



**TÉCNICO**  
LISBOA

# **Integração Empresarial**

2º semestre - 2016/2017

1ª Entrega - Service Trees

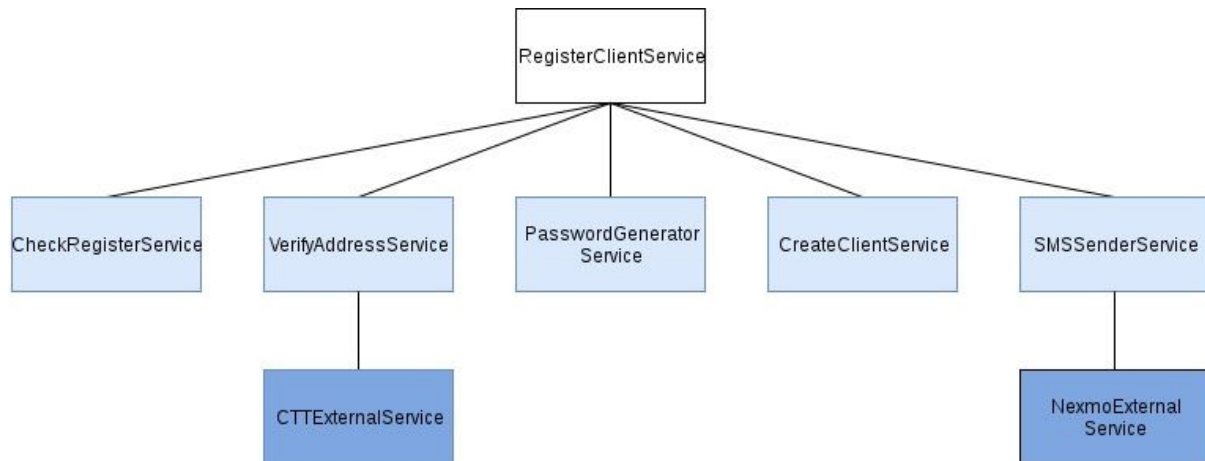
## **Grupo 9**

79023 - Carolina Fernandes

79055 - Pedro Ribeiro

Este relatório apresenta a arquitetura orientada a serviços (SOA) seguida para o desenvolvimento da *FormoPlace* (*Formo* é o latim para *moda*).

## 1. Register Client Service Tree

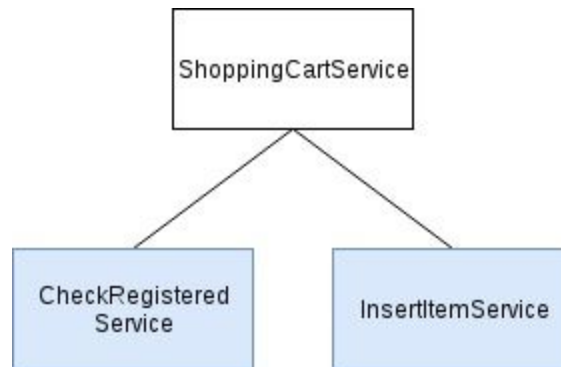


Para a implementação deste serviço, foram usados os seguintes serviços:

- **RegisterClientService:**
  - (String Email, String Address, String PhoneNumber) -> (String Response)
  - Tenta criar um utilizador, caso este não exista, e introduzi-lo na base de dados *client*. Para tal, atribui-lhe um user id e gera uma password aleatória, a qual é enviada ao cliente através de um SMS. Retorna se teve ou não sucesso.
- **CheckRegisterService:**
  - (String Email) -> (String Response);
  - Usa um adaptador para aceder à base de dados de modo a validar se um cliente está registado na base de dados *client*.
- **VerifyAddressService:**
  - (String Address) -> (Boolean IsOk);
  - Utiliza o serviço externo dos CTT para validar se a morada introduzida é válida.
- **PasswordGeneratorService:**
  - ( ) -> (String Password);
  - Gera uma password alfanumérica aleatória com um tamanho específico (8 caracteres).
- **CreateClientService:**
  - (String Email, String Password, String Address, String PhoneNumber) -> ( );
  - Usa um adaptador para aceder à base de dados de modo a criar um novo cliente na base de dados *client*.
- **SMSSenderService:**
  - (String To, String PhoneNumber) -> (boolean isOk);

- Utiliza o serviço externo do Nexmo para mandar uma SMS de confirmação para o número indicado no registo, com o id do novo cliente e a password gerada.

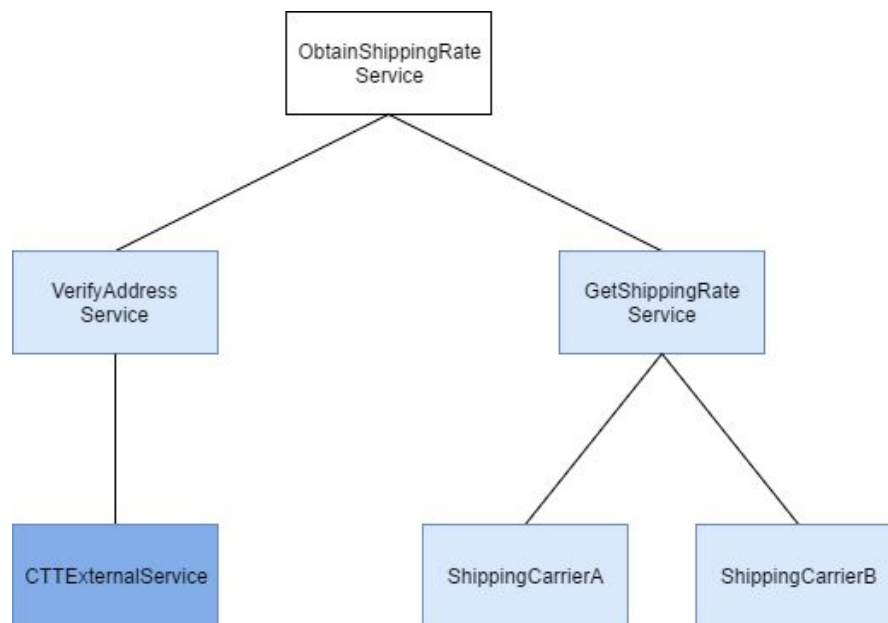
## 2. Add merchandise Service Tree



Para a implementação deste serviço, foram usados os seguintes serviços:

- **ShoppingCartService:**
  - (int Userid, String password, int Itemid, int Quantity) -> (String Response);
  - Tenta inserir um item no carrinho de compras de um utilizador registado. Retorna se teve sucesso ou não.
- **CheckRegisteredService:**
  - (int Userid, String password) -> (boolean isRegistered);
  - Usa um adaptador para aceder à base de dados de modo a validar se um cliente está registado na base de dados *client*.
- **InsertItemService:**
  - (int Userid, String password, int Itemid, int Quantity) -> ( );
  - Usa um adaptador para aceder à base de dados de modo a inserir uma nova entrada na tabela *cart*.

### 3. Request shipping rate Service Tree



Para a implementação deste serviço, foram usados os seguintes serviços:

- **ObtainShippingRate:**
  - (String BoutiqueAddress, String ClientAddress, int ShippingTimes) -> (String Response);
  - Verifica se a morada de envio é válida e, caso seja, calcula entre dois providers A e B qual o que oferece um preço mais baixo. Retorna se teve sucesso ou não.
- **VerifyAddressService:**
  - (String Address) -> (Boolean IsOk);
  - Utiliza o serviço externo dos CTT para validar se a morada introduzida é válida.
- **GetShippingRateService:**
  - (String boutiqueAddress, String clientAddress, int shippingTimes) -> ( double shippingName, String carrierName);
  - Compara as ofertas das duas shipping carriers (A e B) e devolve a melhor taxa e o respectivo Shipping Carrier. Invoca os serviços de ShippingCarrierA e ShippingCarrierB, que calculam um número random entre 2 algarismos correspondente à taxa.