Projecto de Engenharia de Software
Relatório Entrega Final

Repositório A-02-12-14

André Bispo	nº66941
Ricardo Leitão	nº69632
Diogo Pinto	nº69905
João Silva	nº70338
Pedro Soldado	nº70184
João Amorim	nº69310

Diagrama de Classes

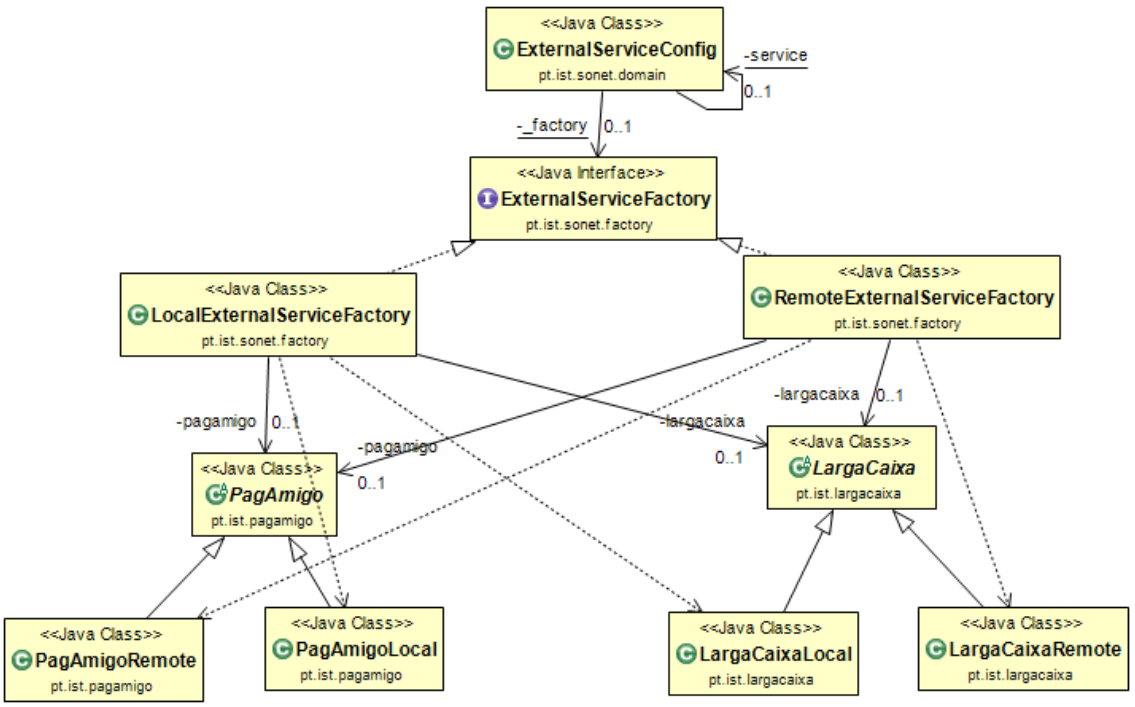

Figura 1 – Diagrama de Classes representativo da integração ES-SD

A classe ExternalServiceConfig corresponde à aplicação do padrão de desenho Singleton, guardando uma ExternalServiceFactory responsável por sempre que necessário, alternar entre os modos de execução local e remoto. A classe LocalExternalServiceFactory corresponde a uma fábrica que permite criar os serviços PagAmigo e LargaCaixa em modo local enquanto que a fábrica RemoteExternalServiceFactory permite criar os serviços em modo remoto.

Os serviços em modo remoto têm apenas as chamadas aos métodos criados no âmbito do projecto de Sistemas Distribuídos.

Refactorização de Código

Para permitir a ligação entre os projectos de Engenharia de Software e Sistemas Distribuidos foi necessário proceder a uma refactorização de código, de modo a ser possível chamar métodos remotos sem ser necessário recompilar toda a SoNet.

Para tal, foi necessário compilar todas as classes dos serviços PagAmigo e LargaCaixa e criar um .jar de modo a termos acesso a uma biblioteca compilada de todos os métodos necessários na SoNet.

As classes PagAmigoRemote e LargaCaixaRemote são as classes que realizam a tradução das chamadas aos métodos dos serviços PagAmigo e LargaCaixa da SoNet nos Webservices respectivos. Assim, cada método nestas classes utiliza o UDDI para encontrar o serviço e chama o método através desse serviço.

Testes realizados

Testes de serviços

Ao longo da realização do projecto, todos os serviços implementados foram testados depois de serem realizados. Coube a cada programador, aquando responsável pela realização do serviço, testá-lo conveninentemente, tendo em conta o caso de execução normal, ou seja, sem erros, e os casos de excepção (utilizador que invoca serviço que não existe, etc…)

Além destes testes realizados, foram ainda implementados testes Junit aos serviços de adicionar amigo, criar nova entidade organizacional, obter publicações de agente e votar publicação.

No teste Junit de adicionar amigos, são cobertos os casos em que não existe o agente que tentamos adicionar, bem como o caso em que o limite de amigos foi ultrapassado, sendo as excepções lançadas correctamente tratadas.

No teste de criação de nova entidade organizacional, o principal teste centra-se no facto de podermos estar a tentar registar uma organização com um *username* que já existe.

Nos testes de obter publicações de agentes, são testados os casos de excepção de a publicação que se pretende encontrar não existir bem como o agente a quem pertence a publicação não existir.

Por último, no teste de votar numa publicação, são testados os casos de excepção em que o utilizador tenta votar numa publicação que não existe, em que tenta votar numa publicação sua, em que tenta votar duas vezes na mesma publicação e ainda quando a publicação atinge o limite de votos por se tornar numa publicação polémica.

Testes de dominio

Os testes de domínio, baseados na criação de todos os elementos da SoNet, foram testados sobretudo na primeira e segunda entregas, que realizaram as alterações de domínio mais significativas. Estes testes foram muitas vezes executados com acessos às bases de dados de modo a confirmar a realização de criações de publicações, agentes ou comentários.

A partir da 2ª entrega, não foi necessário realizar mais testes a este nível, visto todas as interacções com os domínios passarem a ser feitas através de serviços, previamente testados.

Testes de apresentação

Os testes na camada de apresentação foram realizados em tempo de execução, isto é, à medida que as interfaces e as interações eram criadas, visto se tratar apenas de uma componente visual, acente em serviços previamente criados e testados.

Esta é, inclusivé, a maior vantagem de uma arquitectura por camadas, permitindo um desenvolvimento incremental e facilitando o processo de testes. Esta arquitectura providencia-nos uma abstracção importante na medida em que, por exemplo, na realização de um serviço, tendo nós a garantia de que a camada de domínio se encontra correctamente testada, todos os problemas encontrados provêem de erros nessa mesma camada e não em camadas inferiores.

Testes de integração

Depois de realizada a integração de ES e SD, foram realizados testes de coerência. Realizaram-se testes que lidam directamente com os serviços do PagAmigo e LargaCaixa , tanto no modo local como remoto. Para confirmar o sucesso nestes testes de coerência, foram realizadas trocas entre modo local e remoto e em cada uma dessas trocas realizados donativos e interacções com o serviço LargaCaixa.

O controlo do sucesso destes testes, em modo remoto, foi feito através da consulta do estado das bases de dados.