



Informações úteis.

- O teste é sem consulta.
- O teste tem a duração de **2 horas**.
- Na avaliação das respostas são usados os habituais critérios de boa modelação: correcção, generalização, modularização, compreensão (o que inclui a legibilidade) e facilidade de evolução.
- Resolva as eventuais ambiguidades e omissões do enunciado da melhor forma, identificando-as e propondo a sua solução no próprio enunciado.
- Pode escrever a lápis ou a esferográfica, mas aconselha-se o uso de lápis e borracha.

Parte I

O objectivo é desenvolver o *FunParkGuide* para a Serra do Gerês, que permita aos visitantes explorar o que o Gerês tem para oferecer através de vários tipos de visitas guiadas. A ideia é propor um sistema que ofereça visitas guiadas moldadas aos gostos de cada um, adaptando o comportamento do sistema de acordo com o local geográfico onde o utilizador se encontra. O sistema terá por isso que ser sensível ao contexto (*context-sensitive*), de modo a guiar o turista durante a sua visita, garantindo a sua segurança, eliminando os riscos associados a certo tipo de passeios que podem durar horas por trilhos acidentados.

O *FunParkGuide* obtém informação sobre o local onde o turista se encontra e os pontos de interesse mais próximos através de uma rede *wireless* de alta velocidade instalada em vários pontos da Serra. Os visitantes terão por isso que transportar consigo um *tablet-PC* com *wireless*. O *FunParkGuide* deverá oferecer:

- Acesso a informação sobre o Gerês, fauna local, caminhadas, lagos, rios e cascatas, pontos de descanso, restaurantes típicos, incluindo a informação sobre a sua posição actual.
- Geração de visitas à medida baseadas nas preferências dos visitantes. Por exemplo, uma pessoa pode desejar apenas ver as cascatas, o lago e o rio, ou visitar Vilarinho das Furnas e incluir na visita uma paragem para as delícias gastronómicas da região, ou então estar interessado nas caminhadas que exijam passagens por trilhos difíceis, escaladas, etc.
- Capacidades de guia electrónico durante todo o percurso da visita, ajudando os visitantes a moverem-se de um local para outro, mostrando o ponto mais próximo a visitar e o caminho para lá chegar.
- Envio e recepção de mensagens de/para outros utilizadores do sistema e com centro de Turismo do Gerês. Isto ajuda grupos de visitantes a manterem-se em contacto.
- Acesso a serviços oferecidos por sistemas externos, como por exemplo enciclopédias electrónicas. Para isto você só tem que supor que o *FunParkGuide* tem um *browser* para acesso à *internet*.

O *FunParkGuide* acede a um repositório central com os dados relevantes do Gerês que permite o suporte às facilidades requeridas, como por exemplo tipos de caminhadas classificadas segundo a sua duração e grau de dificuldade, fauna, vegetação, pontos de repouso com acesso a água potável, restaurantes. A manutenção desta informação é da responsabilidade dos centros de apoio ao Turismo em Portugal (externos ao seu sistema).

- a) Considere as seguintes qualidades: **facilidade de modificação, desempenho, mobilidade, disponibilidade, facilidade de compreensão, compatibilidade, facilidade de utilização** (*user friendly*), **segurança** (*safety*), e **resistência a ataques** (*security*). Escolha as 3 que lhe pareçam mais importantes no contexto do *FunParkGuide*, e justifique a sua escolha.
- b) Dos estilos arquitecturais invocação implícita, sistema baseado em regras, *object-oriented*, *blackboard*, e cliente-servidor use uma combinação de **pelo menos 2** para implementar este sistema. **Construa o diagrama de componentes** usando uma abordagem *top-down*. Nota: Se sentir necessidade de criar símbolos especiais para realizar este exercício, não se esqueça de os explicitar, criando uma legenda apropriada para o seu diagrama.
- c) Descreva **sucintamente** o objectivo de cada um dos elementos principais, mais abstractos, da sua arquitectura.
- d) Discuta a forma como o seu desenho arquitectural suporta cada um dos três NFRs escolhidos.

Parte II

Assinale com **X** cada afirmação que considerar verdadeira. Cada questão terá 4 afirmações, das quais 0, 1, 2, 3, ou 4 podem ser verdadeiras. Cada resposta correcta vale um ponto, cada resposta incorrecta vale -1 ponto (por isso, evite responder quando não tiver a certeza da resposta). A pontuação mínima deste grupo é de 0%, ou seja, se por absurdo falhar mais respostas do que as que acertar, a penalização não passa para os restantes grupos do teste.

1. Numa arquitetura de software

- Os componentes representam a computação
- Os conectores representam a interacção entre componentes
- As interfaces ligam componentes a outros componentes
- As interfaces ligam componentes a conectores

2. A arquitetura

- É uma fase no processo de desenvolvimento
- Deve satisfazer os requisitos de todos os *stakeholders*
- Satisfaz os atributos de qualidade do sistema, por isso impondo neste restrições
- Engloba o chamado “desenho arquitectural”, que incide sobretudo nos principais algoritmos do sistema a desenvolver

3. Há a chamada arquitetura *prescritiva* e a arquitetura *descritiva*

- A arquitetura prescritiva é a arquitetura implementada
- A arquitetura descritiva é a arquitetura planeada
- Idealmente, a evolução da arquitetura deve começar pela evolução da arquitetura prescritiva
- Um desvio arquitectural é a introdução de decisões importantes de desenho que não estão previstas nas decisões de desenho da arquitetura prescritiva

4. Entre os atributos de qualidade observáveis em tempo de execução

- A funcionalidade do sistema depende directamente da sua estrutura
- Os vários atributos de qualidade são, em geral, completamente independentes entre si
- A segurança (*safety*) diz respeito à capacidade de um sistema resistir a tentativas não autorizadas de utilização, ou modificação de comportamento, enquanto mantém a capacidade de providenciar um serviço a utilizadores legítimos
- A portabilidade, capacidade de um sistema correr em diferentes ambientes computacionais é um atributo apenas observável em tempo de execução

5. No NFR framework

- Os “*NFR softgoals*” são requisitos funcionais de alto nível que devem ser satisfeitos
- As operacionalizações de *softgoals* são possíveis soluções, ou alternativas de desenho, que ajudam a satisfazer os “*NFR softgoals*”
- Os “*claim softgoals*” ajudam a explicar o contexto para um *softgoal*, ou *link* de interdependência entre *softgoals*
- Os *softgoals* e as suas interdependências explícitas e implícitas são, em geral, representados por uma árvore

6. Sobre as vistas de uma arquitectura

- A vista lógica descreve os mecanismos de concorrência e sincronização de uma arquitectura
- Um dos focos da vista de implementação é a gestão de configurações
- Cada vista do modelo de vistas “4+1” de Kruchten corresponde a um ou mais diagramas de UML e cada diagrama UML corresponde a apenas uma das vistas do modelo “4+1”
- A escolha das vistas a documentar deve depender dos *stakeholders* a quem se destina a documentação da arquitectura de software

7. No estilo de invocação implícita

- Em vez de chamar directamente um procedimento, um componente pode anunciar, ou difundir, um ou mais eventos
- Os componentes podem registar o interesse num evento, associando um procedimento a esse evento
- Quando o evento é anunciado, o sistema encarrega-se de invocar todos os procedimentos que foram registados nesse evento
- O anúncio por um componente de um evento pode causar a invocação implícita de procedimentos de outros componentes

8. No estilo blackboard

- Há suporte a transformações concorrentes feitas sobre dados partilhados
- Os componentes funcionam como unidades de processamento que são, simultaneamente, fontes de informação
- O estilo de interacção é tipicamente síncrono
- A consistência da informação é relativamente simples de manter

9. Quando temos de escolher uma abordagem de modelação da arquitectura

- Devemo-nos preocupar com o que uma determinada linguagem não nos permite modelar
- A escolha deve depender dos estilos arquitecturais que pretendemos usar
- Os requisitos não-funcionais podem normalmente ser deixados para segundo plano, nesta escolha
- O suporte à verificação de consistência entre vistas não é importante, em sistemas de grande dimensão, porque essa verificação é tão complexa que deve ser avaliada por peritos

10. Na avaliação de arquitecturas

- A consistência é uma propriedade externa do modelo arquitectural
- A consistência de nomes é, normalmente, fácil de verificar a um nível arquitectural
- A consistência de interfaces garante a respectiva consistência de comportamentos
- A compatibilidade de uma arquitectura é uma propriedade externa do modelo arquitectural que garante a adesão a restrições tais como as impostas por um determinado estilo arquitectural