**INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA**

Engenharia Informática e de Computadores

**Sistemas Distribuidos**

**3º Trabalho Prático**

**Web Services**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Membros do Grupo** | | | | | |
| **31768** | **Ricardo Romão** | **31923** | **Nuno Sousa** | **26657** | **Ricardo Neto** |

**Análise do Problema**

**Arquitectura**

De acordo com o enunciado, pretende-se o desenvolvimento de um sistema que, no final, permita a consulta de informação relativa a filmes em exibição em vários cinemas.

A fim de o permitir, deverão ser disponibilizados vários *webservices* (um por cinema) que disponibilizem toda a informação relativa ao próprio cinema e que, adicionalmente, ofereçam funcionalidades de reserva de filmes.

Para que todos os *webservices* disponibilizados pelos cinemas possam ser encontrados, deverá ser desenvolvido um serviço de directório, designado por Broker, que disponibilize essa informação, bem como possibilite o registo e cancelamento de registo de cinemas.

**Tipos de Repositórios**

No desenvolvimento da solução optou-se por distinguir entre dois tipos de locais para armazenamento de dados: repositórios de leitura e repositórios de leitura e escrita.

No que respeita aos repositórios de leitura, optou-se por utilizar um tipo Singleton que ofecere-se essa funcionalidade. Este tipo será responsável por garantir que as operações de consulta são feitas de forma *thread safe*.

Os repositórios de leitura e escrita foram implementados com recurso ao .NET Remoting, oferecendo assim o acesso a dados pela via de um objecto remoto com activado no modo Singleton.

**Broker *Webservice***

**Descrição**

O Broker é o serviço cuja localização deverá ser conhecida por todos os Clientes que pretendam obter informações acerca da localização de serviços disponibilizados por cinemas, bem como, os nomes destes.

Este deverá ter acesso a um repositório de leitura e escrita que guarde os dados relativos a cada cinema.

**Arquitectura**

A arquitectura escolhida para o processo relativo ao Broker foi a seguinte:

**Repositório**

No sentido de centralizar o acesso ao directório de cinemas, optou-se por implementar a entidade *RegistryServer*, sendo esta activada remotamente no modo Singleton.

Inicialmente considerou-se a criação de um tipo Singleton em que, por cada instância do *webservice* se garantiria uma única instância do mesmo.

Assim, no mesmo processo do *webserver* existiria apenas uma instância para acesso ao repositório, contudo, esta solução apresentava desvantagens no caso de se querer por exemplo fazer um *scaling* horizontal ao sistema. Neste caso iriam haver várias instâncias Singleton em cada processador o que iria provocar anomalias no resultado das leituras e escritas.

Os dados escolhidos para cada cinema foram o seu nome e o endereço de acesso ao seu *websevice* e, a forma escolhida para a persistência desses dados, foi em formato XML, de acordo com o seguinte exemplo:

<cinemas>

<cinema>

<name>BBC - BlockBusterCalifornia</name>

<url>http://localhost:3354/WSCinema.asmx</url>

</cinema>

</cinemas>

**Interface IWSBroker**

A fim de especificar as operações disponibilizadas pelo *webservice* do Broker, definiu-se a interface IWSBroker com a seguinte estrutura:

|  |
| --- |
| Interface IWSBroker  Interface responsável por definir as operações expostas pelo webservice de Broker. |
| GetCinemas() : List<CinemaRegistry>  Método que devolve uma lista de objectos representativos de cinemas e respectivos *endpoints*. |
| RegisterCinema(string name, string url)  Método que permite o registo de um cinema, recebendo o seu nome e endereço onde o *webservice* é disponibilizado. |
| UnregisterCinema(string name)  Método que permite apagar o registo de um cinema |

**Cinema *Webservice***