

## Laboratório 1 (versão 3.0): Criação de serviços usando Parlay/OSA

### Notas prévias à realização do Laboratório 1

A realização deste laboratório requer uma preparação prévia com uma duração aproximada de 2 horas (1 hora para leitura da bibliografia mais 1 hora para a ambientação com a plataforma).

#### Referências bibliográficas para o laboratório

- [MoKl03] A. Moerdijk, L .Klostermann; "Opening the networks with Parlay/OSA: standards and aspects behind the APIs"; IEEE Network 17, 2003
- [Stre04] R. M. Stretch; "The Parlay API – Allowing Third Party Application Providers Safe and Secure Access to Network Capabilities"; BT Technology Journal, November 2004

#### Ambientação com a plataforma a usar no laboratório

- A máquina virtual VMWare com a plataforma NRG da Ericsson foi distribuída nas aulas
- Arrancar a máquina virtual e utilizar a plataforma realizando as seguintes experiências (estas experiências encontram-se descritas em:  
[http://iscte.pt/~rhcl/material/IGRS/parlay\\_osa\\_jain/parlay.html](http://iscte.pt/~rhcl/material/IGRS/parlay_osa_jain/parlay.html)):
  1. Efetuar chamadas, e enviar SMS, entre telemóveis em diferentes cenários.  
Exemplos nos seguintes vídeos:
    - [http://iscte.pt/~rhcl/material/IGRS/parlay\\_osa\\_jain/Phones.htm](http://iscte.pt/~rhcl/material/IGRS/parlay_osa_jain/Phones.htm)
    - [http://iscte.pt/~rhcl/material/IGRS/parlay\\_osa\\_jain/Busy.htm](http://iscte.pt/~rhcl/material/IGRS/parlay_osa_jain/Busy.htm)
    - [http://iscte.pt/~rhcl/material/IGRS/parlay\\_osa\\_jain/SwitchOff.htm](http://iscte.pt/~rhcl/material/IGRS/parlay_osa_jain/SwitchOff.htm)
    - [http://iscte.pt/~rhcl/material/IGRS/parlay\\_osa\\_jain/SMS.htm](http://iscte.pt/~rhcl/material/IGRS/parlay_osa_jain/SMS.htm)
  2. Utilizar a funcionalidade do serviço *NulCallApp*:  
Exemplos nos seguinte vídeo:
    - [http://iscte.pt/~rhcl/material/IGRS/parlay\\_osa\\_jain/NullApp.htm](http://iscte.pt/~rhcl/material/IGRS/parlay_osa_jain/NullApp.htm)
  3. Utilizar a funcionalidade do serviço *CallMonitor*:  
Exemplos no seguinte vídeo:
    - [http://iscte.pt/~rhcl/material/IGRS/parlay\\_osa\\_jain/CallMonitor.htm](http://iscte.pt/~rhcl/material/IGRS/parlay_osa_jain/CallMonitor.htm)
  4. Utilizar a funcionalidade do serviço *CallRedirect*:  
Exemplos no seguinte vídeo:
    - [http://iscte.pt/~rhcl/material/IGRS/parlay\\_osa\\_jain/CallRedirect.htm](http://iscte.pt/~rhcl/material/IGRS/parlay_osa_jain/CallRedirect.htm)
  5. Visualizar o processo de compilação de uma aplicação.  
Exemplos no seguinte vídeo:
    - [http://iscte.pt/~rhcl/material/IGRS/parlay\\_osa\\_jain/structure.htm](http://iscte.pt/~rhcl/material/IGRS/parlay_osa_jain/structure.htm)

<b>GRUPO:</b>	
---------------	--

## Laboratório 1 (versão 3.0): Criação de serviços usando Parlay/OSA.

*Nota: Os pontos do enunciado inseridos numa caixa cinzenta são parte da preparação prévia ao Laboratório.*

### Introdução e objectivos:

O objectivo deste laboratório é estudar a arquitectura Parlay/OSA para o desenvolvimento de serviços de telecomunicações usando a sua API normalizada, nomeadamente através da análise de alguns exemplos. Os aspectos a estudar são:

- Estudo de algumas classes (API) definidas pelas recomendações Parlay/OSA.
- Estudo da interacção entre a plataforma (*Framework* e *Service Capabilities Server*) e as aplicações.
- Estudo da estrutura de classes usualmente envolvida numa aplicação.
- Estudo da estrutura da aplicação e sua relação com os serviços/aplicações de telecomunicações.
- Estudo dos mecanismos de interacção entre a plataforma (“rede”) e as aplicações
  - Pontos de interacção (*events*) entre a rede (plataforma) e as aplicações
  - Lógica das aplicações, programação assíncrona (*callbacks*, classes **AppX**).

### 0. Instalação e teste da plataforma

- O seu nome de utilizador é definido da forma IGRS11-t1-0X (em que X é o grupo) a palavra de passe é idêntica para todos os grupos.
- Na directória C:\Laboratorios\Maquinas Virtuais encontra-se a pasta IGRS-2010-Lab1 que contém a máquina virtual a usar neste laboratório e que deverá copiar para a sua pasta MyDocuments
- No ficheiro de configuração da máquina virtual altere o último dígito do endereço físico da interface Ethernet virtual para X. Arranque a máquina virtual.
- Nessa máquina virtual já se encontra instalado o NRG da Ericsson bem como as aplicações exemplo. Alguns items para serem explorados.
  - **Atalho para ‘NRG Simulator’:** Crie dois terminais (601 e 8001) e: estabeleça chamadas entre eles, troca de SMS em diferentes situações (e.g., para um terminal desligado); posição no mapa. (Não feche o simulador NRG.)
  - **Atalho para ‘NRG Applications’:** Permite lançar e controlar aplicações através de interfaces gráficas. Lança a interface gráfica de uma delas e arranque-a (*deploy*).
  - **Atalho para pasta ‘NRG folder’:** Pasta onde se encontra o código fonte e *bytecode* das aplicações bem como todas as ferramentas do NRG simulator.
  - **Atalho para Parlay Core JavaDOC:** JavaDoc para os *Packages* e respectivas classes que estão implementadas no NRG (comparar com acetatos).
  - **Siga:** Start → Todos os programas → Ericsson → Network...SDK → R5A02 Consulte a documentação presente no documento .

# 1. Estudo de algumas aplicações exemplo

## 1.1. Aplicação NullCallApp (21)

A utilização de uma funcionalidade (i.e., serviço) da rede por uma qualquer aplicação obriga sempre a interagir com a *Framework Parlay*:

- Autenticar a aplicação com a *Framework* (necessário uma única vez.)
- Descobrir o(s) SCS que fornecem esse serviço.
- Negociar e aceitar os SLA relativos à utilização desse serviço

### 1.1.1. Utilização da aplicação

*Esta aplicação não oferece qualquer funcionalidade aos utilizadores finais, implementando apenas as fases de autenticação descoberta e negociação de acesso aos serviços.*

O “resultado” desta aplicação é obter uma referência para um recurso computacional (neste caso um objecto Java) através do qual a aplicação pode usar os recursos da “rede” (e.g., iniciar uma chamada).

*Esta aplicação encontra-se na diretoria*

C:\Program Files\Ericsson\Network Resource Gateway SDK\R5A02\examples\nullcallapp

Execute a aplicação correndo a mesma de uma janela de comandos Windows, através dos seguintes passos ([http://iscte.pt/~rhcl/material/IGRS/parlay\\_osa\\_jain/NullApp.htm](http://iscte.pt/~rhcl/material/IGRS/parlay_osa_jain/NullApp.htm)):

1. Arranque o simulador NRG da Ericsson
2. Obtenha uma janela de comandos Windows
3. Na janela de comandos Windows mude a directória de trabalho para:  
C:\Program Files\Ericsson\Network Resource Gateway SDK\R5A02\
4. Execute o comando  
run\_examples.bat
5. Escolha a aplicação Null Call Application: 21

### 1.1.1. Estudo e análise da aplicação

a. Comente as mensagens relevantes que surgem na consola à luz das interacções entre as aplicações e a *Framework* (i.e., a que passo nessas interacções corresponde).

Indique ainda qual o SCS que é requerido

**b.** Use um editor de texto para abrir os ficheiro Main.java e Feature.java com a lógica da aplicação.

Identifique quais os **objectos e respectivas classes** que nesta aplicação (não necessariamente pertencentes à implementação da aplicação) são as implementações computacionais dos seguintes elementos numa arquitectura Parlay/OSA.

Assinale os que pertencem à implementação da aplicação e os que pertencem à plataforma.

**Lógica do serviço (aplicação) :**

**Framework:**

**(Service Capability Server) :**

## 1.2. Aplicação CallMonitor (22)

A utilização de um serviço numa aplicação corresponde sempre à instanciação de dois objectos pares:

- um na plataforma representando um SCS (ou neste caso o simulador NRG) através do qual se invocam as diferentes funcionalidades do serviço em causa (ex., adição de uma *Leg a uma chamada*);
- outro na própria aplicação (cujo código deve ser escrito pela(o) criador(a) da aplicação) e que serve para a aplicação ser contactada pela plataforma onde corre o serviço (ex., anunciar a conclusão de uma chamada). (Este objecto é usado pela plataforma como o objecto para efectuar o *callback* associado a esse serviço).

### 1.2.1. Utilização da aplicação

*Esta aplicação monitoriza o estado de chamadas efectuadas para o número 601 ou para números que correspondem ao padrão “800\*”. A monitorização é efectuada imprimindo na consola o conjunto de eventos que são comunicados pela plataforma (o simulador NRG) à aplicação.*

*Esta aplicação encontra-se na diretoria*

*C:\Program Files\Ericsson\Network Resource Gateway SDK\R5A02\examples\callmonitor*

Execute a aplicação correndo a mesma usando o ícone “NRG Applications”, através dos seguintes passos ([http://iscte.pt/~rhcl/material/IGRS/parlay\\_osa\\_jain/CallMonitor.htm](http://iscte.pt/~rhcl/material/IGRS/parlay_osa_jain/CallMonitor.htm)):

1. Arranque o simulador NRG da Ericsson
2. Crie alguns terminais móveis (pelo menos um deles deverá ter o número 601)
3. Utilizando o ícone “NRG Applications” escolha a aplicação CallMonitor (22)
4. Efetue chamadas para o número 801 em diferentes situações (número livre, ocupado, ...) e observe a forma como os eventos são reportados na consola e/ou como são redirecionadas para o número 601

## 1.2.2. Análise de algumas classes que formam a (ou são usadas pela) aplicação - *deployment*

a. A lógica desta aplicação encontra-se codificada no ficheiro

C:\...\examples\callmonitor\Feature.java

Considerando apenas a fase de implantação do serviço (*deployment*) indique:

**O serviço (SCS) que é utilizado pela aplicação:**

**Qual o objecto e a classe/interface que permite usar esse SCS:**

**Consultando a documentação JavaDoc desta classe indique algumas das funcionalidades que o SCS oferece:**

b. Por forma a que os SCS possam interagir com a lógica da aplicação é necessário que esta possua objectos adequados para *callback*. (Estas classes são usualmente referidas como **Processors** sendo usada uma por SCS utilizado.)

Nesta aplicação é a classe MPCCProcessor. Edite o ficheiro **MPCCProcessor.java** e indique:

**Qual o objecto e a classe que do lado da aplicação permitem que esse SCS interaja (i.e., contacte) a aplicação:**

**A interface da qual esta classe é uma implementação:**

**Os métodos da interface que são implementados pela classe MPCCProcessor**

**A razão pela qual não é necessário que todos os métodos da interface sejam implementados pela classe MPCCProcessor**

### **1.2.3. Análise da interacção entre a aplicação e a rede - *provisioning***

a. De forma a fazer a provisão (*provisioning*) das aplicações é necessário recorrer à classe TpCallNotificationRequest de forma a definir quais as condições e o modo em que essa interacção é efectuada. Considere a chamada à funcionalidade SCS que é feita no final do método. Consultado o JavaDOC respeitante a esta classe indique:

**Qual a função do campo CallNotificationScope e como se relaciona com os serviços de valor acrescentados originados e terminados:**

**Qual a função do campo CallEventsRequested e como se relaciona com os pontos numa chamada (consulte o JavaDOC da classe respectiva.)**

b. Considerando a chamada à funcionalidade do SCS que permite o registo em notificações comente qual a classe e o valor para o campo appCallControlManager (ver JavaDoc)

c. Para cada um dos eventos que é usado no pedido de notificação indique: (Consultando o JavaDOC e as secções 7.6.2.22 a 7.6.2.35 do document 29198-4-460 presente em ...à R5A02 à Documentation à Standards).

<b>Evento (Objecto)</b>	<b>Event Type</b>	<b>Critério(s) Adicionais</b>	<b>Modo como é armado. (E quem fica com o controlo da chamada.)</b>	<b>"DP" equivalente r DFP da IN</b>
termReleaseRequest				

### 1.3. Aplicação CallRedirect (23)

A lógica de serviço de algumas aplicações requerem que a execução do fluxo normal de uma chamada (a chamada básica na terminologia da Rede Inteligente)

#### 1.3.1. Utilização da aplicação

*Esta aplicação, para um determinado número definido num ficheiro de configuração (801):*

- *avisa a ocorrência do estado ocupado,*
- *caso esse número se encontre fora da rede (i.e., desligado) a chamada é redirecionada para um outro número (também especificado no ficheiro de configuração - 601).*

*Esta aplicação encontra-se na diretoria*

C:\Program Files\Ericsson\Network Resource Gateway SDK\R5A02\examples\callmonitor

Execute a aplicação correndo a mesma usando o ícone “NRG Applications”, através dos seguintes passos ( [http://iscte.pt/~rhcl/material/IGRS/parlay\\_osa\\_jain/CallRedirect.htm](http://iscte.pt/~rhcl/material/IGRS/parlay_osa_jain/CallRedirect.htm) ):

1. *Arranque o simulador NRG da Ericsson*
2. *Crie alguns terminais móveis (pelo menos um deles deverá ter o número 601)*
3. *Utilizando o ícone “NRG Applications” escolha a aplicação CallMonitor (22)*
4. *Efetue chamadas para o número 801 em diferentes situações (número livre, ocupado, ...) e observe a forma como os eventos são reportados na consola e/ou como são redirecionadas para o número 601*

#### 1.3.1. Utilização da aplicação

Apresente o diagrama de classes (UML) das classes utilizadas nesta aplicação.

(faça um diagrama semelhante ao fornecido na documentação da Ericsson na Secção 6.1 de

<C:\... R5A02\simulator\doc\Ericsson...Kit.htm> ).

### **1.3.2. Descrição da lógica aplicação (usando o INCM)**

Descreva a lógica do serviço de acordo com as recomendações para o Plano Funcional Global (GFP) do modelo conceptual da IN (INCM)

### **1.3.3. Estudo e representação da evolução da chamada e aplicação**

Apresente o diagrama de *Connection View States* e o diagrama de sequência para as seguintes situações: (Nota: Os diagrama de sequência devem ser semelhantes aos apresentados nos exemplos fornecido pela Ericsson na Secção 6 de <C:\...\R5A02\simulator\doc\Ericsson...Kit.htm> ).

- a. É feita uma chamada com sucesso para o número 801 (a chamada é atendida nesse número).

**b.** É feita uma chamada para o número 801 quando esse número está envolvido numa chamada.

- c. É feita uma chamada com sucesso para o número 801 quando esse número está desligado da rede.

### 1.3.4. Interacção entre a aplicação e a plataforma (rede de telecomunicações)

a. Indique qual a função do método **route**

b. No método **route** encontra-se a seguinte interacção com o SCS, indique:

```
public void route( TpMultiPartyCallIdentifier aCall,
                   TpAddress aSource,
                   String aDestination)
{
    TpCallLegIdentifier newLeg = aCall.CallReference.createAndRouteCallLegReq
        (aCall.CallSessionID,
         new TpCallEventRequest[0],
         createAddress(aDestination), aSource,
         new TpCallAppInfo[0], theLegCallback); }
```

**Qual a função deste método:**

**Qual a função de cada um dos seus parâmetros (comente os valores que são passados):**

**Descreva sumariamente como poderia modificar a aplicação para que a mesma pudesse ser notificada quando a *leg* criada atingisse determinados pontos na chamada:**

c. Nas funções **startNotifications** e **route** classe MPCCProcessor informa a plataforma NRG (“SCS do operador”) da necessidade da aplicação interagir com a rede em alguns pontos da chamada.

Método (start.../route)	Event Type	Critério(s) Adicionais	Scope	Modo como é armado. (Controlo da chamada.)	“DP” equivalente na DFP da IN