

Programação de Sistemas Distribuídos

Trabalho Prático

MyPhotoAlbum.com

Paulo Gandra Sousa

Alexandre Bragança

{pag,atb}@isep.ipp.pt

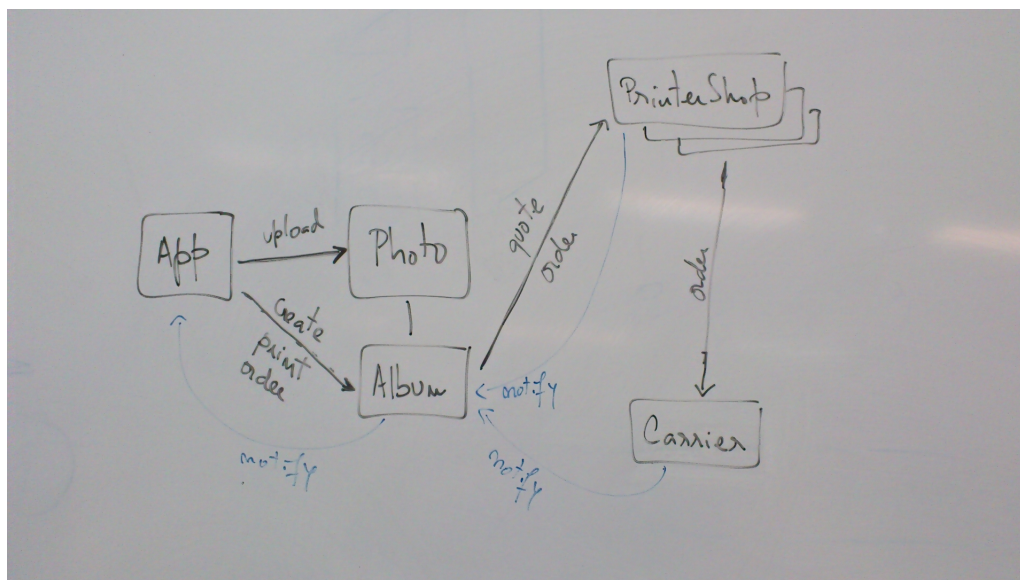
2014/2015

Objectivo

O trabalho consiste na conceção e desenvolvimento de um sistema distribuído que implementa parcialmente o negócio do site fictício MyPhotoAlbum.com. Pretende-se simular a cadeia de entidades que participam no negócio de produção e venda interativa de álbuns de fotografias através de um site na Internet.

O ênfase principal do trabalho será na especificação das interfaces de cada componente e na respetiva implementação do protocolo de interação, não sendo dado realce à lógica de negócio inerente nem a aspetos técnicos não relacionados com sistemas distribuídos, ex., base de dados e *user interface*.

Cenário Principal



O trabalho consiste na conceção e desenvolvimento de um sistema distribuído que implementa parcialmente o negócio do site fictício MyPhotoAlbum. Pretende-se simular a cadeia de entidades que par-

ticipam no negócio de produção e venda interativa de álbuns de fotografias através de um site na Internet.

O cenário comum de interação com o site MyPhotoAlbum.com relativamente à criação de álbuns é o seguinte:

1. O utilizador entra no site e, de alguma forma, identifica-se perante o site. Se o utilizador não estiver registado deve ser dada a possibilidade de o fazer.
2. O utilizador pode criar álbuns para catalogar as fotografias. Cada álbum é caracterizado por um nome, uma pequena descrição e possivelmente por um período de tempo, ex., "Férias 2015" de 2015-07-23 a 2015-08-12.
3. O utilizador pode fazer o upload de fotografias para os álbuns. Cada fotografia é acompanhada por uma pequena descrição e por uma data.
4. O utilizador pode *construir* "álbuns para impressão" em formato físico e digital (PDF). Os álbuns para impressão são construídos escolhendo fotos existentes nos álbuns "normais", indicando a ordem de impressão, o título, o tema e uma mensagem de abertura. **A geração do PDF pode ser um processo demorado.**
5. Estes álbuns para impressão devem ficar disponíveis para o Utilizador voltar a imprimi-los no futuro.

O cenário comum de interação com o site MyPhotoAlbum.com relativamente à encomenda de álbuns é o seguinte:

1. O utilizador entra no site e, de alguma forma, identifica-se perante o site. Se o utilizador não estiver registado deve ser dada a possibilidade de o fazer.
2. O utilizador seleciona a opção de encomenda de álbuns.
3. O utilizador constrói o álbum que deseja imprimir selecionando fotografias de álbuns existentes.
4. O utilizador indica qual o tema do álbum, ex., Vida
5. O sistema deve apresentar o preço a pagar por cada impressão de álbum.
6. O utilizador seleciona a opção de confirmar encomenda.
7. O utilizador indica a morada de entrega da encomenda.
8. O sistema apresenta o preço total da encomenda (incluindo o preço da impressão e do transporte).
9. O utilizador confirma a encomenda (a nossa simulação não inclui questões relacionadas com o pagamento).
10. O utilizador, para além de receber o álbum na morada indicada, irá receber (logo que possível) um link para descarregar as versões PDF dos seus álbuns encomendados.
11. O utilizador pode consultar regularmente o estado da sua encomenda. O utilizador (aplicação cliente/web site) deve ser sempre notificado de alterações no estado da encomenda. Esta situação pode também servir para indicar que as versões PDF dos álbuns já estão disponíveis para download.

Descrição Adicional

As entidades envolvidas, e suas principais responsabilidades, são:

- **App.** Esta aplicação simula a interação do utilizador com o site MyPhotoAlbum.com (neste documento por vezes designados também por site). Esta interage com os utilizadores "finais" (consumidores) e com o backoffice. Esta aplicação pode ser uma aplicação node (tal como desenvolveu nas aulas PL) ou uma aplicação web em javascript.

- **Backoffice.** Aplicação (ou aplicações) node que implementa os serviços da entidade que gere o MyPhotoAlbum.com. Este interage com a App, os PrinterShops e os Carrier.
- **PrinterShop.** Devem ser simulados pelo menos 3 PrinterShops: A, B e C. O Backoffice usa os PrinterShops para “imprimir” os álbuns que os seus clientes encomendam. Os PrinterShops são entidades independentes do MyPhotoAlbum.com e, por isso, **cada um tem a sua interface** de comunicação diferente. Sempre que o Backoffice necessita de satisfazer uma encomenda de um cliente ele inicia um processo de seleção do PrinterShop. Esse processo inclui o pedido de orçamento a todos (estes orçamentos são importantes para o backoffice identificar a melhor opção em termos de custo interno, não tem impacto no preço final ao cliente que já foi estabelecido). Com base em todos os orçamentos o Backoffice seleciona a melhor opção e coloca a encomenda. A lógica de seleção está descrita mais a frente neste documento. As PrinterShops necessitam de saber a morada de entrega da encomenda assim como uma "forma" de notificar o backoffice.
- **Carrier.** As PrinterShops trabalham com um Carrier específico. O Carrier é que vai efectuar o transporte dos álbuns para a morada do cliente. Pode assumir que existe um acordo entre cada PrinterShop e o Carrier e que o mesmo Carrier pode ter acordo com mais que um PrinterShop (não é necessário lidar com esse aspecto nesta simulação). O Carrier deve notificar o MyPhotoAlbum.com de mudanças no estado do transporte da encomenda (pelo menos "em transporte" e "entregue")

Requisitos Adicionais:

- Devem ser utilizados serviços REST para a comunicação (excepto se utilizar APIs terceiras que obriguem a outro protocolo).
- Todas as aplicações devem ser desenvolvidas em node (excepto a App que pode simular com um cliente Web em javascript)
- Deve incluir um dos seguintes serviços externos:
 - O Carrier deve usar um serviço real de tempo (meteorologia) para enviar informação de tempo nas suas notificações sobre o transporte
 - O Backoffice deve usar um serviço de citações para incluir uma frase de citação de um dado tema em cada álbum encomendado.
- Note-se que o sistema deve usar a seguinte lógica para satisfazer cada pedido de cliente: o Backoffice deve pedir orçamentos às PrinterShops e escolher com base no preço enviado por cada PrinterShop e pela pegada de carbono do provável trajecto de transporte ($\text{km} * 0.26\text{€} / 1000 \text{ km}$)
- Um site de geolocalização real na Internet que deve ser usado pelo Backoffice para determinar a distância entre a morada do cliente e a do PrinteShop (tal como descrito no ponto anterior). Pode, por exemplo, usar os serviços Geocoding da Google (<https://developers.google.com/maps/documentation/geocoding/>)
- Os preços de impressão dos álbuns pode ser simulados, por exemplo, multiplicando o número de fotografias por um valor fixo. Acresce um valor, também simulado, para o transporte.

Valorizações

- Incluir na sua solução um "pequeno" serviço em .Net ou Java que deve verificar a "força" das passwords quando um utilizador se regista. O algoritmo de cálculo a usar pode ser muito simples. Deve seguir o documento no moodle que explica como desenvolver serviços REST em Java e .Net.
- Utilização do 0MQ (<http://zeromq.org> - ver capítulo 4 do "Node.js the Right Way") para implementar um serviço que possa escalar em função da carga. Por exemplo, no processo de upload de fotografias ou criação dos PDF dos álbuns.

Tópicos Omissos

Em casos não explícitos neste documento, deverá efetuar as suas opções de acordo com as suas necessidades e **justificá-las**. É fundamental que a solução seja devidamente descrita no relatório.

Regras de Funcionamento

O trabalho deverá ser efetuado em grupos de três elementos. A data limite de entrega do trabalho é **12/01/2015 até às 10:00** via moodle, sendo as apresentações efetuadas de preferência na mesma semana em data a anunciar no moodle.

O Trabalho terá 2 datas de entrega

Primeira Entrega

Na primeira entrega, até **12/12/2015 às 10h00**, o grupo enviará via moodle o(s) documento(s) resultantes da análise do problema.

Devem elaborar um powerpoint/PDF apenas contendo:

- esquema do sistema (elementos existentes e suas interligações)
- casos de uso (em formato breve)
- diagrama de sequência do protocolo de interação para cada caso de uso
- API REST. Para cada recurso:
 - URI
 - verbos suportados
 - formato de representação standard
 - request body
 - response body + status code
- padrões utilizados: quais e onde (se aplicável)

Segunda Entrega

Na segunda entrega, até **12/01/2015**, o grupo enviará, via moodle, o código fonte da sua implementação e possíveis atualizações dos documentos de análise produzidos na primeira entrega.

Nos documentos que o grupo enviar deve ter em consideração os seguintes pontos:

- A arquitetura do sistema distribuído implementado
- Justificação da divisão efetuada em termos de processos/aplicações
- Identificação das interfaces e contratos de cada componente do sistema
- Avaliação do sistema segundo as características dos sistemas distribuídos (modelo arquitetural, transparência, segurança, tolerância a falhas, tipos de comunicação, mecanismos de comunicação, protocolos de interação, escalabilidade, heterogeneidade)
- Justificação das decisões tomadas tendo em conta as alternativas existentes

- Identificação das tecnologias utilizadas (designação, versão, vendor)
 - Sistema operativo
 - Application server
 - Web server
 - Linguagem de programação
 - ...
- Código fonte produzido. A implementação deve focar na estruturação do sistema e não na funcionalidade de negócio propriamente dita (ex., é aceitável que uma operação de negócio para consulta de catalogo de produtos retorne uma lista fixa de produtos hardcoded no programa)

Avaliação

A avaliação do trabalho consistirá:

- **1a entrega: 35%**
- **2a entrega: 65%**

A avaliação levará em conta os seguintes critérios:

- Qualidade da solução proposta (conteúdo do relatório/análise)
- Produto de software
- Redação e estrutura do relatório
- Apresentação do trabalho

Aconselha-se o uso de ferramentas de controlo de versões, ex., Git, para o desenvolvimento do trabalho pela equipa.

Bom trabalho!