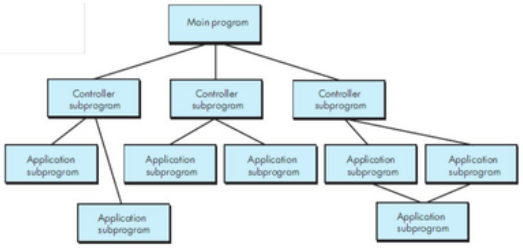
Desvantagem

- Performance/disponibilidade
- Segurança
- Especialização

Variação importante: Cliente-Servidor

- Cliente: interface com o usuário (solicita informação ao servidor)
- Servidor: serviço especializado (processamento, acesso a dados, regras de negócios)

- **Arquitetura de chamada e retorno:**
- Possibilitam uma estrutura relativamente fácil de modificar e escalar
- Arquitetura em camadas (OO)
  - Componentes do sistema são encapsulados em objetos
  - A comunicação e coordenação entre os componentes é realizado através da troca de mensagens (métodos dos objetos)
- Arquitetura de Programa principal/Subprogramas
  - Arquitetura hierarquizada
  - Programa principal utiliza subprogramas
  - Os subprogramas pode utilizar outros subprogramas



Atividade de Revisão

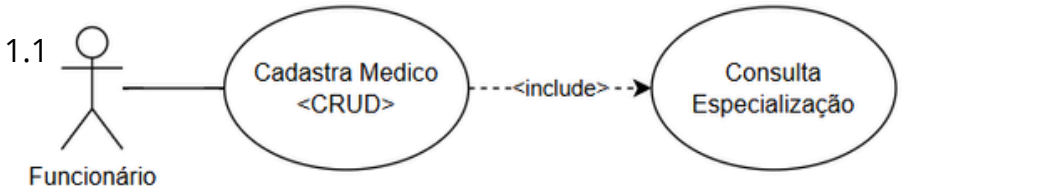
Seja uma clínica médica com médicos de diversas especializações (cada médico atua apenas em 1 especialização na clínica). A clínica agenda consultas para pacientes em seus consultórios. 1 médico atende em 1 consultório, mas cada consultório pode ser usado por N médicos de acordo com o horário.

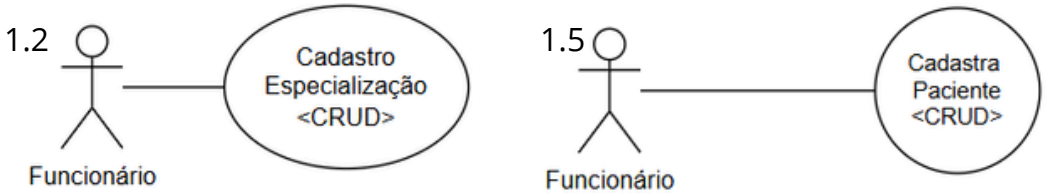
Temos os seguintes atributos:

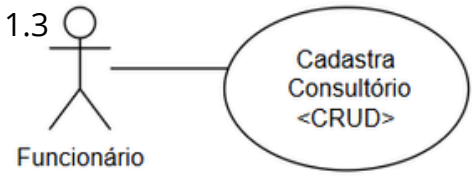
- **Médico** - CRM, nome, telefone, endereço;
- **Paciente** - CRM, nome, telefone, endereço, indicação;
- **Consultório** - número, andar;
- **Consulta** - número, data, horário, especialização;
- **Especialização** - id, nome

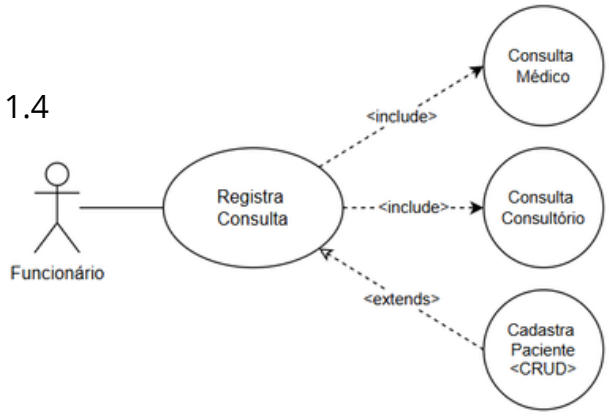
Questão 1: Cenários
1.1 cadastro de médico
1.2 cadastro de especialização
1.3 cadastro de consultório
1.4 registro de consulta
1.5 cadastro de paciente

Questão 2: Casos de Uso

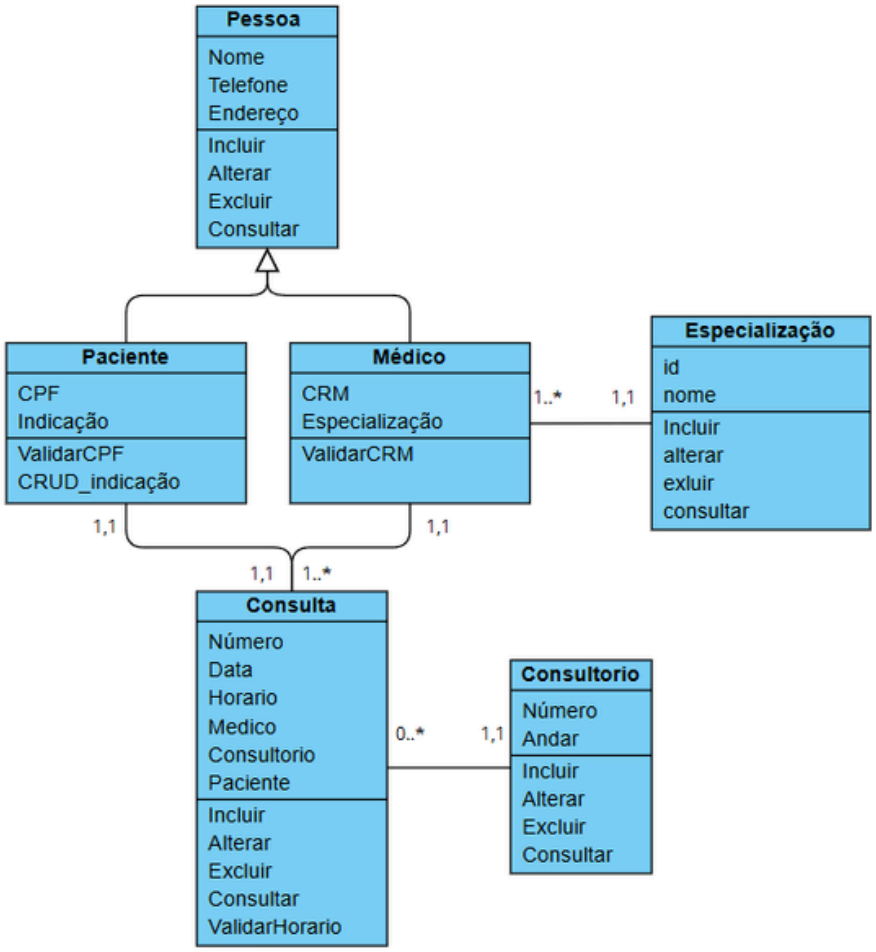






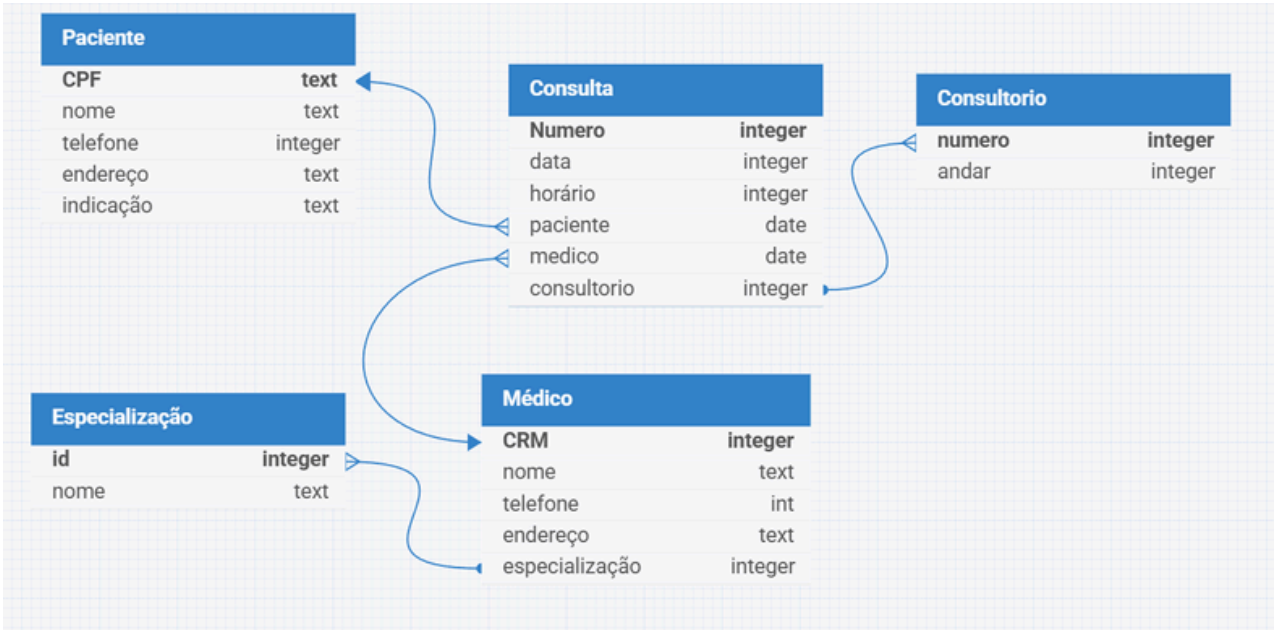


Questão 4: Diagrama de Classes



Questão 4: Diagrama de Robustez para Registro de Consulta

Questão 6: Modelo de Banco de Dados



Atividade de Revisão Canvas

Pergunta 1 0 / 0 pts

Considerando a prototipação nos projetos detalhados, assinale a afirmação FALSA.

- O desenvolvimento rápido e iterativo do protótipo é essencial para que os custos sejam controlados e os stakeholders possam experimentá-lo no início do processo de desenvolvimento do software.
- Mesmo não conseguindo capturar requisitos relacionados com desempenho e eficiência, os protótipos têm um bom emprego na elicitação e na validação dos requisitos do sistema.
- Um protótipo é usado para demonstrar conceitos, experimentar opções de projeto e descobrir mais sobre o problema e suas possíveis soluções.
- A adoção de práticas de qualidade no desenvolvimento do protótipo possibilita seu aproveitamento como parte do código do sistema final entregue ao cliente.

Correto! Protótipos devem ser descartados após serem demonstrados aos clientes.

Pergunta 2 0 / 0 pts

Os padrões BAIXO ACOPLAMENTO e ALTA COESAO sao:

- o contrario daquilo que deve ser feito
- nao se referem a projeto de software
- desnecessarios pois aumentam a dificuldade

Correto! necessarios em um bom projeto

Pergunta 3 0 / 0 pts

Uma forma de avaliar a usabilidade de um projeto de interface e interação é por meio do uso de métricas que funcionam como critérios de medição da usabilidade do sistema considerado. São critérios recomendáveis no projeto de interação:

- tempo médio entre falhas de máquina e tempo para atendimento
- tempo para realizar uma tarefa e para realizar documentação
- forma como estão distribuídas as tabelas no BD e seus relacionamentos

Correto! tempo consumido com erros e frequência de uso da ajuda

Pergunta 4

0 / 0 pts

Em relação ao desenho (*design* visual), que tem um impacto significativo na credibilidade e usabilidade de um *site*, é correto afirmar:

O fundo deve chamar mais atenção do que a informação, desde que seja relacionado ao tema do site. Um fundo de impacto imprime uma personalidade diferenciada ao site.

Correto!

A função do site e a informação, devem ser soberanas sobre o desenho. Qualquer tipo de conformação que beneficie o desenho em detrimento da informação, usabilidade e funcionalidade do site deve ser abandonada.

Todos os tipos de informação devem ser disponibilizados em uma longa lista sem mecanismos de classificação, pois o usuário pode localizar a informação desejada por meio da opção de busca do navegador.

Não se deve usar espaço em branco para separar conteúdos ou assuntos diferentes. Devem-se usar linhas grossas para permitir uma percepção melhor da separação de conteúdo.

Utilizar um projeto padrão de páginas passa a não ser necessário, uma vez que o usuário possui, por experiência, contato com uma grande diversidade de sites com diferentes desenhos.

Pergunta 5

0 / 0 pts

Projeto de subsistema composto por um conjunto de classes abstratas e concretas. Estabelece a arquitetura para aplicações em um domínio. Uma aplicação específica é construída a partir da criação de subclasses específicas para a aplicação, sendo essas subclasses das classes abstratas. A reutilização leva a uma inversão de controle.

A afirmação acima refere-se a:

Pacote

Correto!

Framework

Biblioteca de classes

Componente de software

Pergunta 6

0 / 0 pts

I) DESIGN SYSTEM é uma abordagem de projeto de interface e interação que padroniza, para os fornecedores, clientes e colaboradores de uma determinada organização, características como Layouts, componentes padrão, cores e etc. E desta forma todos os produtos web ou mobile daquela organização estarão seguindo os mesmos princípios em interfaces.

II) O uso de DESIGN SYSTEM além de padronizar, facilita a manutenção e redução de custos de projeto.

Correto!

As 2 afirmativas estão corretas.

As 2 afirmativas estão erradas.

Apenas a afirmativa I está correta.

Apenas a afirmativa II está correta.

Pergunta 7

0 / 0 pts

Diante da crescente demanda por automatização de processos de negócio, o gerente de desenvolvimento de sistemas de informação busca a maximização do reúso de software. A abordagem de reúso que utiliza abstrações genéricas, não incluindo detalhes de implementação, que mostram objetos abstratos e concretos e interações, é:

bibliotecas de programas.

framework de aplicação;

desenvolvimento baseado em componentes;

Correto!

design pattern;