



Universidade do Porto
Faculdade de Engenharia

FEUP



Joel Rebelo de Castro Alvim

Portal de Serviços na Vodafone Telecel, SA

Confidencial

Portal de Serviços na Vodafone Telecel S.A.

Joel Rebelo de Castro Alvim

Relatório do Estágio Curricular da LEIC 2001/02

Orientador na FEUP: Prof. Manuel Ricardo

Orientador na Vodafone Telecel S.A.: Eng. Henrique Necho



Faculdade de Engenharia da Universidade do Porto
Licenciatura em Engenharia Informática e Computação

2002-09-10

UNIVERSIDADE DO PORTO
Faculdade de Engenharia
BIBLIOTECA M

N.º 65324

CDU 004(047.3)

Data 14 / 3 / 2003

À Gata

Resumo

O presente documento destina-se a ilustrar o projecto “Portal de Serviços na Vodafone Telecel S.A.”, desenvolvido na empresa do mesmo nome. Este projecto constitui uma solução para agregar aplicações já existentes, disponibilizadas conjuntamente e de uma forma segura através de um Portal. Para esse efeito foi utilizado o produto *iPlanet Portal Server* da Sun.

Neste documento referem-se pormenorizadamente as tecnologias e outros produtos utilizados em desenvolvimento. É ainda feita uma apresentação sobre os objectivos principais que o projecto visa atingir, bem como normas da empresa a cumprir. Descrevem-se aqui as aplicações disponibilizadas pelo Portal, e em que medida tiveram ou não de ser alteradas para conseguir a sua integração. Apresenta-se também uma solução de autenticação de utilizadores no Portal, através do seu *hardware*. São ainda documentados os primeiros mecanismos de centralização de utilizadores e palavras-chave ao longo da rede Vodafone Telecel S.A.

Encontram-se anexados neste documento alguns capítulos que descrevem fases mais residuais, como sejam o sítio *Web* do projecto na rede interna Vodafone Telecel S.A, a construção de uma nova aplicação a disponibilizar no Portal, e a implementação na aplicação WebParceiros, do mecanismo de segurança baseado em equipamento físico desenvolvido para o Portal.

Este projecto criou os fundamentos iniciais para disponibilizar futuramente aplicações e serviços de um modo seguro, flexível e centralizado.

Agradecimentos

A todos os colaboradores da Vodafone Telecel S.A, em especial aos Engenheiros Rui Carvalho, Mário Correia, Jorge Sousa e Henrique Necho, por me terem proporcionado um ambiente agradável de trabalho em equipa, mostrando-se totalmente disponíveis e profissionais em todos os momentos.

Ao meu colega Miguel Pereira por cultivar a sua iniciativa e perseverança na resolução de problemas.

A todos os meus colegas e mentores da FEUP, e em especial aos Professores Manuel Ricardo e Raul Vidal pelo seu rigor e dedicação académicos.

A todas as pessoas que estiveram presentes nos maus momentos, e que ao mesmo tempo festejaram os bons.

Índice de Conteúdos

1	Introdução.....	1
1.1	Sobre Este Relatório	1
1.2	Apresentação da Vodafone Telecel S.A.	1
	Perfil da Empresa.....	2
	Pontos Fortes.....	2
1.3	O Estágio – Projecto Portal de Serviços	3
1.4	Normas de Trabalho na Vodafone Telecel S.A.....	4
	Normas Específicas Deste Estágio	4
1.5	Organigrama do Projecto.....	5
2	Tecnologias Transversais.....	6
2.1	Tecnologias.....	6
	JAVA	6
	J2EE	7
	EJB	8
	Servlets	9
	JSP (Java Server Pages).....	9
	LDAP	9
	CVS.....	11
2.2	Software Utilizado	12
	iPlanet Portal Server 3.0	12
	BEA Weblogic 5.1	15
	Oracle 8i (8.1.7)	17
	Sun ONE Directory Server 5.1	17
	Apache ANT	18
3	Considerações Sobre o Projecto.....	19
3.1	Aplicações Envolvidas	20
	Aplicações Já Existentes.....	20
	Aplicações Novas.....	20
3.2	Plano de Acção.....	21
3.3	O iPlanet Portal Server como Solução Integrada.....	21
4	Desenvolvimento Efectuado.....	23
4.1	Arquitectura do Projecto Piloto.....	23
4.2	Configurar o iPlanet Portal Server	24
	Administração do iPlanet Portal Server.....	25
	Domínios Disponíveis	25
	Criação de Utilizadores e Papéis	26
	Configuração da Gateway.....	27
	Configuração e Preparação do Serviço de Directórios	27
4.3	Desenvolver sobre o iPlanet Portal Server	28
	API's Java de Programação.....	28
4.4	Módulo de Autenticação – LdapMis	29
	Arquitectura do Módulo de Autenticação	29
	Implementação do Módulo de Autenticação	30

- 4.5 Núcleo Operativo do Portal34
- 4.6 Single Sign-On39
 - Alterações ao Directório LDAP 39
 - iwtUserApps - Um Novo Componente do iPlanet Portal Server 40
 - Servlets Adaptadoras 41
 - Mecanismos de Single Sign-On 42
- 4.7 Integração do Portal com Aplicações Existentes44
 - Replicação de Utilizadores das Aplicações para o Directório LDAP 44
 - Arquitectura 44
 - Descrição de Implementação 45
- 5 Conclusão 48
- Referências e Bibliografia 50
- ANEXO A: FeedbackReports, uma Nova Aplicação 51
- ANEXO B: Sítio Web do Estágio..... 55
- ANEXO C: Ramificação ao Projecto do Portal de Serviços..... 60
- Anexo D: Mapa Do Estágio 64

Índice de Figuras

Fig. 1-1 Organigrama do Projecto "Portal de Serviços" 5

Fig. 2-1 Tecnologias – Estado actual da gama de plataformas Java..... 7

Fig. 2-2 Tecnologias - Arquitectura típica J2EE..... 7

Fig. 2-3 Tecnologias – Exemplo de um Directório LDAP..... 10

Fig. 2-4 Tecnologias - Entrada LDAP com múltiplos atributos..... 10

Fig. 2-5 Tecnologias - Redireccionamento de pedidos LDAP 11

Fig. 2-6 Software - Arquitectura do iPlanet Portal Server 12

Fig. 2-7 Software – Estrutura em papéis do iPlanet Portal Server..... 13

Fig. 2-8 Software - Arquitectura da Netlet..... 15

Fig. 2-9 Software - Arquitectura do BEA WebLogic..... 16

Fig. 3-1 Portal de Serviços - Diagrama de Casos de Uso 19

Fig. 3-2 Portal de Serviços - Ecrã típico de um Portal construído com o *iPlanet Portal Server* 22

Fig. 4-1 Portal de Serviços - Arquitectura Física do Piloto..... 23

Fig. 4-2 Portal de Serviços - Ecrã de Administração do iPlanet Portal Server..... 25

Fig. 4-3 Portal de Serviços - Alteração do Aspecto de Entrada nos Domínios 26

Fig. 4-4 Portal de Serviços Interdependência entre Papéis do iPlanet Portal Server 26

Fig. 4-5 Portal de Serviços - Estrutura LDAP de Utilizadores 28

Fig. 4-6 Módulo de Autenticação - Diagrama de Estados..... 29

Fig. 4-7 Módulo de Autenticação - Diagrama de Sequência..... 30

Fig. 4-8 Módulo de Autenticação - Administração do Componente iwtUserMac 31

Fig. 4-9 Módulo de Autenticação - Diagrama de Componentes 31

Fig. 4-10 Módulo de Autenticação - Diagrama de Classes..... 32

Fig. 4-11 Diagrama de Sequência da Activação de Máquina 34

Fig. 4-12 Núcleo Operativo - Diagrama de Estados 34

Fig. 4-13 Área de Trabalho do Portal de Serviços 35

Fig. 4-14 Núcleo Operativo - Diagrama de Componentes 36

Fig. 4-15 Núcleo Operativo - Diagrama de Sequência 37

Fig. 4-16 Núcleo Operativo - Diagrama de Classes..... 38

Fig. 4-17 Single Sign-On - Alteração ao LDAP 39

Fig. 4-18 Single Sign-On - Administração de Aplicações 41

Fig. 4-19 *Single Sign-On* - Diagrama de Classes 41

Fig. 4-20 Single Sign On - Diagrama de Estados Antigo da Aplicação WebParceiros 42

Fig. 4-21 Single Sign On - Novo Diagrama de Estados da Aplicação WebParceiros 43

Fig. 4-22 Single Sign On - Diagrama de Estados da Aplicação FeedbackReports 43

Fig. 4-23 Replicação - Arquitectura Lógica..... 45

Fig. 4-24 Replicação - Diagrama de Estados do Mecanismo de Replicação 46

Fig. A-1 FeedbackReports - Diagrama de Casos de Uso..... 51

Fig. A-2 FeedbackReports - Diagrama de Fluxos..... 51

Fig. A-3 FeedbackReports - Diagrama de Classes..... 52

Fig. A-4 FeedbackReports - Diagrama de Componentes 53

Fig. A-5 FeedbackReports - Ecrã de Submissão de Dados 53

Fig. A-6 FeedbackReports - Ecrã de Relatórios 54

Fig. B-1 Sítio Web Interno - Diagrama de Casos de Uso 55

Fig. B-2 Sítio Web Interno - Diagrama de Fluxos 55

Fig. B-3 Sítio Web Interno - Diagrama de Classes 56

Fig. B-4 Sítio Web Interno Diagrama de Componentes..... 56

Fig. B-5 Sítio Web Interno - Ecrã de Entrada 57

Fig. B-6 Sítio Web Interno - Ecrã de Listagem de Documentos 57

Fig. B-7 Sítio Web Interno - Ecrã de Inserção de Grupos..... 58

Fig. B-8 Sítio Web Interno - Ecrã de Inserção de Novos Documentos 58

Fig. C-1 WebParceiros - Diagrama de Estados do Novo Login..... 60

Fig. C-2 WebParceiros - Alterações ao Diagrama de Classes 61

Fig. C-3 WebParceiros - Sequência de Estados do Novo Login 62

Fig. C-4 WebParceiros - Diagrama de Sequência do Novo Login..... 63

1 Introdução

Nos dias de hoje assiste-se, cada vez mais, a uma disponibilização pela *Internet*, de serviços essenciais que envolvem transacções de dinheiro.

Este tipo de serviços são sensíveis para uma empresa, devido ao risco de fraude inerente. No entanto são os serviços mais apreciados pelos clientes, porque lhes permite fazer a partir de casa operações que de outro modo iriam exigir uma possível deslocação a um centro de atendimento, apoio ou Caixa Multibanco.

Disponibilizar e automatizar estes serviços pela *Internet* trará vantagens para a empresa e para o cliente. O cliente ficará satisfeito pelo comodismo e eficácia do serviço prestado. Por seu lado, a empresa lucra nessa fidelização, e não menos importante, pode reduzir os custos dos seus centros de atendimento e serviços de apoio.

A Vodafone Telecel S.A. têm sido pioneira na implementação de novas tecnologias, e nesta área não quer ficar atrás, efectuando esforços para disponibilizar um cada vez maior número deste tipo de serviços *online*.

1.1 Sobre Este Relatório

Este relatório reflecte o fruto do trabalho de uma equipa de projecto na Vodafone Telecel S.A, constituída essencialmente por mim e pelo colega de estágio Miguel Pereira. Assim sendo, existiu algum esforço conjunto das duas partes para atingir determinados objectivos, como sejam a documentação e investigação das plataformas utilizadas. No entanto no desenrolar do projecto, cada elemento acabou por efectuar tarefas mais distintas. Com o risco de não se perder o contexto do que foi feito, e porquê, este relatório refere aspectos não inteiramente desenvolvidos pelo autor e nesses casos denota-se o facto por escrito.

A estrutura deste relatório encontra-se dividida do seguinte modo: o 1º capítulo apresenta o estágio, a empresa e as normas de trabalho. Seguidamente, o 2º capítulo descreve as tecnologias e *software* utilizado. Este capítulo, resultante de uma investigação inicial conjunta, foi escrito em co-autoria com o colega de estágio Miguel Pereira. O 3º capítulo descreve com mais pormenor os objectivos do projecto, e outras aplicações com que irá interagir. Segue-se o capítulo 4º que visa documentar todo o desenvolvimento efectuado. Finalmente o capítulo 5º contém a conclusão do trabalho, sua utilidade para a empresa e perspectivas futuras sobre o mesmo. Este relatório têm ainda em anexo 3 capítulos que descrevem respectivamente a construção de uma nova aplicação chamada *FeedbackReports*, o desenvolvimento do sítio *Web* interno do Projecto, e a implementação de um mecanismo de segurança numa aplicação já existente: *WebParceiros*.

Por motivos de simplicidade da escrita e leitura do texto, a plataforma *iPlanet Portal Server* é intitulada por *iPS* em vários pontos deste relatório.

1.2 Apresentação da Vodafone Telecel S.A.

A Vodafone Telecel S.A é uma empresa dedicada à prestação de serviços móveis de Voz e de Dados, operando sobre uma rede móvel GSM¹ de âmbito nacional, possuindo ainda uma

¹ *Global System for Mobile Communications*, permite voz e dados (como mensagens de texto)

licença de exploração do sistema UMTS². Adicionalmente, disponibiliza também serviços fixos especialmente dedicados ao segmento de mercado empresarial.

Inicialmente fundada em 1992, sob o nome de Telecel S.A, rapidamente desenvolveu o mercado celular português oferecendo serviços úteis e acessíveis aos clientes. Em Janeiro de 2001 juntou-se ao grupo Vodafone, fortalecendo a sua imagem, força, tecnologia e posição.

Nesta data, a Vodafone Telecel S.A. constitui o segundo maior operador no mercado Português de Telecomunicações, detendo a segunda maior cota do mercado total das receitas. É também reconhecida como uma das melhores empresas portuguesas e um dos melhores operadores celulares internacionais.

Os seus objectivos, claramente definidos, enunciam-se como:

- Satisfação dos Clientes
- Satisfação dos Accionistas
- Satisfação dos Colaboradores da Empresa

Perfil da Empresa

A Vodafone Telecel S.A, fundada a 18 de Outubro de 1992, dedica-se à satisfação das necessidades dos seus clientes, procurando criar ofertas de serviços com qualidade e valor superiores.

A sua posição no mercado define-se como sendo a de um operador de Telecomunicações e Informação, com um negócio base centralizado na comunicação móvel, apostando ainda na oferta de serviços de comunicação de dados.

Para manter uma posição de liderança nos segmentos de mercado celular empresarial, desenvolve ofertas complementares dos serviços fixos, participando ainda em projectos de Tecnologias de Informação e Redes que potenciam o aumento das receitas e reduções significativas de custos. Como resultado, fidelizam-se os clientes.

A Vodafone Telecel S.A. possui ainda uma vasta oferta em todos os segmentos de mercado, reconhecida positivamente pelos seus clientes como uma das melhores.

Pontos Fortes

Apoio a Clientes

Qualquer cliente Vodafone pode recorrer a um serviço de apoio disponível 24 horas por dia, e certificado pela normal de qualidade ISO 9002, atribuída pela Associação Portuguesa de Certificação.

Estrutura Accionista

A Vodafone Telecel S.A. está cotada na Euronext Lisbon desde 1996, sendo uma das empresas com maior capitalização bolsista.

² Universal Mobile Telecommunications System, sistema de comunicação de 3ª geração

Serviços Fixos

Em Janeiro de 2000, a Vodafone Telecel S.A. lançou um serviço de acesso indirecto, utilizando o prefixo 1091 nas chamadas nacionais e internacionais de rede fixa, com um tarifário mais competitivo.

Em Junho de 2000 começaram a ser disponibilizados serviços de acesso directo na rede fixa, através de tecnologia FWA (*Fixed Wireless Access*).

Em Janeiro de 2001, disponibilizou-se um acesso indirecto nas chamadas locais, regionais e para redes móveis nacionais.

Cobertura Global

A Vodafone Telecel S.A. serve perto de 100% da população de Portugal Continental e Regiões Autónomas, nas frequências de 900 Mhz e 1800 Mhz.

O serviço de *Roaming* é disponibilizado aos clientes em todos os continentes, e mais de uma centena de países e territórios.

Internet

Em Setembro de 1999, a Vodafone Telecel S.A. tornou-se o 1º operador a oferecer um serviço alargado e gratuito à *Internet*, tendo sido também pioneira na utilização da tecnologia WAP (*Wireless Access Protocol*) a partir da rede móvel.

Em Novembro de 2001, a Vodafone Telecel S.A. realizou uma parceria com o grupo Vizzavi Europe, com vista a implementar um Portal em Portugal com conteúdos nacionais e internacionais, disponibilizando ainda serviços de comunicação e entretenimento.

Serviços Úteis e Inovadores

A Vodafone Telecel S.A. disponibiliza serviços complementares aos da rede móvel GSM, sublinhando a sua visão em satisfazer o cliente. Entre esses serviços destacam-se os de Mensagens Multimédia, *Voice Mail* por voz, Correio Multimédia, Localização Celular, Redes Integradas de Voz e Dados, Assistência Expresso, Telemultibanco, *Reuters* e MMS³.

1.3 O Estágio – Projecto Portal de Serviços

O estágio pretende integrar e facilitar a disponibilização de serviços da Vodafone Telecel S.A. através da *Internet*.

A empresa possui algumas aplicações que dispõe para os seus clientes e agentes. Por vezes cada uma delas utiliza a sua própria gestão de *login's* e palavras-chave, base de dados de utilizadores e esquemas lógicos de representação. O esforço de manutenção separado de cada um destes componentes torna-se por vezes difícil. Centralizar de algum modo todos estes serviços, seria bastante vantajoso. Disponibilizá-los conjuntamente através de um Portal, de uma forma segura, fiável e eficaz seria o ideal.

É à luz destes objectivos que surgiu esta proposta de estágio: realizar alguns testes e formar um piloto com a plataforma sugerida (*iPlanet Portal Server*). Este piloto irá disponibilizar um Portal de acesso às aplicações fornecidas para os Agentes da rede de distribuição da Vodafone.

³ *Multimedia Messaging System*, permite enviar mensagens com dados Multimédia.

Caso este piloto se revele fiável, poder-se-á futuramente incluir serviços e aplicações que são disponibilizadas a outro tipo de clientes que não só aos Agentes.

1.4 Normas de Trabalho na Vodafone Telecel S.A.

A Vodafone Telecel S.A. utiliza normas de trabalho comuns que devem ser cumpridas por todos, de modo a facilitar a gestão dos projectos, bem como integração de novas equipas futuras.

Um projecto interno deverá estar sempre a cargo de uma direcção responsável. Numa fase inicial é sempre construído um sítio *Web* na rede interna da Vodafone, para informar e disponibilizar documentação produzida durante o projecto.

Uma vez iniciado o projecto, este passa por 3 fases distintas:

- Fase de Desenvolvimento, em que se constrói um ambiente mínimo para testar, criar e evoluir o projecto. São ainda criados documentos de *General Design*⁴ e *Detail Design*⁵ para entregar aos elementos responsáveis pela fase a seguir.
- Fase de Pré-Produção (também conhecida como *System-Test*), em que o projecto é implementado numa arquitectura física semelhante à final. Nesta fase, são realizados testes que visam detectar falhas a nível de carga, segurança, usabilidade da aplicação, e ainda erros gerais não detectados na fase anterior.
- Fase de Produção, na qual o projecto é integrado numa arquitectura final disponibilizada aos clientes.

Estas 3 fases repetem-se ciclicamente na adição de novas funcionalidades e evolução do projecto.

Normas Específicas Deste Estágio

Foi acordado que todas as semanas, se realizaria uma reunião semanal, na qual se debateria o estado actual do projecto, fomentando novas ideias, soluções e resolução de problemas detectados.

Para qualquer reunião, teria de ser elaborada uma acta correspondente, focando os pontos essenciais debatidos. Essa acta iria conter os passos a efectuar durante a semana corrente, e seria sujeita a aprovação pela direcção do projecto.

Todas as actas produzidas iriam estar disponíveis aos elementos do projecto, num local partilhado na rede interna da Vodafone. Deste modo seria possível observar com precisão, o percurso do projecto, o que foi feito, e os passos e decisões importantes tomados.

⁴ Documenta globalmente a aplicação produzida, incluindo o seu fluxo e arquitectura.

⁵ Documenta detalhadamente o desenvolvimento efectuado, incluindo o código produzido.

1.5 Organigrama do Projecto

Os integrantes desta equipa organizam-se do seguinte modo:

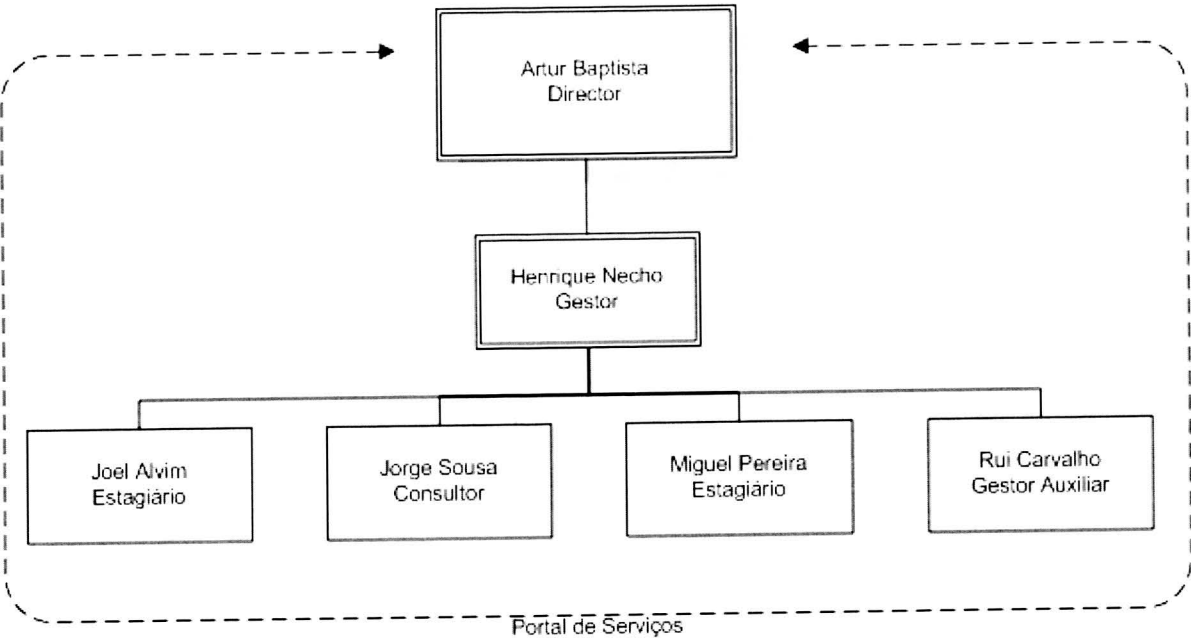


Fig. 1-1 Organigrama do Projecto “Portal de Serviços”

Os estagiários Joel Alvim e Miguel Pereira encarregaram-se da maior parte do desenvolvimento efectuado, sendo apoiados em todo o momento pelo gestor auxiliar Rui Carvalho, na indisponibilidade de contactar o gestor Henrique Necho.

O consultor Jorge Sousa apenas integrou a equipa nos últimos 2 meses do projecto, tendo supervisionado o código desenvolvido e fornecido ideias para o melhorar.

2 Tecnologias Transversais

Este capítulo destina-se a ilustrar as tecnologias e suportes lógicos transversais a todo o projecto. Sempre que for pertinente, as tecnologias específicas a certas partes do trabalho serão discutidas nos respectivos capítulos, para uma maior compreensão dos contextos em que foram utilizadas.

A Vodafone, como empresa competitiva que é, possui uma panóplia de aplicações actualmente em uso e que suportam todo o seu negócio, em áreas tão distintas como seja na área de bases de dados, aplicações de gestão das relações com clientes (CRM, *Costumer Relationship Management*), servidores aplicativos que alojam toda a lógica de negócio inerente aos seus processos, contabilidade e suporte ao planeamento de recursos (ERP, *Enterprise Resource Planner*), gestão documental, *workflow*, correio electrónico corporativo, etc.

Com um número tão grande de aplicações o processo de manutenção de utilizadores e palavras-chave torna-se essencial mas também algo moroso. Nesta área centra-se este projecto, que utiliza directa ou indirectamente quase todas as aplicações, mas centra-se especialmente nas agregação de utilizadores (existentes em várias bases de dados) e aplicações (existentes em múltiplos servidores *Web* e/ou aplicativos).

Um dos requisitos transversais a toda a empresa, e por inerência aos projectos desenvolvidos no seu seio, é a segurança. Por este facto a investigação tecnológica levada a cabo teve sempre essa preocupação em mente.

2.1 Tecnologias

JAVA

Linguagem de programação criada pela Sun em 1995, tendo como objectivos juntar a facilidade de utilização à versatilidade do C++ e alcançar a portabilidade entre máquinas. Esta linguagem orientada por objectos ganhou um forte apoio tanto académico, pela sua curva de aprendizagem rápida, como a nível empresarial, pela sua flexibilidade.

Com o advento da plataforma J2EE, em 2000, foi dado mais um passo na sua entrada no mercado empresarial, obrigando a sua principal concorrente, a Microsoft, a propor uma plataforma e linguagens com algumas características semelhantes.

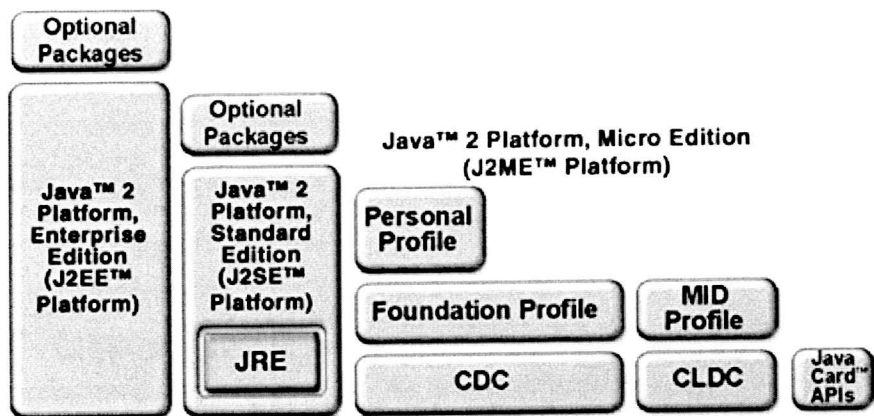


Fig. 2-1 Tecnologias – Estado actual da gama de plataformas Java

Também por volta do ano 2000 foi introduzida a *Java 2 Platform, Micro Edition*, que fornece uma arquitectura modular e escalável que suporta vários tipos de dispositivos móveis com diferentes funcionalidades, suporte físico, memória, etc.

Um fabricante de dispositivos móveis que queira suportar esta plataforma apenas terá que escolher a configuração e perfil certos para as características do seu dispositivo.

Entretanto houve especializações da linguagem, como sejam as *Servlets*, *JSP's*, *JDBC* além do já referido *J2EE*. Na secção seguinte serão descritas estas tecnologias.

J2EE

A tecnologia Java 2 Platform, Enterprise Edition (*J2EE*) aborda o desenho e desenvolvimento de aplicações empresariais, usando componentes. Esta plataforma oferece um modelo aplicacional distribuído e multi-ligado (*multitiered*). Além disso possui um modelo de segurança unificado e um controlo de transacções flexível, possibilitando ainda a reutilização de componentes e troca de dados através de *XML*.

Uma característica importante desta tecnologia é que as soluções desenvolvidas não se encontram dependentes de nenhum produto específico; o consumidor final pode escolher o produto que melhor se adapte aos seus requisitos.

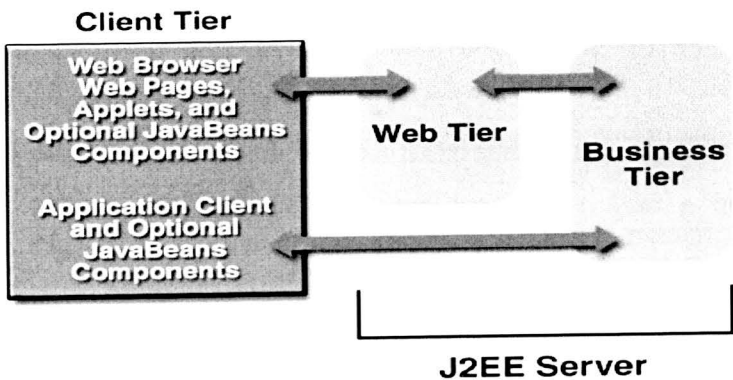


Fig. 2-2 Tecnologias - Arquitectura típica J2EE

Componentes J2EE

As aplicações J2EE são constituídas por componentes. Um componente J2EE é uma unidade de software funcional e específica que é depois integrada numa aplicação J2EE, comunicando com os restantes componentes. A especificação J2EE oficial define o seguinte:

- Clientes aplicacionais e *Applets* são componentes que correm no cliente.
- *Java Servlet* e *Java Server Pages* (JSP) são componentes *Web* e correm no servidor.
- EJB's são componentes de negócio que correm no servidor

Os componentes J2EE são escritos na linguagem de programação *Java*, sendo compilados como qualquer programa nessa linguagem. As únicas diferenças são:

- Verificação da validade formal e compatibilidade com a especificação oficial.
- Assemblagem numa aplicação J2EE.
- Entrega para produção onde são executados e geridos por um servidor J2EE.

EJB

Os EJB's são componentes *Java* reutilizáveis que implementam lógica de negócio e permitem desenvolver aplicações empresariais distribuídas e baseadas em componentes. Os EJB's residem num contentor (*EJB Container*) que fornece serviços como persistência, segurança, transacções e concorrência, sendo assim totalmente portáteis para qualquer servidor que seja compatível com as normas definidas pela Sun.

Componentes de um EJB

- Interface remota, que expõe a lógica de negócio ao cliente
- Interface local (*home interface*), que os clientes usam para criar, localizar e apagar instâncias de EJB's.
- Classe *Bean*, que implementa a lógica de negócio

Modo de Funcionamento de um EJB

Criação de um EJB:

- Codificação da lógica de negócio da aplicação numa classe.
- Especificação dos parâmetros de entrega (*deployment*) num *deployment descriptor*.
- Acondicionamento do EJB num ficheiro JAR⁶.
- Configuração do servidor aplicacional para carregar e executar esse EJB

A partir daqui, as aplicações clientes conseguem localizar o EJB e invocar os métodos definidos na interface remota.

⁶ Ficheiro JAR – ficheiro que empacota e opcionalmente comprime outros ficheiros, normalmente usado em classes *Java*

Servlets

Uma *Servlet* é uma classe *Java* que corre num servidor compatível. As *Servlets* HTTP são um caso especial que recebem e manipulam um pedido HTTP, fornecendo também uma resposta HTTP, normalmente na forma de uma página HTML.

As *Servlets* HTTP têm acesso a todas as interfaces de programação do *Java*, incluindo todas as classes do JDK (*Java Development Kit*), residindo aí uma das suas maiores vantagens, já que transporta para o domínio das aplicações *Web* o poder de uma linguagem de programação completa.

JSP (Java Server Pages)

As JSP são páginas dinâmicas para a *Web*, especificadas pela Sun Microsystems que combinam HTML com código *Java*.

Ao criar conteúdo dinâmico, é mais conveniente escrever JSP's do que *Servlets* HTTP, porque no primeiro caso mistura-se código *Java* directamente nas páginas HTML. Nas *Servlets*, faz-se *output* de código HTML dentro do código em *Java*, o que torna o trabalho de *Designers Web* muito mais difícil.

LDAP

O LDAP (*Lightweight Directory Access Protocol*) é um protocolo para gerir e aceder a um serviço de directórios na Internet, baseado numa arquitectura cliente-servidor.

Um serviço de directórios é uma aplicação distribuída de base de dados, cujo fim é o de gerir entradas e atributos num directório, tornando-os disponíveis a outros utilizadores e aplicações.

Um directório é constituído por entradas que contém informação descritiva. Esta informação é armazenada em atributos. Um atributo pode ter mais do que um valor, e pode mesmo chegar a conter dados binários (p.ex: uma imagem).

Um serviço de directórios é semelhante a uma base de dados mas tende a guardar valores mais descritivos. Além disso a informação é muito mais vezes lida do que escrita, e como tal não é necessário implementar modelos de transacções complexos e de *roll-back*⁷.

Organização de directórios nos servidores LDAP

Sendo um serviço global de directórios, os dados no LDAP encontram-se organizados hierarquicamente, começando no topo e dividindo-se em ramos. Ao nível mais alto da hierarquia, as entradas representam grandes organizações e, abaixo destas, os ramos conduzem a organizações mais pequenas, podendo acabar finalmente em indivíduos ou recursos específicos.

⁷ Mecanismo numa base de dados que permite desfazer as actualizações até ao último estado válido

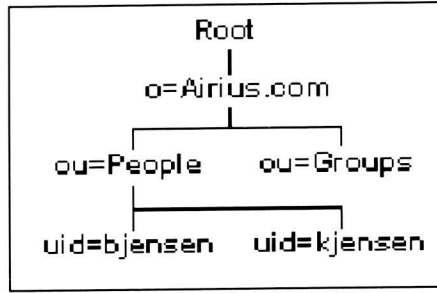


Fig. 2-3 Tecnologias – Exemplo de um Directório LDAP

Cada entrada é representada unicamente por um *Distinguished Name* (DN). Um DN é constituído por um nome que identifica essa entrada como sendo única naquele nível hierárquico, e por um caminho de nomes que traçam o caminho desde essa entrada até ao topo, ou *root*.

Neste caso, “uid” representa o identificador único da entrada, “ou” representa a unidade organizacional à qual a entrada pertence e, finalmente, “o” representa a entrada de mais alto nível.

Os dados são representados no formato de (*atributo:dados*). No caso anterior, o *Distinguished Name* corresponde a “dn:uid=bjensen,ou=People,o=airius.com”.

Esta figura ilustra esta situação:

```

dn: uid=bjensen, ou=people, o=airius.com
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
cn: Babs Jensen
sn: Jensen
givenName: Babs
givenName: Barbara
mail: bjensen@airius.com
  
```

Fig. 2-4 Tecnologias - Entrada LDAP com múltiplos atributos

Pode-se dizer ao directório quais os atributos permitidos ou obrigatórios numa dada entrada, definindo classes de objectos para essa entrada. Para se usarem classes de objectos é necessário primeiro identificá-las ao directório LDAP, adicionando-as esquema (*schema*). Um esquema constitui todo o conjunto de classes que o servidor LDAP reconhece como sendo válidos. Para tal, a verificação de esquemas no servidor deve estar ligada, permitindo que o servidor verifique se os atributos obrigatórios de uma dada classe existem, e que não existem atributos não permitidos no directório. É aconselhável não alterar muito o esquema base. Caso esse esquema seja demasiado limitador, pode ser melhorado com o uso de classes. No entanto, mudar completamente o esquema já existente pode levar a problemas de interoperabilidade entre diferentes servidores LDAP.

Os dados armazenados num directório LDAP podem ser distribuídos por outros directórios LDAP. Por exemplo, um servidor LDAP que armazene informação de Airius.com pode conter entradas representando unidades organizacionais e indivíduos da América do Norte, enquanto que outro servidor LDAP pode conter as mesmas entradas, com dados referentes à Europa. Alguns servidores LDAP estão definidos para redireccionar pedidos para outros servidores LDAP. Por exemplo, se um servidor de Airius.com receber um pedido de informação acerca de um empregado no ramo Airius Corporation, este servidor pode redireccionar o pedido para o servidor LDAP no ramo Airius Corporation. Deste modo um conjunto de servidores LDAP pode aparentar ser um só directório, porque mesmo que um servidor LDAP não contenha a informação pretendida, o pedido pode ser redireccionado para o servidor que possua essa informação.

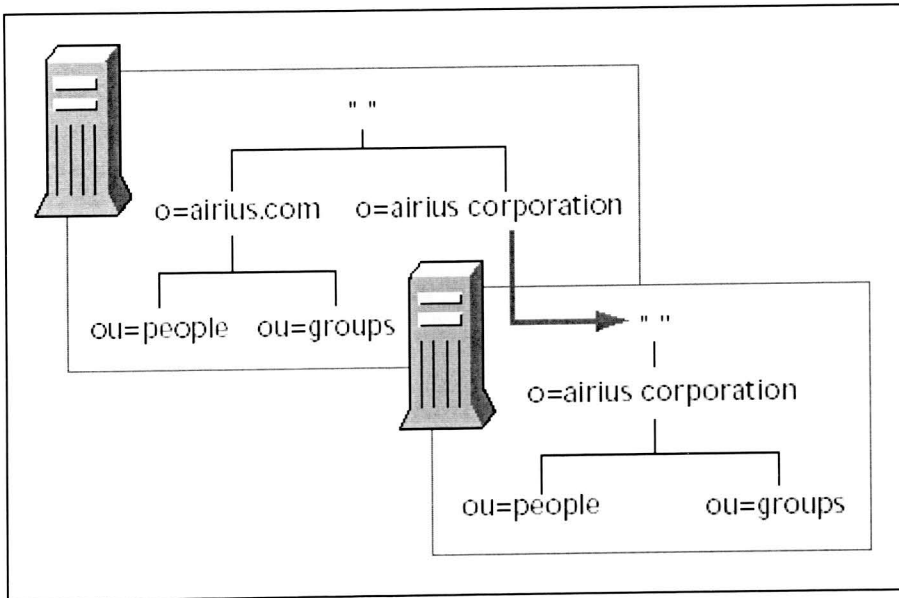


Fig. 2-5 Tecnologias - Redireccionamento de pedidos LDAP

CVS

O CVS⁸ permite gerir um repositório de versões, tornando-se especialmente útil quando existem várias pessoas que podem modificar os mesmos ficheiros. Este sistema actua

⁸ Significa *Concurrent Versions System*

mantendo uma história de alterações, tendo cada uma dessas alterações a o autor, a data, a hora e um comentário.

Assim, este sistema permite responder a questões como:

- Quem fez determinada alteração?
- Quando foi feita?
- Porque foi feita?

2.2 Software Utilizado

iPlanet Portal Server 3.0

Arquitectura:

O *iPlanet Portal Server* (ou iPS, como referido ao longo deste documento) permite uma ligação segura entre uma máquina residente na *Internet* e uma *Intranet* privada. Essa *Intranet* está normalmente delimitada por uma *Gateway* e por *firewalls* que inibem o acesso público a áreas privadas e também o acesso privado a certas áreas públicas.

Uma *Intranet* empresarial proporciona serviços importantes e necessariamente não-acessíveis ao comum dos utilizadores da *Internet*. No entanto, existem serviços que certos utilizadores (clientes, parceiros, empregados fora do local de trabalho) por vezes necessitam; neste contexto entra o uso de uma rede privada virtual (VPN, *Virtual Private Network*). Um serviço destes encapsula uma rede privada sobre uma rede pública.

Este software constitui uma maneira de proporcionar uma sessão virtual segura usando a rede pública (*Internet*). Neste aspecto o *iPlanet Portal Server* implementa uma VPN mas com algumas vantagens em relação às VPN's normais, a saber:

- Não é necessária configuração a nível de cliente, usada por exemplo em redes PPTP.
- Está disponível a todos os clientes via *browser Web*.
- Ao contrário das redes baseadas em *IPsec* em que a cifragem de dados se encontra ao nível 3 do modelo OSI, com este produto a cifragem encontra-se ao nível aplicacional; assim, apenas necessita de haver confiança a nível de *browser Web*.

Tipicamente a configuração usada é a apresentada na figura 2-6:

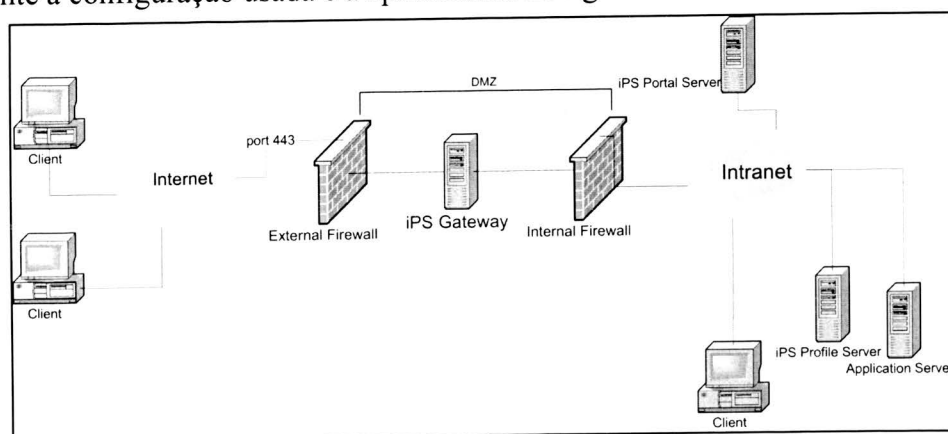


Fig. 2-6 Software - Arquitectura do iPlanet Portal Server

Existe um componente, o *iPortal Server Gateway*, que reside numa zona desmilitarizada (DMZ⁹); esse componente serve de ponte segura entre o *browser* do cliente (correndo sobre HTTPS) e o servidor iPS, que se encontra dentro da *Intranet* da empresa. Por sua vez, este servidor comunica com os recursos que se querem tornar acessíveis, tais como servidores aplicativos e servidores *Web*.

As funções mais importantes desempenhadas por este componente são:

- Autenticação: a autenticação dos utilizadores nos diferentes modos disponíveis é feita por este componente. Entre os modos disponíveis encontram-se o LDAP, UNIX, Domínios *Windows NT*, *Safeword*, *RSA SecurID* e certificados digitais.
- Reescrita de endereços: modificação em tempo real do código HTML e *JavaScript* de modo a rectificar os endereços, juntando-lhes o endereço da própria *Gateway* e forçando assim o browser a aceder aos conteúdos passando por este componente.
- Restrições: apenas os endereços, domínios e servidores configurados na administração são acessíveis. Assim, evita-se o caso de, depois de criada uma sessão válida na *gateway*, se aceder a um endereço não protegido por *firewall* na rede interna.

Para a autenticação e configuração do acesso dos utilizadores existe um servidor LDAP que é usado pelos dois componentes para obter informações. A componente *iPlanet Profile Server* é fornecida com o produto e é obrigatoriamente usada. Existem outro tipo de dados que podem residir noutro servidor LDAP, tais como utilizadores, palavras-chave e dados de configuração.

Arquitectura baseada em papéis (roles)

Esta arquitectura consiste numa árvore de papéis em que o(s) domínio(s) gerais (ex: Vodafone) estão no topo. Descendo nessa árvore temos papéis atribuídos a grupos de utilizadores e, por fim, a utilizadores específicos.

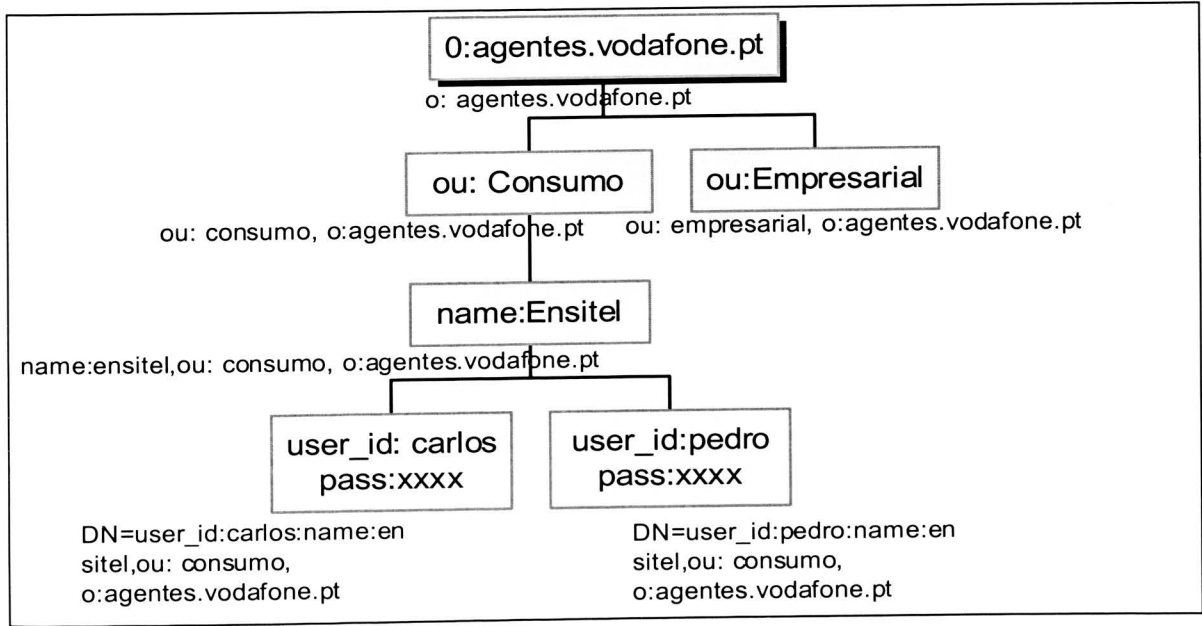


Fig. 2-7 Software – Estrutura em papéis do iPlanet Portal Server

⁹ De-militarized Zone - Consiste em duas firewalls, chamadas externa (para a rede pública) e interna (para a rede privada). Assim o acesso ao componente de rede que lá reside está protegido de ambos os lados.

Estes papéis possibilitam as seguintes funcionalidades:

- Configurar áreas de trabalho de utilizadores, relativamente a *layouts* e conteúdos.
- Definir políticas de controlo de acesso, tais como a lista de aplicações que um utilizador pode usar e URL's a que pode aceder.
- Definir o tipo de autenticação usada (LDAP, NT, etc.), e o tempo máximo de sessão
- Aceder a aplicações integradas no ambiente *iPlanet Portal Server*

Carga Balanceada

Numa configuração com várias *gateways* pode usar-se um balanceador de cargas conseguindo assim um ponto de entrada comum e deixando a cargo do balanceador o reencaminhamento de tráfego pelo melhor caminho.

Single Sign-On

Uma vez autenticado, o utilizador nunca mais terá que fornecer qualquer *login* e palavra-chave durante a vida da sua sessão no servidor.

Interfaces de programação

As interfaces seguintes são fornecidas para integrar uma qualquer aplicação na moldura aplicacional do *iPlanet™ Portal Server*:

- *Access Control and Profile*: Serve para aceder a informações de perfil por parte de uma aplicação.
- *Session*: Serve para as aplicações validarem a sessão e acederem a dados da mesma para um qualquer utilizador.
- *Logging*: Proporciona um serviço de escrita, leitura e modificação de *logs*.
- *Content Provider*: Fornece métodos para integrar conteúdo aplicacional externo no *desktop* do *iPlanet™ Portal Server*.
- *Pluggable Authentication*: Permite escrever módulos adicionais de autenticação em *Java™*.

No caso do uso do *Java* como linguagem de programação da aplicação a comunicação com o *iPlanet™ Portal Server* é facilitada pois são fornecidas classes que podem ser usadas e que tornam a camada de comunicação transparente.

Netlet

Fornece uma ligação segura entre um qualquer cliente que corra num sistema com um browser com suporte de Java, e um recurso de rede atrás de uma "firewall" empresarial.

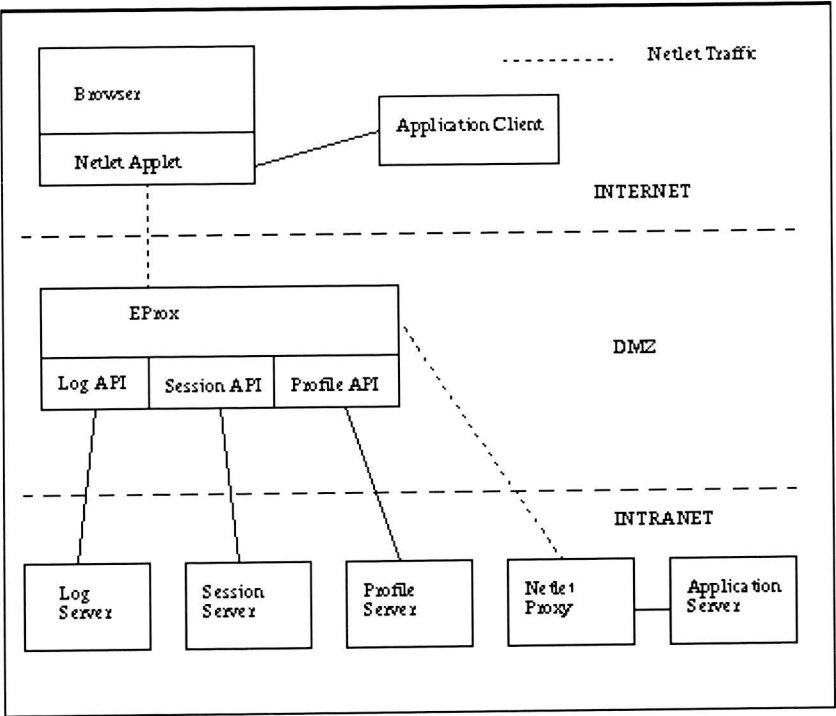


Fig. 2-8 Software - Arquitectura da Netlet

A Netlet Consiste numa *Applet Java* que cria um túnel cifrado entre a máquina cliente e uma máquina denominada *Netlet Proxy*. A partir daqui é possível, por exemplo, configurar na máquina cliente um programa de correio electrónico para aceder ao correio corporativo, cujo servidor reside na intranet empresarial.

BEA Weblogic 5.1

O servidor aplicacional BEA WebLogic é uma plataforma de comércio electrónico, que permite criar e disponibilizar rapidamente aplicações seguras, escaláveis e de fácil gestão. Os detalhes “baixo nível” do sistema são geridos transparentemente pelo servidor, sendo deixado ao arquitecto a preocupação com a lógica de negócio e a apresentação.

Arquitectura do Servidor Aplicacional WebLogic

A arquitectura típica do BEA Weblogic 5.1 baseia-se numa interpretação da arquitectura *multitier*¹⁰ e é composta por três camadas:

- a camada do cliente, compreende os programas que acedem às aplicações (browsers *Web*, aplicações *Web*, aplicações isoladas)
- a camada do meio, que contém no próprio servidor aplicacional
- a camada de bastidores, que é composta por aplicações legadas, sistemas de gestão de base de dados e aplicações de suporte ao planeamento de recursos (ERP, *Enterprise Resource Planner*).

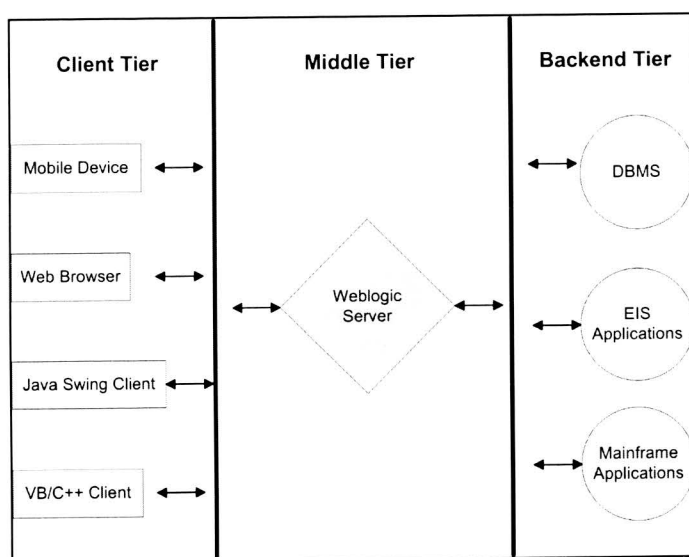


Fig. 2-9 Software - Arquitectura do BEA WebLogic

O servidor WebLogic implementa a tecnologia J2EE. As aplicações J2EE são baseadas em componentes standard e modulares; assim, um servidor que implemente a tecnologia fornece necessariamente um conjunto de serviços e controla automaticamente muitos dos detalhes e comportamentos das aplicações.

Vantagens

Algumas vantagens no uso dum servidor deste tipo são as seguintes:

- Implementação de standards abertos: a plataforma J2EE é um standard criado pela Sun acessível e implementável por qualquer pessoa.
- Portabilidade: as aplicações feitas podem facilmente ser migradas entre os vários servidores aplicacionais existentes no mercado.
- Robustez: com a sua arquitectura bem definida e modular reduz-se a hipótese de erros

¹⁰ Arquitectura dividida em várias camadas, que se interligam por fluxos de informação bidireccionais, e em que cada uma dessas camadas realiza funções bem definidas e distintas das restantes. A vantagem desta arquitectura é que se pode substituir uma camada por outra semelhante que opere do mesmo modo, sem afectar a inter-operabilidade global da arquitectura.

- Escalabilidade: podem ser acrescentados componentes e aplicações mas também modificados a apresentação gráfica de uma aplicação sem modificar a lógica de negócio já implementada.

Como contrapartida, existe a necessidade de planear e estruturar muito bem toda a aplicação sob pena de se perder em performance ou em escalabilidade.

Oracle 8i (8.1.7)

Trata-se de um SGBD (Sistema de Gestão de Base de Dados) relacional utilizado por milhares de empresas um pouco por todo o mundo. A descrição completa das características deste popular software sai do alcance deste relatório; a sua utilização neste projecto limitou-se ao uso dos *drivers* JDBC¹¹, consulta e modificação de tabelas, criação de conjuntos de tabelas e programação em PL/SQL

Sun ONE Directory Server 5.1

Utiliza um servidor LDAP, e constitui um dos componentes fundamentais do ambiente Sun ONE¹².

À medida que as empresas começam a utilizar a *Internet*, torna-se necessário gerir os potenciais utilizadores que irão usar esses mesmos serviços. Neste contexto, a solução de acesso deve ser flexível e escalável, a nível da gestão de utilizadores e mecanismos de segurança. Pode-se então armazenar e gerir perfis de utilizadores, bem como mecanismos de autenticação ao longo de todo um sítio *Web*.

As principais vantagens do *Sun ONE Directory Server 5.1*, em relação a um servidor LDAP comum são:

- Elevada disponibilidade e escalabilidade, suportando todo um negócio, à medida que cresce com potenciais utilizadores e aplicações.
- Gestão de segurança e acessos, através de ferramentas próprias para gerir os utilizadores, os seus dados e as políticas que definem os privilégios de acesso. Potencia-se a privacidade do cliente na maior parte das aplicações de comércio electrónico, ao mesmo tempo que se torna possível introduzir segurança adicional (como certificados digitais) noutras aplicações mais sensíveis.
- Integração com outras fontes de dados dentro de uma empresa, permitindo disponibilizar informação já existente, com melhor qualidade de serviço e custos de administração mais reduzidos.
- Dispõe de uma *interface* de programação (em C e Java) que permite ainda melhor integração com aplicações já existentes.

¹¹ JDBC - Java DataBase Connectivity – conjunto de normas independentes que definem a maneira de ligar um programa em JAVA e uma base de dados. Se um Sistema de Gestão de Base de Dados implementar essas normas terá que escrever código específico para o seu produto e fornecer *drivers*, tal como acontece com o Oracle

¹² Sun Open Environment Networking, plataforma de suporte ao ciclo de vida da informação de uma empresa

Apache ANT

O Ant é uma ferramenta de desenvolvimento multi-plataforma e extensível desenvolvida em *Java*. O conceito base desta ferramenta é ser parecido com o *make*¹³ mas sem as suas desvantagens. Assim, os seus ficheiros de configuração são construídos em XML¹⁴ e podem incluir comandos (chamados tarefas) tão variados como compilar, copiar, enviar através de FTP¹⁵ para o servidor de desenvolvimento, etc.

Existem extensões ao ANT que adicionam tarefas que permitem, por exemplo, construir automaticamente um EJB (compilar, verificar, empacotar, copiar).

Em suma, é uma ferramenta de eleição que reduz o esforço posto na configuração e preparação do ambiente de desenvolvimento e permite a automatização de uma miríade de tarefas.

¹³ make – ferramenta desenvolvida originalmente para o sistema operativo Unix que controla a criação de executáveis a partir dos ficheiros de código fonte de um programa.

¹⁴ XML – eXtended Markup Language

¹⁵ FTP – File Transfer Protocol

3 Considerações Sobre o Projecto

Neste projecto é necessário fornecer aos parceiros da Vodafone um acesso centralizado via *Internet* a aplicações sensíveis da empresa. Além disso, o Portal irá servir também como uma ferramenta para auxiliar o acesso de utilizadores de *Back-Office* às aplicações, dentro da própria Vodafone.

Existem diariamente queixas e problemas de utilizadores nas aplicações. Quando esses problemas não constam numa lista pré-definida de problemas comuns, a queixa passa para os utilizadores de *Back-Office*. Estes utilizadores são colaboradores internos da Vodafone, que podem simular um utilizador normal numa dada aplicação, tornando-se mais fácil descobrir as causas dos erros que ele se queixa de terem aparecido.

Todos os utilizadores dos parceiros, terão um *login* e uma palavra-chave própria. Adicionalmente, será configurado no servidor de perfis do *iPlanet Portal Server*, quais as aplicações a que esses utilizadores poderão aceder. O acesso a essas mesmas aplicações será feito através de *Single Sign-On*. Isto significa que um utilizador só tem de inserir o seu *login* e palavra-chave uma única vez: quando entra no Portal. O acesso às restantes aplicações deve ser transparente.

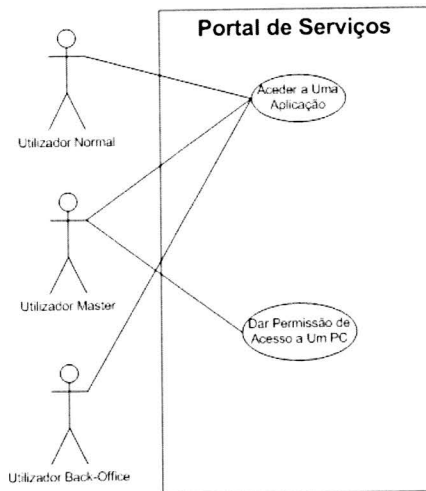


Fig. 3-1 Portal de Serviços - Diagrama de Casos de Uso

É também um requisito, que um dado utilizador normal do Portal não seja validado somente com o seu *login* e palavra-chave. Deve-se certificar que ele acede de um determinado PC, e que a esse PC foi dada permissão para aceder ao Portal. Esta permissão é configurada pelo próprio responsável por um dado agente (designado por utilizador *Master*). Este pode definir quais os PC's de uma loja que terão acesso ao Portal, e quais não terão. Esta restrição surge para evitar possíveis fraudes. Isto porque qualquer trabalhador de um agente que tivesse um par *login*/palavra-chave, poderia efectuar operações sensíveis a partir de casa, por exemplo. Note-se no entanto que um utilizador de *Back-Office* não necessita deste tipo de validação física.

Finalmente, e como meio de integrar este projecto numa possível migração futura para um ambiente mais centralizado, tornou-se necessário replicar para o directório de utilizadores do

Portal os novos utilizadores criados nas aplicações de segundo nível. Esta fase de replicação inicial terá também de ser implementada.

3.1 Aplicações Envolvidas

As seguintes aplicações constituem o corpo de serviços disponíveis aos parceiros.

Aplicações Já Existentes

WebParceiros

Com esta aplicação, um cliente pode deslocar-se ao parceiro Vodafone mais próximo da sua área de residência, e dispor de funcionalidades que lhe permitem aceder a informações específicas dos serviços (carregamento de vitaminas, alteração de tarifário, etc.), podendo inclusive efectuar algumas alterações.

Estas são processadas automaticamente sem intervenção dos serviços de apoio, o que traz vantagens a nível interno (economizam-se recursos) e externo, porque o cliente irá perceber todo o processo como sendo rápido e eficaz. Este é um dos aspectos que os clientes mais valorizam.

SFAWeb

Esta aplicação permite aos Agentes da Vodafone, aceder a informação sobre as ofertas comerciais, preços e descontos que podem efectuar sobre as mesmas. Para esse efeito utiliza o software Siebel (constitui um CRM, *Customer Relationships Management*).

Aplicações Novas

E-Mail Corporativo

O iPS possui um cliente *Web* de Correio Electrónico, que suporta o protocolo IMAP¹⁶. Basta configurar um utilizador, uma palavra-chave, um servidor IMAP e um servidor SMTP¹⁷, e tem-se todos os dados necessários para aceder à conta correspondente, de um modo funcional.

Assim é possível criar contas de correio electrónico para os agentes Vodafone na rede interna da empresa, sendo estas acessíveis pelo exterior, o que seria de outro modo impossível.

Feedback Reports

Esta aplicação visa facilitar a tarefa dos operadores 1210¹⁸ e 1291¹⁹, de reportar problemas dos utilizadores nas aplicações WebParceiros e SFAWeb. Estes relatórios podem ser tratados no momento pela aplicação, de modo a construir métricas, bem como tomar acções relevantes (como enviar notificações por correio electrónico, etc.).

¹⁶ Internet Message Access Protocol. Permite aceder a mensagens armazenadas num servidor remote, dando a impressão que se trata de um acesso local.

¹⁷ Simple Mail Transfer Protocol. Protocolo de transmissão de correio electrónico para as respectivas caixas de correio a que se destinam.

¹⁸ Operadores responsáveis pelo suporte ao cliente.

¹⁹ Operadores responsáveis pelo suporte ao cliente empresarial.

O desenvolvimento desta aplicação, especificamente desenhada para este projecto, pretende provar o quão fácil e integrado é a criação e utilização de um serviço, utilizando directamente as funcionalidades do Portal (como verificação de Sessões activas e políticas de acesso à aplicação).

Mais informação sobre o funcionamento e desenvolvimento desta aplicação, pode ser encontrada no anexo A do presente documento.

3.2 Plano de Acção

As normas da Vodafone Telecel S.A. impõem que para qualquer projecto iniciado dentro da empresa, seja construído um sítio *Web* na sua *Intranet*, contendo toda a informação e documentação produzida. Este constitui então o primeiro passo do plano, consolidando um esqueleto sólido de informação para todos colaboradores Vodafone que desejam saber mais sobre o projecto.

Segue-se uma fase de investigação, em que se procura aprofundar as possibilidades da plataforma de desenvolvimento proposta (*iPlanet Portal Server*), ao mesmo tempo que se produz documentação sumária focando os aspectos mais importantes dos documentos encontrados sobre essa plataforma e outras tecnologias auxiliares. Isto facilita a entrada de novas pessoas no projecto, permitindo-se a elas uma adaptação mais fácil.

Esta fase de investigação visa não só conhecer melhor a tecnologia que se irá usar, como permite conhecer também outras alternativas e medir pontos fortes e fracos das mesmas.

Sabendo então qual a plataforma de desenvolvimento, chega o momento de obter os recursos físicos necessários para o projecto. Forma-se aqui uma pequena máquina piloto onde será feito o desenvolvimento e testada a solução. Esta fase de desenvolvimento é a mais morosa.

A solução é testada em 3 níveis:

- Bancada de Testes: são definidos vários testes à aplicação de modo a conseguir detectar inconsistências, erros e problemas.
- Testes de *Hacking*: estes testes destinam-se a verificar a robustez da solução.
- Testes de Stress: aqui verifica-se a fiabilidade e disponibilidade da solução.

Após o sucesso destes testes, a solução é considerada viável, e implementada num ambiente de produção. Neste projecto, fica acessível por qualquer PC ligado à *Internet*.

3.3 O iPlanet Portal Server como Solução Integrada

O iPS é um Portal com bastantes funcionalidades. A nível de mecanismos de autenticação oferece alguns tão variados como RSA, SecurID, Unix, NT, Radius e LDAP. Para além disso, contém uma aplicação que permite construir uma área de trabalho personalizada (ou *Desktop*), em que se fornecem vários serviços através de canais.

Graças à sua natureza baseada em papéis (*roles*), é possível configurar variados aspectos do Portal, desde o nível de domínio até ao nível de utilizador. Isto torna esta plataforma flexível e configurável, adaptando-se facilmente à maior parte dos problemas e necessidades.

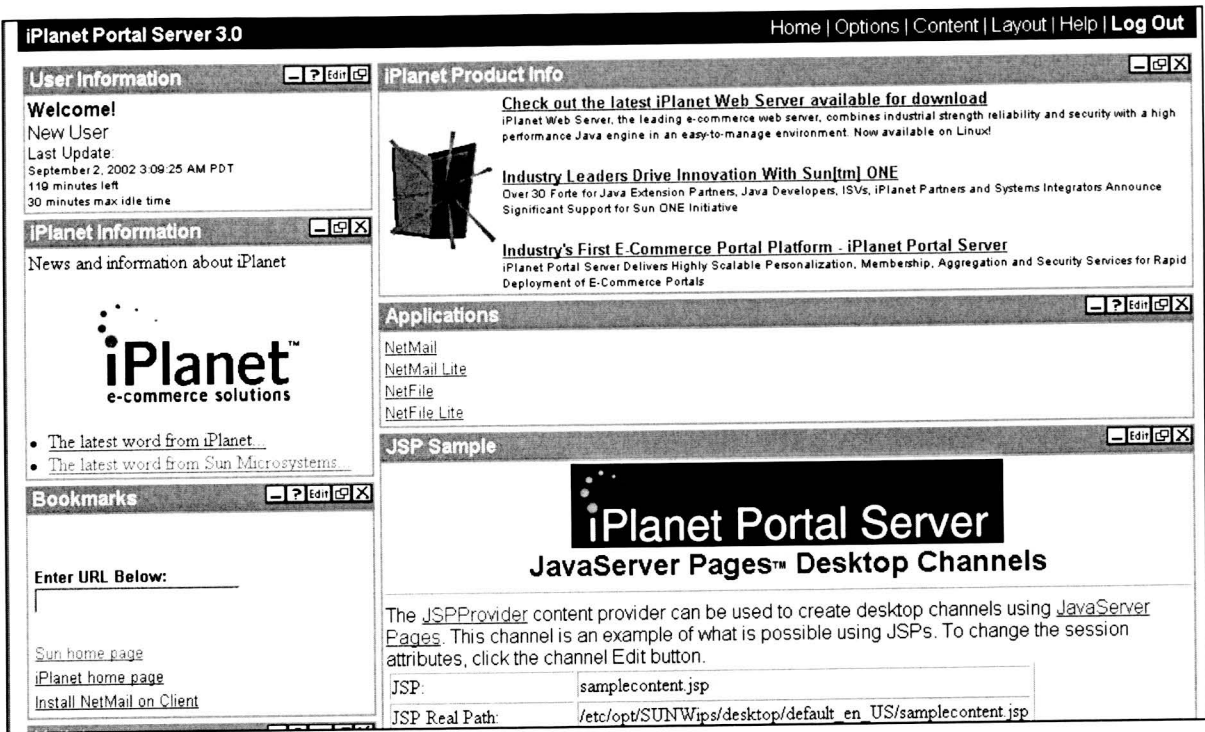


Fig. 3-2 Portal de Serviços - Ecrã típico de um Portal construído com o *iPlanet Portal Server*

A autenticação por LDAP é a ideal, já que este protocolo torna fácil a integração com dados já existentes na empresa, e irá constituir o ponto de migração e centralização de vários sistemas informáticos dispersos. Porém, os mecanismos de autenticação fornecidos para o LDAP, não suficientes, pois validam qualquer utilizador que possua um *login* e palavra chave correctos.

Como referido anteriormente, é imperativo para a empresa garantir que o detentor do *login* *x*, e palavra chave *y*, só consiga aceder ao Portal a partir do computador *z*.

A aplicação que fornece a área de trabalho também se revela algo complexa para os objectivos do projecto, pelo que irá ser adoptada uma solução baseada em JSP's ²⁰ simples.

²⁰ Java Server Pages, analisadas no capítulo 2

4 Desenvolvimento Efectuado

Este capítulo destina-se a documentar os principais desenvolvimentos efectuados durante o projecto. Outras fases mais marginais de desenvolvimento estão documentadas em anexos ao relatório.

4.1 Arquitectura do Projecto Piloto

A arquitectura do projecto piloto visa imitar ao máximo a implementação final da solução. A diferença principal entre esta arquitectura e a arquitectura final é que neste caso todos os componentes se encontram fisicamente na mesma máquina. Adicionalmente, o ambiente final será constituído por máquinas com melhores desempenhos. Neste projecto piloto usou-se uma máquina com as seguintes características:

- Sistema Operativo SunOS²¹ 5.8
- 512 Mb Memória
- Processador UltraSPARC-III

Esta máquina encontra-se fisicamente delimitada por duas *firewalls*, que marcam a zona desmilitarizada (DMZ), como se pode observar no seguinte diagrama:

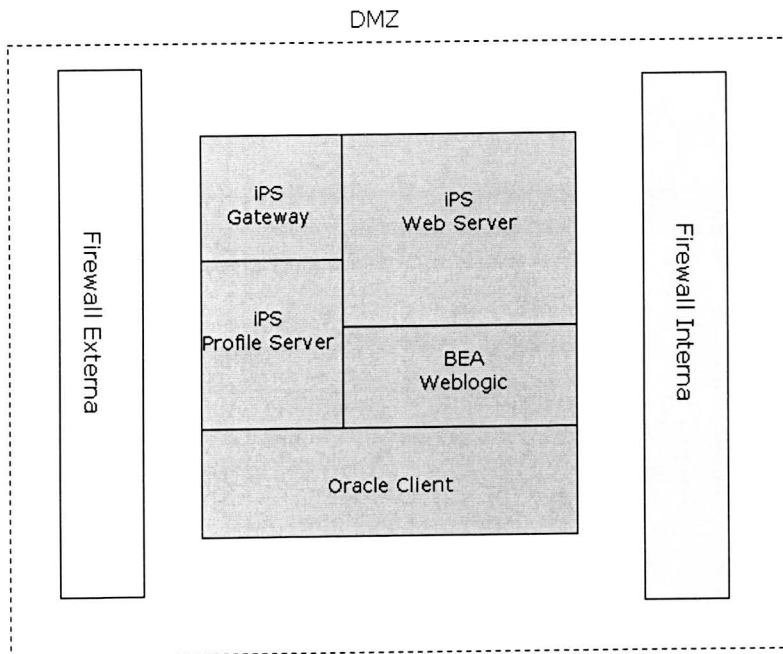


Fig. 4-1 Portal de Serviços - Arquitectura Física do Piloto

A máquina Sun contém vários elementos essenciais para o funcionamento correcto do piloto:

- iPlanet Portal Server 3 Service Pack 4
- Bea WebLogic 5.1

²¹ Sistema Operativo desenvolvido pela Sun Microsystems

- Oracle Client 8.1.7

Os componentes essenciais do *iPlanet Portal Server* são:

- iPS Gateway
- iPS Web Server
- iPS Profile Server (directório LDAP)

A Gateway funciona como ponto de entrada seguro no Portal, aceitando ou rejeitando pedidos consoante exista ou não uma sessão activa do utilizador no portal. É esta gateway que também funciona como intermediária, encarregando-se de reencaminhar os pedidos HTTP para o servidor *Web* ao mesmo tempo que recebe as páginas requisitadas, e as reenvia ao utilizador.

Por sua vez, o *Profile Server* assiste todo este processo, já que é responsável por definir os privilégios de cada utilizador no Portal e suas aplicações, bem como definir as opções que podem ou não ser acedidas. Ao mesmo tempo guarda de uma forma segura e persistente os atributos das aplicações do Portal (tais como posicionamento dos canais na sua área de trabalho, *login's* e palavras chave das aplicações de segundo nível, etc.).

Suportando o Portal, encontra-se o servidor aplicacional *WebLogic* da BEA, responsável por disponibilizar as aplicações de segundo nível a que se queira aceder.

Finalmente, abaixo de toda esta camada, está um cliente Oracle responsável por efectuar ligações às bases de dados que armazenam informação utilizável pelo servidor aplicacional, ou mesmo pelo próprio Portal.

4.2 Configurar o iPlanet Portal Server

Esta secção engloba os principais passos de configuração tomados no iPS para a sua integração com outras aplicações, e para funcionar cumprindo os objectivos propostos.

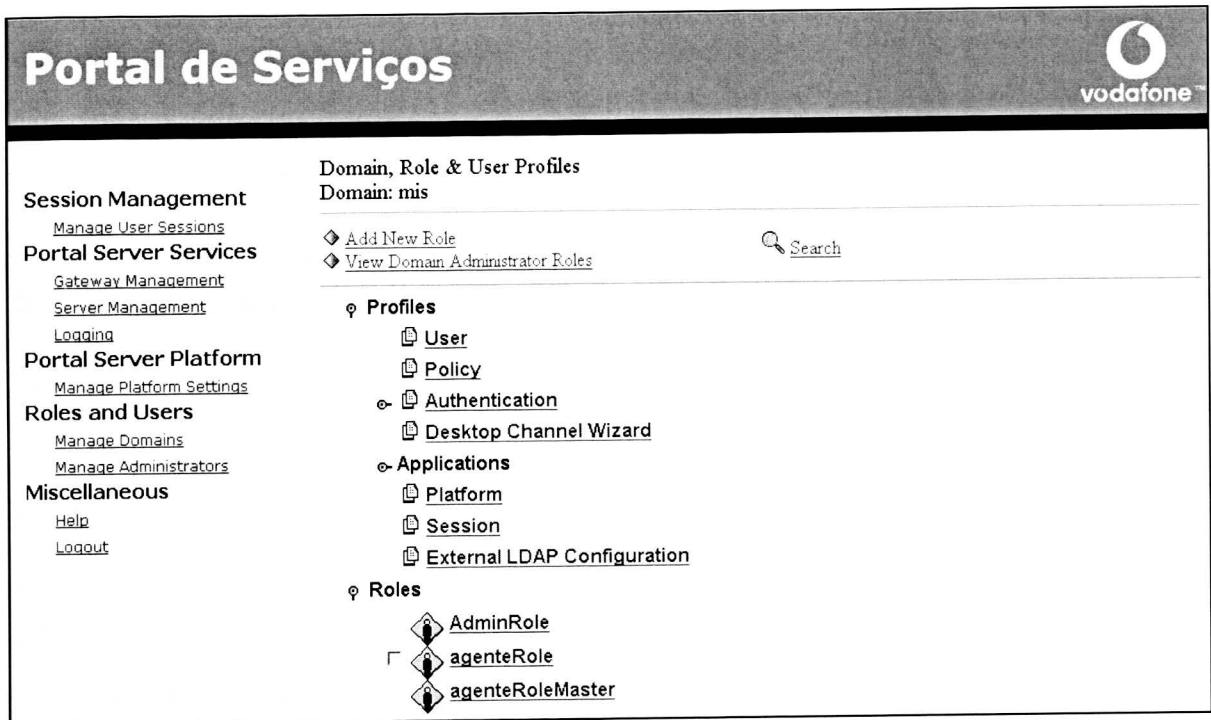
Administração do iPlanet Portal Server

Fig. 4-2 Portal de Serviços - Ecrã de Administração do iPlanet Portal Server

O iPS fornece uma consola de administração *Web* capaz de configurar a maior parte dos aspectos do Portal. Esta consola torna-se útil na implementação dos requisitos propostos no projecto Portal de Serviços.

É através desta consola que se configuram domínios, utilizadores, papéis, permissões de acesso, mecanismos de autenticação, entre muitos outros aspectos que tornam este iPS tão flexível. Esta mesma consola utiliza padrões HTML que podem ser alterados para lhe dar um aspecto mais conforme com a empresa.

A utilização desta ferramenta é essencial e omnipresente durante todo o projecto, para levar a cabo os passos seguintes.

Domínios Disponíveis

O iPS funciona suportando-se em domínios. Cada domínio pode ter um conjunto de configurações próprias para ele, consoante o perfil dos utilizadores que lhe irá aceder. Inclusive, o aspecto das páginas mostradas aos utilizadores em cada domínio, pode ser alterado, mudando unicamente os padrões HTML desse domínio.

O Portal desenvolvido, será utilizado principalmente pelos parceiros Vodafone. Adicionalmente será dado acesso aos utilizadores *Back-Office*. Numa fase posterior poder-se-á alargar a outra gama de utilizadores.

Após discussão, concordou-se em criar um domínio no Portal para estes dois tipos de utilizadores. Os parceiros irão pertencer ao domínio *mis*, e os utilizadores *Back-Office* ao domínio *ipsbackweb*. Para estes utilizadores o único pormenor que necessitam de saber para entrar no seu domínio, é o nome do mesmo. Por exemplo,

- <https://portal.servicos.pt/mis>
- <https://portal.servicos.pt/ipsbackweb>

A criação de um domínio é feita simplesmente através da consola de Administração do iPS. Posteriormente, se for necessário mudar o aspecto das páginas *Web* iniciais do domínio, tem de se alterar os padrões HTML para esse mesmo domínio. Essa acção foi tomada, para tornar o Portal mais coerente no contexto do seu projecto.

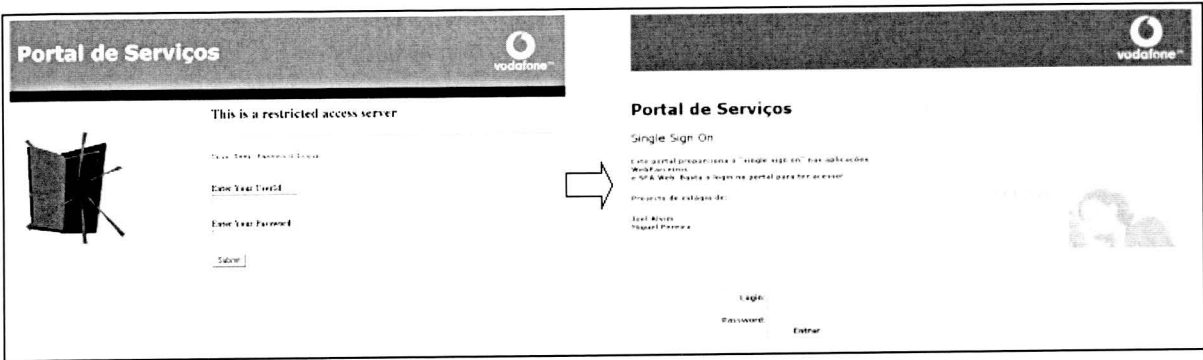


Fig. 4-3 Portal de Serviços - Alteração do Aspecto de Entrada nos Domínios

Criação de Utilizadores e Papéis

Dentro de cada domínio disponível existem um ou mais papéis. Um utilizador do Portal terá então que pertencer a um papel, e consequentemente a um domínio.

Os vários atributos, políticas de acesso e configurações podem ser definidos para um domínio, papel (*Role*) e utilizador. O iPS interpreta sempre os valores definidos a nível mais baixo, e os que não consegue encontrar aí, tenta procurar num nível superior. Daí a informação ser herdada.

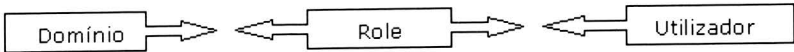


Fig. 4-4 Portal de Serviços Interdependência entre Papéis do iPlanet Portal Server

Por exemplo, é possível restringir o acesso a uma máquina a nível de domínio, de forma a que nenhum dos seus utilizadores lhe possam aceder. Do mesmo modo, é possível retirar o acesso a uma aplicação, a nível de papel, que nenhum utilizador que pertença a esse papel lhe poderá aceder. E finalmente, é possível definir que um utilizador específico não tem acesso também ao seu correio electrónico.

A nível do projecto, foram criados alguns papéis e utilizadores dentro dos domínios definidos na secção anterior.

No domínio *ipsbackweb*, criou-se um papel:

- *backRole*, onde irão estar todos os utilizadores de *Back-Office*.

No domínio *mis*, foram criados 3 papéis:

- *agenteRole* – onde irão estar todos os utilizadores das lojas dos parceiros Vodafone
- *agenteRoleMaster* – irá albergar um utilizador que corresponde ao responsável por determinado parceiro.
- *adminRole* – onde estarão os utilizadores com privilégios de administrador, podendo configurar papéis, políticas de acesso e aplicações, mas apenas no domínio a que pertence este papel.

Configuração da Gateway

Um dos componentes essenciais do iPS é a *Gateway*. Além de constituir o ponto de entrada comum no Portal, é responsável por enviar os pedidos HTTP dos clientes ao servidor *Web*, e recolher as respostas. Nesta fase aparece uma funcionalidade vital: *Rewriting* do conteúdo dos ficheiros HTML que recebe do servidor *Web* e das aplicações de segundo nível.

Rewriting, ou reescrita, consiste em substituir todas as ocorrências e hiper-ligações encontradas no HTML resultante, de modo a que estas apontem para elas próprias, mas precedidas pelo endereço da *Gateway*. Isto para forçar que qualquer pedido que o cliente faça ao Portal, passe forçosamente pela *Gateway*, reforçando a segurança. Se a *Gateway* for o único ponto de entrada aberto para a *Internet* e os pedidos não passarem por ela, as suas respostas não poderão chegar à máquina destino.

Suponhamos que se acede normalmente ao Portal pela *Gateway*, e se iria receber um HTML do servidor aplicacional contendo uma hiper-ligação <http://servidorx.vodafone.pt/aplicacao/>. Para que o cliente envie o pedido pela *Gateway*, a hiper-ligação teria de ser semelhante a <https://gatewayx.vodafone.pt/http://servidorx.vodafone.pt/aplicacao/>.

Assim, é necessário configurar a *Gateway* para procurar em locais do HTML, possíveis ligações a reescrever. Isto pode acontecer dentro de código *Javascript*, argumentos de chamada a *Applets Java*, código *XML* e código *HTML* normal.

A configuração genérica funciona na maior parte dos casos, mas é sempre necessário forçar a *Gateway* para reescrever em locais próprios, como funções *Javascript*²² próprias das aplicações utilizadas. Esta configuração encontra-se completa para as aplicações fornecidas, mas é sempre necessário verificar se novas aplicações não irão precisar de reconfigurar a *Gateway*.

Finalmente, para se poder utilizar as API's *Java* do iPS numa máquina diferente daquela onde corre o servidor *Web* do Portal, é necessário enviar-lhe uma *Cookie* contendo o estado actual da Sessão do utilizador no Portal. A lista de máquinas para as quais a *Gateway* envia essa *Cookie* tem de ser configurada previamente à sua utilização.

Configuração e Preparação do Serviço de Directórios

O serviço de directórios do Portal (também designado por *Profile Server*), não só armazena os perfis de utilizador, como pode guardar outra informação importante.

²² Linguagem de *scripting* que corre no browser da máquina cliente

Para que um utilizador possa entrar no Portal, é necessário armazenar em algum sítio, a lista de todos os utilizadores, respectivos *login*'s e palavras-chave. Criou-se a seguinte estrutura para armazenar estes dados:

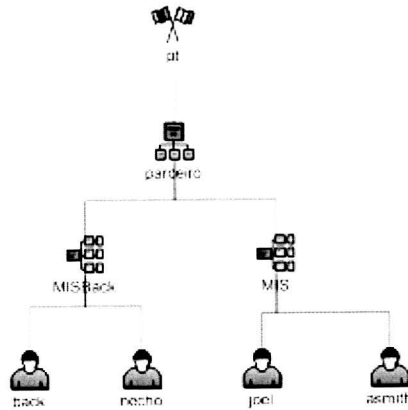


Fig. 4-5 Portal de Serviços - Estrutura LDAP de Utilizadores

Dentro da divisão organizacional MIS, encontram-se todos os utilizadores correspondentes aos agentes da Vodafone. A divisão MISBack agrupa os utilizadores internos da Vodafone, tais como utilizadores de *Back-Office*, 1210 e 1291.

4.3 Desenvolver sobre o iPlanet Portal Server

Esta secção destina-se a ilustrar concretamente as interfaces de programação oferecidas pelo iPS, que serão utilizadas nas próximas secções.

API's Java de Programação

Para melhor se integrar nas aplicações já existentes, o iPS disponibiliza API's de programação em *Java* que permitem a qualquer aplicação escrita nessa linguagem aceder aos serviços principais do Portal. Estas API's foram exploradas neste projecto, mas salienta-se que existe outra forma de comunicar com o Portal, através de código XML embebido em pedidos HTTP.

Profile API

A Profile API permite ler e escrever no Servidor de Perfis. Torna-se possível às aplicações utilizarem informação desse servidor, bem como armazenar dados de forma persistente, constituindo um perfil.

Session API

Esta API fornece meios de obter a Sessão de um dado utilizador no Portal, e verificar se ela é válida, ou não (se expirou, se foi destruída, etc). Permite também ler e adicionar dados à sua Sessão.

Logging API

Através desta API é possível às aplicações registarem acções num repositório comum, que é o mecanismo de *logs* do Portal.

Pluggable Authentication Module API

Como o nome indica, esta API permite desenvolver módulos de autenticação próprios que funcionam dentro do esquema do Portal. Estes módulos são “encaixados” como extensões, sendo imediatamente utilizáveis.

4.4 Módulo de Autenticação – LdapMis

Para que um utilizador do Portal seja validado à entrada pelo seu *login*, palavra-chave e PC físico do qual acede, foi necessário criar um novo módulo de autenticação.

Graças à *Pluggable Authentication API*, esta tarefa de integração e desenvolvimento tornou-se mais fácil porque faz uso de normas e classes *Java* bem definidas e testadas. Adicionalmente, foi necessário recorrer à API de programação *Java* para LDAP, da Netscape.

Arquitectura do Módulo de Autenticação

Este módulo de autenticação apoia-se directamente no componente LDAP já existente no Portal. Este armazena no servidor de perfis todos os dados necessários para uma ligação correcta a um servidor LDAP já existente, de modo efectuar uma pesquisa posterior para validar os dados de autenticação do cliente.

Seguidamente apresenta-se um esquema que ilustra o fluxograma deste módulo concreto.

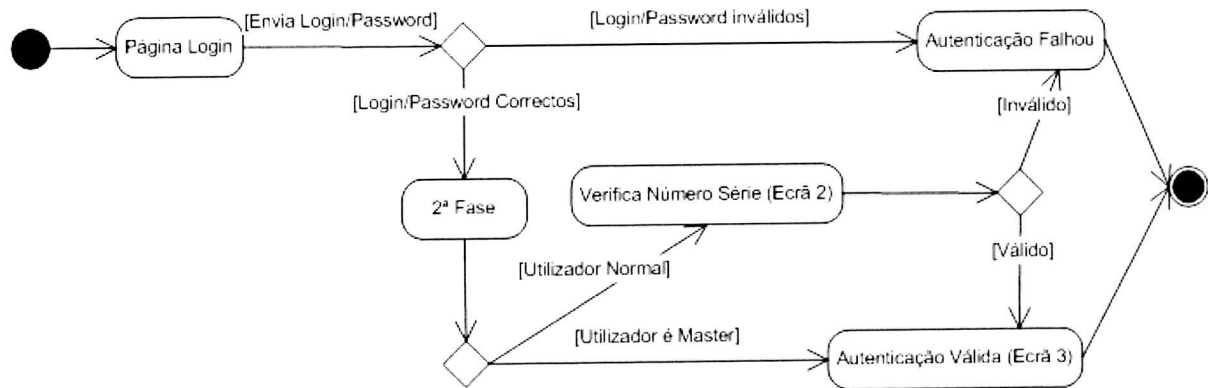


Fig. 4-6 Módulo de Autenticação - Diagrama de Estados

Tal como se pode ver neste diagrama, o primeiro passo do processo de autenticação consiste no envio do *login* e palavra-chave do utilizador ao módulo de autenticação. Se estes valores estiverem errados, o processo acaba imediatamente e a autenticação é considerada incorrecta. Se estiverem correctos, é necessário (ou não), validar também o endereço físico da máquina que está a aceder. Se esse endereço físico constar na perfilagem do Portal, como um endereço correcto para o utilizador que está a aceder, a autenticação é bem sucedida.

Existe um caso especial em que o módulo de autenticação sabe que não necessita de validar o endereço físico do PC. Nesta fase do projecto, esse caso pertence ao utilizador *Master*. Isto, porque este utilizador é responsável por registar nas suas lojas os PC's que terão acesso ao Portal. Para o fazer, ele próprio necessita de ter acesso ao Portal com a máquina que ainda não se encontra registada no mesmo.

Foram discutidas várias hipóteses em reunião na Vodafone, e chegou-se à conclusão que a melhor maneira de atingir este objectivo seria confiar nos utilizadores *Master* e dar-lhes um acesso apenas baseado em *login* e palavra-chave.

Note-se que no diagrama existe a página de *login* inicial, bem como 2 ecrãs adicionais. Estes ecrãs correspondem a páginas HTML. Estas páginas têm funcionalidades específicas, que serão referidas posteriormente. O seguinte diagrama ilustra a sequência de processos que ocorre numa autenticação bem sucedida, de um utilizador normal:

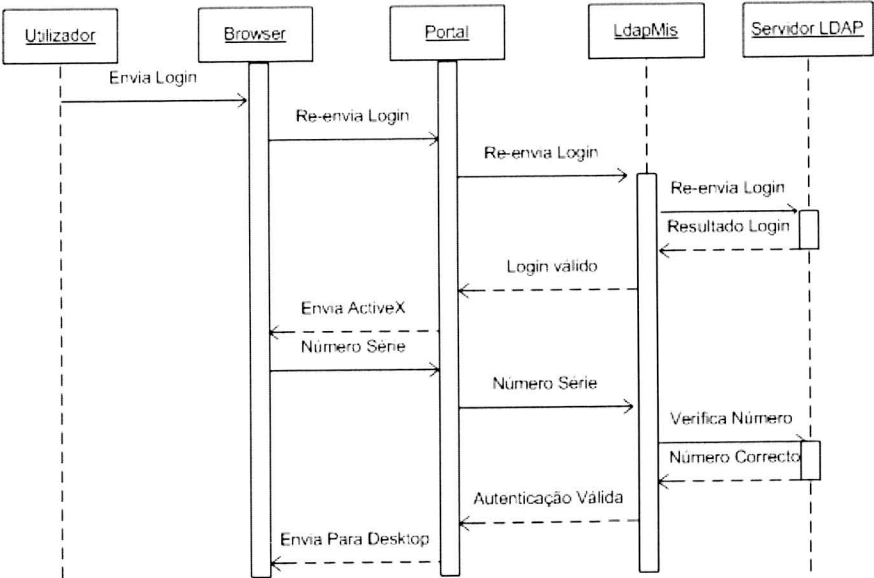


Fig. 4-7 Módulo de Autenticação - Diagrama de Sequência

Existem alguns componentes importantes neste diagrama. O LdapMis actua como um autenticador completo, que recebe informação de fora, e a valida no Servidor Ldap. Para isso ele usa componentes da perfilagem do Portal, nomeadamente o *iwtLdap* já existente e o *iwtUserMac* desenvolvido no âmbito deste projecto, descrito a seguir.

Com o componente *iwtLdap* é possível efectuar a ligação correcta ao servidor Ldap, validando aí o *login* e palavra-chave do utilizador. Por outro lado, o componente *iwtUserMac* possibilita a validação do número de série do PC. Este número é um identificador físico da máquina que acede (um endereço MAC, ou mesmo o número de série de volume C:). Ele é lido e enviado por um controlo ActiveX, que é descarregado para o *browser* pelo Portal.

Implementação do Módulo de Autenticação

iwtUserMac – Um Novo Componente para o iPlanet Portal Server

Tal como referido anteriormente, é necessário validar além do *login* e palavra-chave de um utilizador, o PC a partir do qual ele acede.

Foi efectuada uma investigação por outro membro do projecto, chegando-se à conclusão que a maneira melhor de o fazer, é através da leitura do endereço MAC do PC cliente.

No entanto, um PC só possui um endereço MAC se tiver pelo menos uma placa de rede. No caso da ausência desta, optou-se pela leitura do número de série do volume C:, que apesar de não ser tão inequívoco como a leitura do MAC, constitui um bom compromisso.

Em qualquer dos casos é necessário armazenar no servidor de perfis do iPS, quais os números de série que têm acesso no Portal.

Para isso foi necessário construir um componente, que define simplesmente uma lista configurável contendo aqueles números. Esta lista é configurável, como nos outros casos, a nível de domínio, papel ou utilizador.

User Attributes

Attribute

MAC Values list

0002a5ddbdf2

0002a5de7cf8

0000397c852d

Add

Delete

Status

Customized

Master User holding Mac Values

Inherited

Fig. 4-8 Módulo de Autenticação - Administração do Componente iwtUserMac

O Módulo de Autenticação

Para desenvolver este módulo foi necessário recorrer a API's de programação existentes, bem como construir código próprio.

O esquema de componentes utilizado é o seguinte:

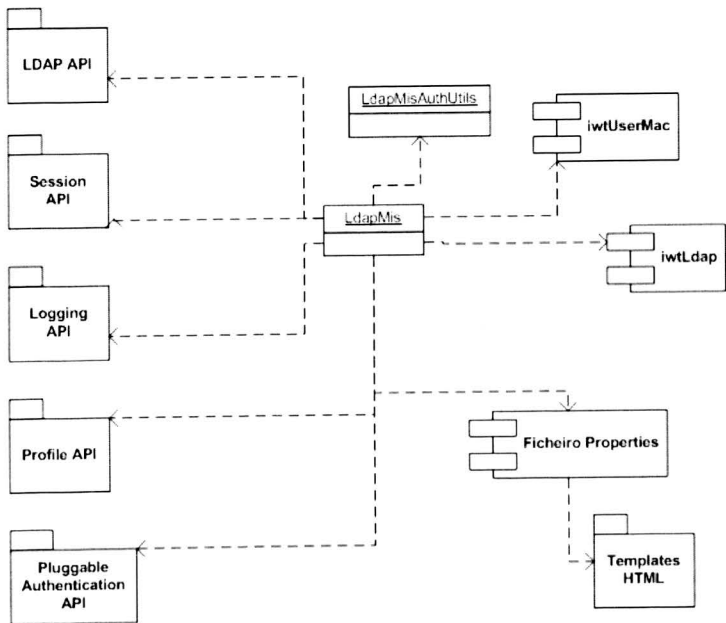


Fig. 4-9 Módulo de Autenticação - Diagrama de Componentes

Como se observa no diagrama, o módulo de autenticação utiliza as API's descritas para efectuar validações internas, registos no próprio portal, bem como consultas aos elementos de perfilagem *iwtLdap* e *iwtUserMac*.

Adicionalmente, existe uma classe denominada *MisAuthUtils* que contém uma lista dos papéis (*roles*) do Portal, que não necessitam de validar o seu número de série. O diagrama de classes resultante é então:

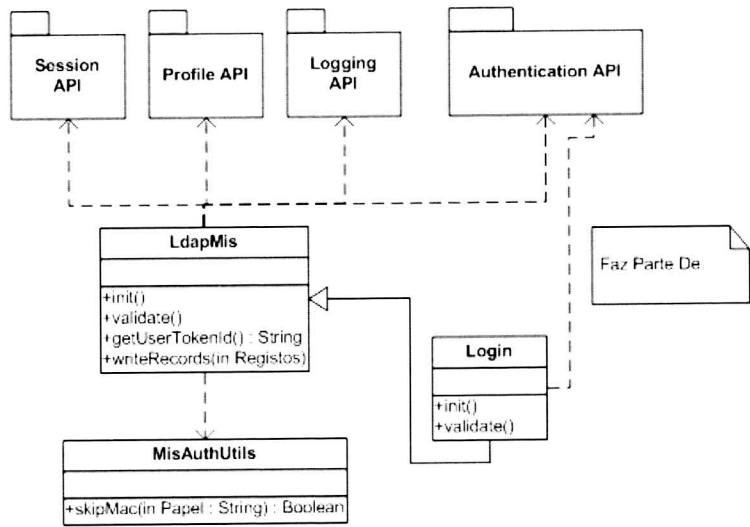


Fig. 4-10 Módulo de Autenticação - Diagrama de Classes

A autenticação do Portal apoia-se na utilização de um processo autenticador. É este processo que permite adicionar novos módulos, contanto que se sigam normas pré-definidas e especificadas nos manuais do iPS. Este autenticador necessita pelo menos de uma classe Java, que estenda a classe *Login*, e de um ficheiro *properties*.

O ficheiro *properties* é vital na construção de qualquer módulo de autenticação para o Portal. Ele define a sequência de ecrãs HTML pelos quais o utilizador terá de passar para concluir a sua validação. O autenticador do Portal irá por sua vez chamar o método *validate()* (estendido na classe *LdapMis*), que irá processar toda esta sequência. Para este módulo de autenticação, foi utilizada uma sequência de 3 ecrãs.

```
SCREEN
HTML login_ldapmis.html
TIMEOUT 120
TEXT LDAP Authentication
TOKEN Enter UserId
PASSWORD Enter Password
```

```
SCREEN
HTML login_mac.html
TIMEOUT 240
TEXT Validating MAC Address
TOKEN Mac Address
```

```
SCREEN
HTML login_out.html
TIMEOUT 10
TEXT Login Out
TOKEN Value
```

O primeiro ecrã define um HTML que pede o *login* e palavra-chave do utilizador. Se o módulo validar este passo, pode passar para o ecrã 2, ou 3. O ecrã 3 existe como ponto de saída do autenticador, tal como se pode observar no diagrama de sequência anterior.

O ecrã 2 contém código HTML que descarrega do Portal um controlo ActiveX, responsável por ler o número de série do cliente. Através de código Javascript, é possível passar esse valor para o *browser*, e preencher um campo escondido num formulário HTML. O mesmo código Javascript submete automaticamente esse formulário para o Portal, comunicando assim o valor lido.

O ecrã 3 simplesmente comunica através de um formulário HTML, um valor pré-definido que o módulo de autenticação espera. Constitui deste modo um simples ponto de saída da validação.

Activação de Máquinas

Tal como se observa na figura 4-6, o utilizador *Master* não necessita de validar o seu PC para obter acesso ao Portal. Isto ocorre pelo facto de ser este mesmo utilizador o responsável por activar as máquinas de onde acede, para serem posteriormente utilizadas pelos seus subordinados.

A activação de uma máquina consiste em colocar na perfilagem do Portal o número de série correspondente à mesma, permitindo que os utilizadores futuros que acessem a partir dela possam ser validados correctamente pelo módulo de autenticação.

O componente que irá ler este número de série é o mesmo que foi referido anteriormente. A única diferença, é que em vez de ser descarregado como um ActiveX, estará embebido numa *Applet Java*²³.

²³ Programa em *Java* tipicamente descarregado de um sítio *Web*, que corre embebido no próprio *browser*.

O utilizador *Master* ao entrar no Portal, possui uma hiper-ligação na sua área de trabalho que permite executar esta activação. A sequência de eventos que ocorrem é:

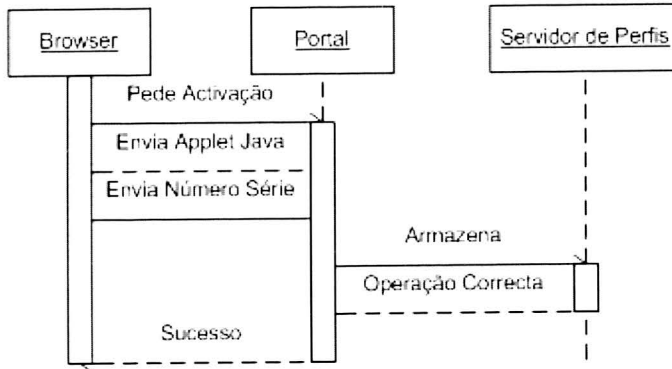


Fig. 4-11 Diagrama de Sequência da Activação de Máquina

Se por algum motivo esta activação falhar, é apresentado ao utilizador uma mensagem de erro apropriada à situação.

Os motivos pelos quais isso poderia acontecer são:

- *Java Machine* do browser não consegue correr a *Applet*
- Latência de comunicação HTTP demasiado elevada
- Erros no acesso ou alteração do Servidor de Perfis

4.5 Núcleo Operativo do Portal

Arquitectura do Núcleo

Esta secção destina-se a documentar os processos e fluxos que ocorrem desde que um utilizador é validado, até que escolhe aceder a uma aplicação disponibilizada, passando pela construção da sua área de trabalho.

Os principais estados definidos são os seguintes:

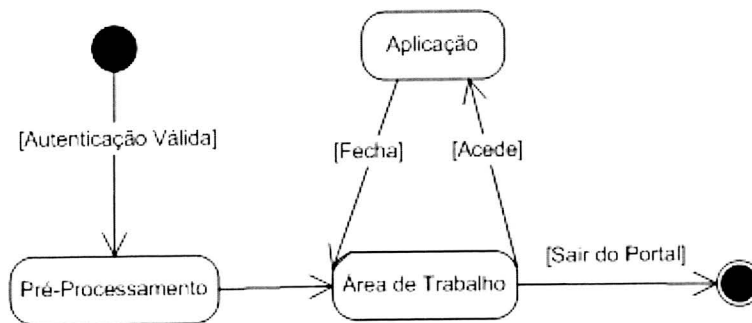


Fig. 4-12 Núcleo Operativo - Diagrama de Estados

Uma vez validado um utilizador, e antes de entrar na sua área de trabalho, passa por uma fase de pré-processamento. Este processamento é efectuado por uma *servlet*. Nesta fase do projecto, os passos que aí ocorrem consistem em:

- Registrar no Portal o papel (*role*) a que pertence o utilizador que entrou.

- Redireccionar esse utilizador para a sua área de trabalho.

A área de trabalho do utilizador é construída através de uma *JSP*, que exhibe as aplicações às quais o utilizador pode aceder. Este acesso encontra-se perfilado, e é definido na administração do Portal, tal como referido na secção 4.6. Além do acesso às suas aplicações, o utilizador tem também a possibilidade de efectuar *logout* do Portal.

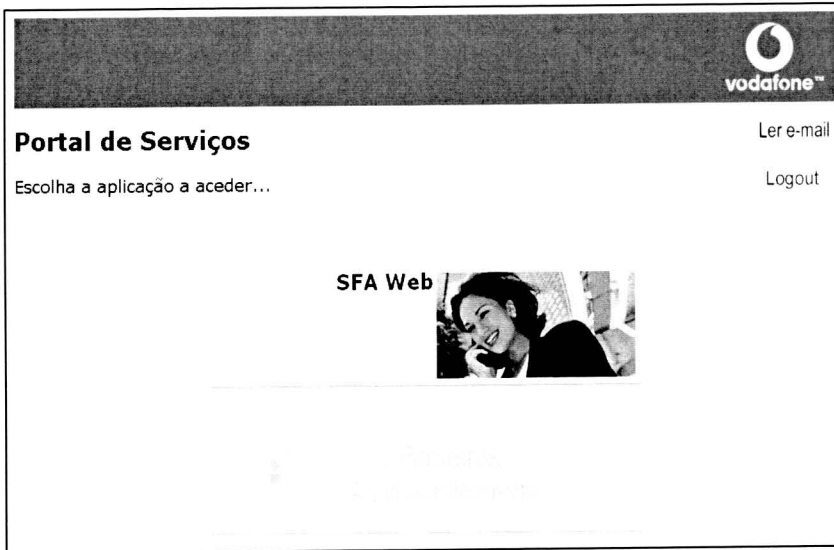


Fig. 4-13 Área de Trabalho do Portal de Serviços

Caso se trate de um utilizador *Master*, ele pode ainda fornecer acesso no Portal ao PC do qual acedeu. O acesso às aplicações é efectuado através de *servlets* específicas (ou adaptadores) que auxiliam o processo de *Single Sign-On*.

No seguinte diagrama pode observar-se melhor os componentes intervenientes:

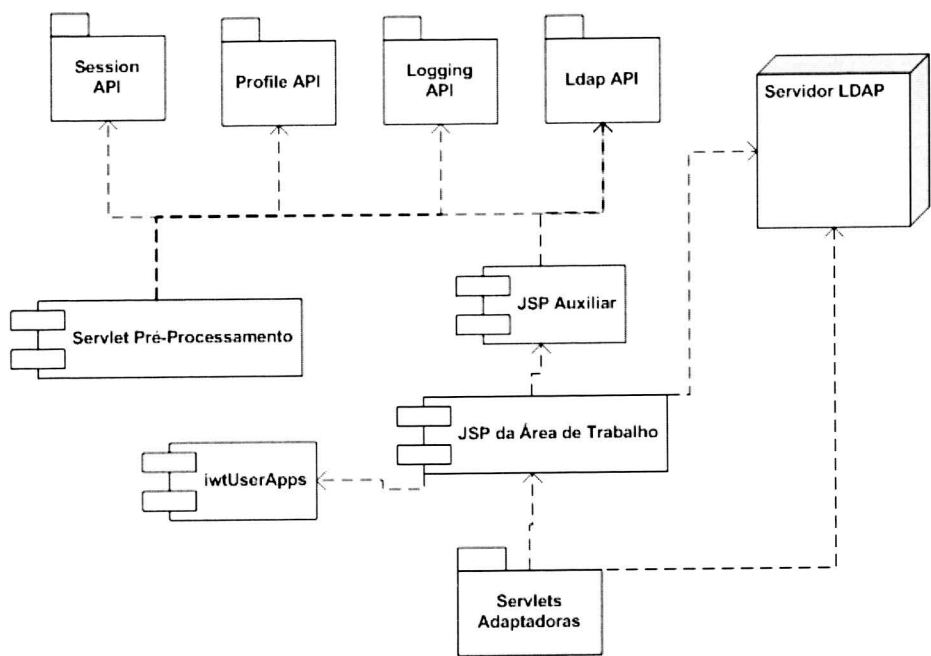


Fig. 4-14 Núcleo Operativo - Diagrama de Componentes

A utilização das API's de Programação do Portal é transversal ao longo de todo este núcleo operativo. O componente JSP Auxiliar inicializa certos objectos Java do Portal que serão utilizados para validar sessões, consultar perfilagem, e registar informação. O componente iwtUserApps guarda no perfil do utilizador actual as aplicações a que este tem acesso (os seus nomes, apenas).

Posteriormente, é a JSP da Área de Trabalho que tem de consultar o directório LDAP e verificar que elas se encontram realmente bem configuradas (possuem dados sobre o servidor onde corre, porto de escuta e URI²⁴ para Single Sign-On). Consoante estejam ou não, a JSP exhibe uma imagem com uma hiper-ligação que aponta para a *servlet* adaptadora que se encarregará de efectuar o *Single Sign-On*.

²⁴ *Uniform Resource Identifiers*, neste caso um apontador para um recurso na *Web*.

Vejamos a sequência de acontecimentos de um acesso bem sucedido a uma aplicação:

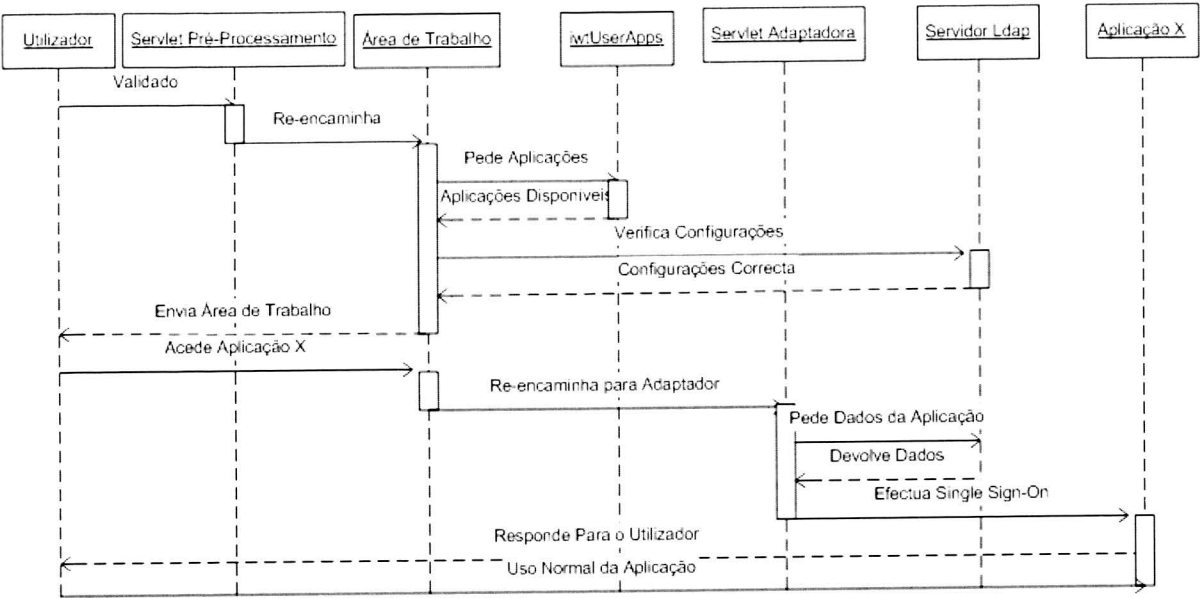


Fig. 4-15 Núcleo Operativo - Diagrama de Sequência

Implementação do Núcleo

Do ponto de vista do código desenvolvido, o núcleo operativo tenta ser o mais transparente possível, fornecendo ao exterior (páginas JSP e Servlets) apenas os métodos de programação essenciais para realizar os passos enunciados anteriormente.

Vejamos então o seu diagrama de classes:

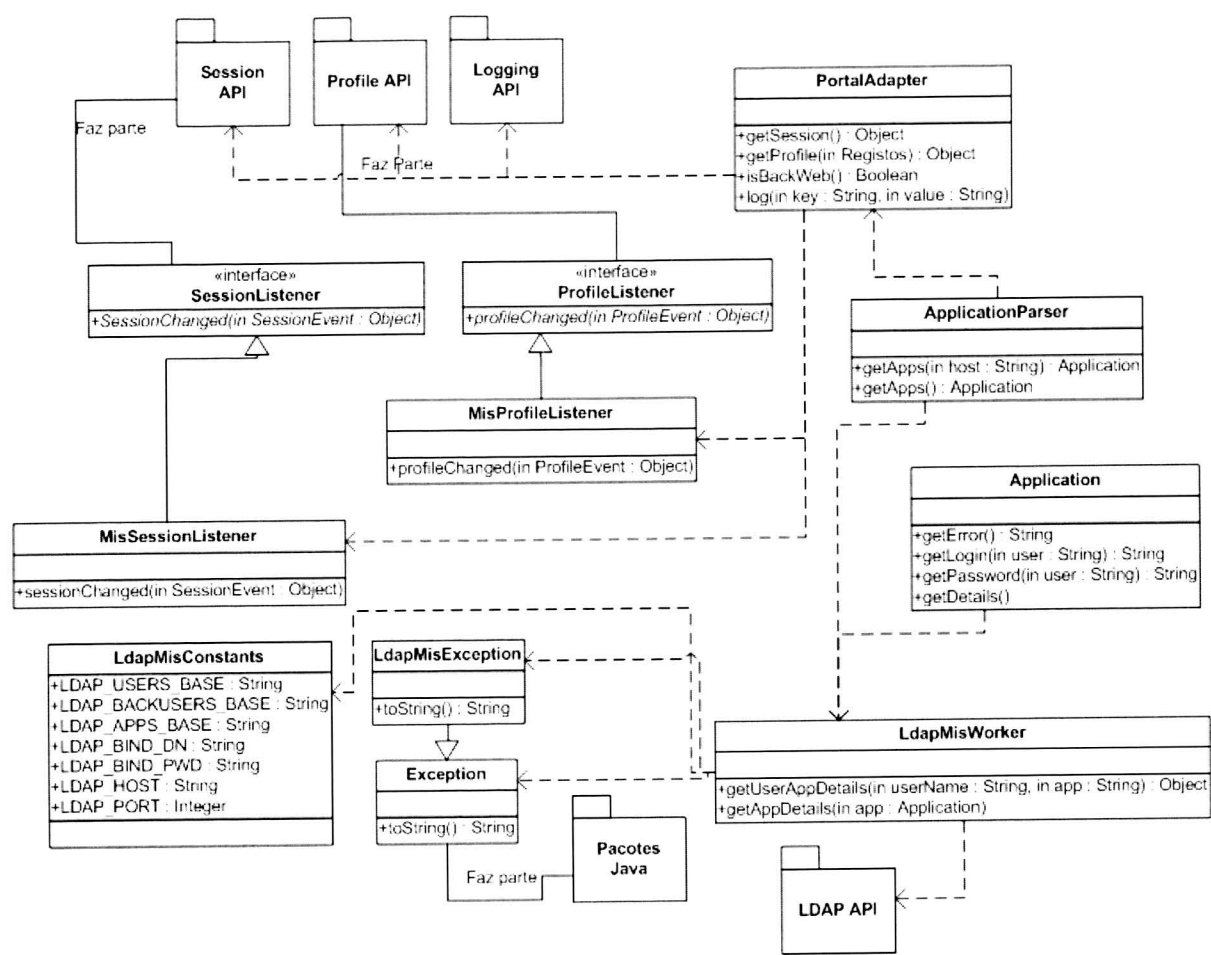


Fig. 4-16 Núcleo Operativo - Diagrama de Classes

Este diagrama constitui o núcleo funcional do Portal, sendo usado posteriormente pela JSP que desenha a área de trabalho e pelas *servlets* adaptadoras de *Single Sign-On*.

Sumariamente, a classe *PortalAdapter* inicializa e instancia objectos importantes do Portal, encapsulando métodos para os utilizar. Fornece meios para registar acções e para obter os objectos de Sessão e Perfilagem, vitais para o funcionamento base do piloto (permitem gerir a sessão actual do utilizador, bem como consultar e escrever no seu perfil). Esta classe cria ainda dois objectos *MisSessionListener* e *MisProfileListener*. Estes permitem detectar mudanças no estado da Sessão e Perfil do utilizador, respondendo a essas mudanças de maneira adequada. Todas as outras classes dependem deste *PortalAdapter* para operarem correctamente.

A classe *ApplicationParser* possui métodos para construir uma *Hashtable* de aplicações disponíveis para o utilizador. Ela lê no seu perfil as aplicações para as quais tem permissão de acesso e, seguidamente, consulta no directório LDAP os detalhes de cada aplicação, encapsulando esses detalhes num objecto *Application*. Este objecto possui métodos para recuperar esses detalhes, bem como para consultar no directório LDAP, o *login* e palavra-chave de um utilizador, na aplicação de 2º nível que o objecto representa.

Abaixo desta camada, existem classes que se encarregam dos aspectos de comunicação com o servidor LDAP.

A classe *LdapMisWorker* possui métodos internos para efectuar ligações ao servidor e gerir excepções. Os seus métodos mais importantes permitem:

- Preencher um objecto *Application* com os detalhes respectivos
- Devolver uma *Hashtable* com o *login* e a palavra-chave de um dado utilizador numa dada aplicação de 2º nível.

Esta classe apoia-se nas suas auxiliares, *LdapMisConstants* e *LdapMisException*. A classe de constantes *LdapMisConstants* possui dados importantes para a ligação ao servidor LDAP, tais como máquina onde corre, porto de escuta, ramo do directório onde procurar as aplicações e utilizadores. A classe *LdapMisException* encapsula todas as excepções resultantes destas pesquisas no servidor LDAP, bem como casos excepcionais próprios do projecto. Destes casos salienta-se:

- Aplicação definida na perfilagem do Portal, mas não presente no directório LDAP
- Aplicação configurada para um utilizador, que não tem configurado o seu *login* e palavra-chave nessa aplicação

4.6 Single Sign-On

Esta secção documenta esforço de desenvolvimento especialmente focado no *Single Sign-On* entre aplicações no Portal.

Alterações ao Directório LDAP

Para efeitos de *Single Sign-On*, é necessário guardar no directório LDAP informação específica de cada aplicação. Essa informação inclui o nome da aplicação, máquina onde corre, porto onde escuta, URL²⁵ de entrada e URL de destino. Entende-se por URL de entrada, o endereço da aplicação que valida os elementos de autenticação. URL de destino consiste no endereço para o qual o utilizador será redireccionado após ter entrado na aplicação.

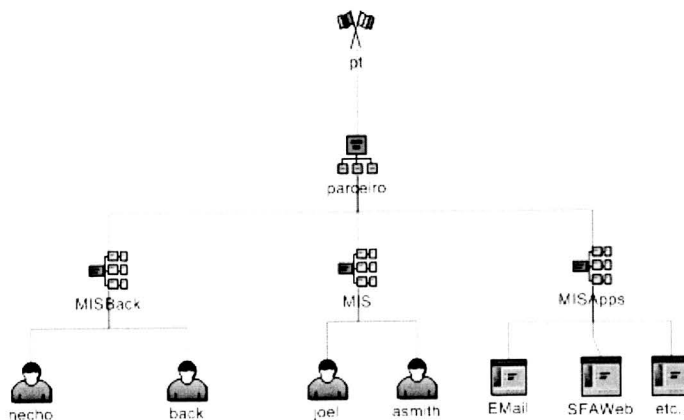


Fig. 4-17 Single Sign-On - Alteração ao LDAP

²⁵ *Uniform Resource Locator*. Especifica um recurso que neste caso está acessível pela Internet.

A divisão MISApps armazena entradas contendo esta informação para cada aplicação de segundo nível disponível.

Adicionalmente, cada utilizador terá de armazenar o seu *login* e palavra-chave nas aplicações de segundo nível a que pode aceder.

Todas estas informações adicionais utilizam atributos não definidos por omissão no esquema (*schema*) das classes de um protocolo LDAP. Por isso é necessário adicionar estes novos atributos a esse esquema. Nesta fase do projecto, estas alterações são efectuadas manualmente pela consola de administração do servidor LDAP. No entanto, torna-se um requisito futuro tornar estas alterações automáticas, ou mais específicas e utilizáveis por um administrador interno na Vodafone.

Foi ainda feito algum esforço neste sentido, e criados os seguintes utilitários que correm na consola do sistema operativo:

- LdapAppsSchema: permite criar uma nova classe no esquema do directório LDAP especificado, que irá suportar os dados necessários para as aplicações de segundo nível.
- LdapManageApps: permite listar, adicionar ou apagar, objectos Aplicação do directório LDAP.
- LdapUserAppsManager: permite adicionar a um dado utilizador, campos contendo o *login* e palavra-chave de uma dada aplicação de 2º nível.

iwtUserApps - Um Novo Componente do iPlanet Portal Server

Devido à natureza expansível do iPS, é fácil adicionar novos módulos à perfilagem do Portal. Sabendo quais as aplicações que se querem disponibilizar, é necessário configurar a nível de domínio as aplicações a que determinados utilizadores ou papéis terão acesso.

Um componente deste género é simples de criar, utilizando um ficheiro XML que descreve o componente e os seus atributos (um atributo pode ser um número, um texto, uma lista de campos texto, caixas de escolha múltipla, etc). A título do exemplo, o XML necessário para este componente é:

```
<iwt:Component name="iwtUserApps"
  ver="1.0"
  desc="Default User Access to External Apps"
  resB="iwtUserApps"
  idx="">
<iwt:Att name="iwtUserApps-available"
  desc="Available Applications"
  type="multichoice"
  idx="u1"
  userConfigurable="TRUE">
  <Rperm>ADMIN</Rperm><Rperm>OWNER</Rperm>
  <Wperm>ADMIN</Wperm>
  <CVal>WebParceiros</CVal>
  <CVal>SFAWeb</CVal>
  <CVal>FeedbackReports</CVal>
  <CVal>Email</CVal>
</iwt:Att>
```

Adicionalmente cria-se um ficheiro denominado *resource bundle* contendo o texto para cada atributo que será exibido na administração do Portal. Isto permite desenvolver componentes, suportando localização geográfica.

Finalmente, é necessário importar para o Portal este novo componente. Isso é feito facilmente através de um utilitário denominado *ipsadmin*, que vem com o produto.

Uma vez criado o componente, a sua utilização efectua-se do modo normal na consola de administração *Web* do iPS.

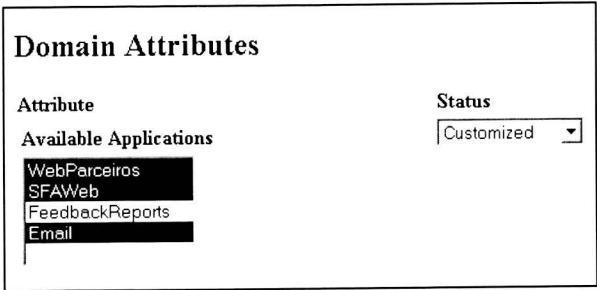


Fig. 4-18 Single Sign-On - Administração de Aplicações

Servlets Adaptadoras

Para implementar o *Single Sign-On*, foi necessário recorrer a *servlets* que funcionam como adaptadores. Todos estes adaptadores funcionam de um modo genérico, mas cada um deles executa código próprio para a aplicação da qual está encarregue. O fluxo deste processo pode ser visto na figura 4-3.

Veamos então o diagrama de classes correspondente:

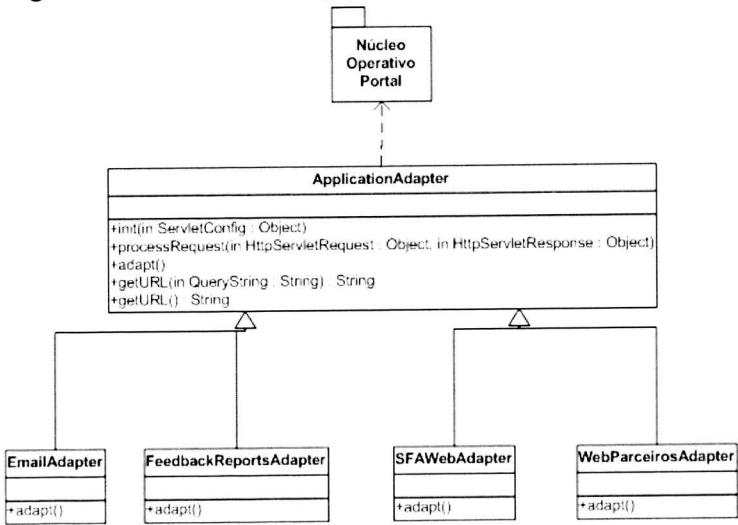


Fig. 4-19 Single Sign-On - Diagrama de Classes

A classe *ApplicationAdapter* define métodos próprios para auxiliar o funcionamento das restantes classes que a estendem. Estas, têm simplesmente de reprogramar o método *adapt()* para efectuar os procedimentos próprios para efectuar *Single Sign-On* na aplicação respectiva.

Mecanismos de Single Sign-On

Efectuar *Single Sign-On* numa aplicação constitui um processo que pode ser bastante generalizável, ou bastante específico. Tudo depende do grau de liberdade que se têm em alterar o código da aplicação de 2º nível.

No âmbito deste projecto, cada uma das aplicações disponibilizadas representa um patamar desta generalização:

- SFAWeb : Não se pode alterar ou tocar no código original
- WebParceiros : Pode-se alterar em parte o código, sujeito a restrições do grupo de trabalho do projecto original
- FeedbackReports : Código completamente livre, já que foi desenvolvido por esta equipa especificamente para este projecto
- E-Mail : Aplicação já por si integrada no Portal, não necessita de ser mexida

Vejamos agora para cada uma das aplicações, os passos tomados para efectuar *Single Sign-On* nas mesmas.

SFAWeb

Quando foi proposta a solução de *Single Sign-On* nesta aplicação, automaticamente colocou-se a restrição de que não se poderia mexer no código original. Constituiria como que uma caixa-negra, para a qual teria de ser arranjada uma ponte de ligação com o Portal.

Um utilizador típico introduz o seu *login* e palavra-chave no formulário de autenticação, e caso estes dados estejam correctos, ele é autenticado na mesma. Para efeitos de *Single Sign-On*, estes passos teriam de ser de algum modo simulados.

A solução actual que se encontrou, consiste na existência de uma página HTML intermédia entre o Portal e a aplicação, que possui um formulário com dois valores chave escondidos: o *login* e palavra-chave do utilizador respectivo. Este formulário é construído pelo adaptador desta aplicação, e submetido através de Javascript para o ponto de entrada nessa aplicação, simulando uma autenticação típica.

WebParceiros

No caso da aplicação WebParceiros, foi possível alterar parte do código da mesma que também foi desenvolvida em Java.

Vejamos como funcionava originalmente a autenticação da aplicação:

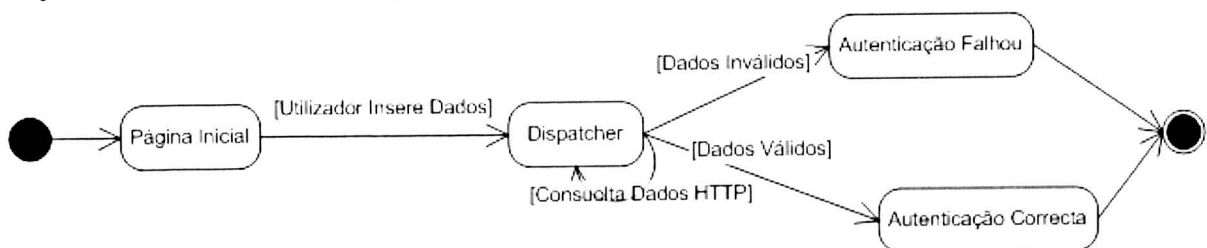


Fig. 4-20 Single Sign On - Diagrama de Estados Antigo da Aplicação WebParceiros

Após o utilizador inserir o seu *login* e palavra-chave na aplicação, estes dados são passados (juntamente com outros escondidos no formulário de entrada) para uma *servlet* que os valida conjuntamente, e constrói toda a camada de apresentação desse utilizador. Nesta fase todos esses dados iniciais eram consultados no pedido HTTP que chegava à *servlet*.

Contudo, no contexto do Portal, esta *servlet Dispatcher*, terá de verificar qual o utilizador do Portal que lhe acedeu, e consultar no directório LDAP, qual o *login* e palavra-chave desse utilizador, para esta aplicação de segundo nível. Tem-se portanto:

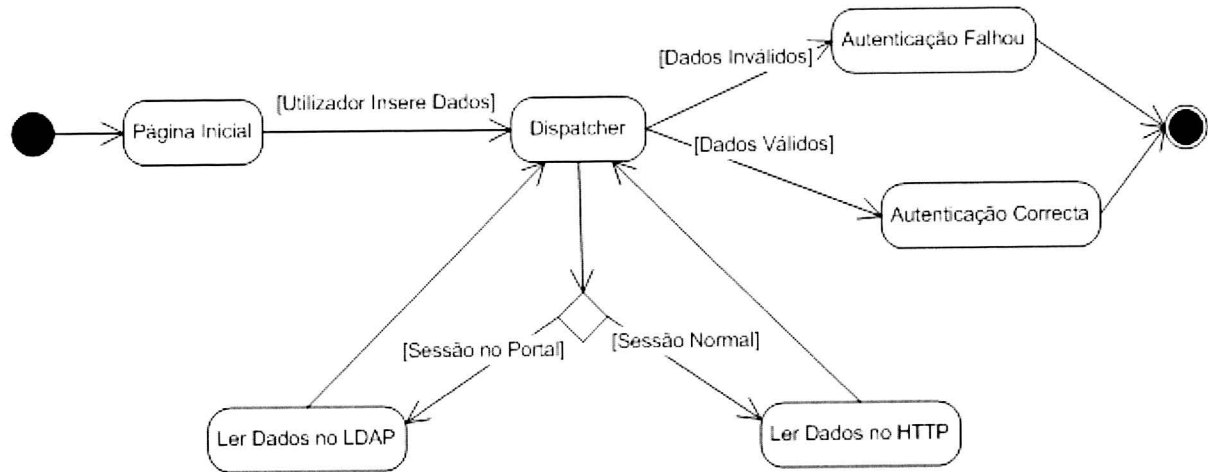


Fig. 4-21 Single Sign On - Novo Diagrama de Estados da Aplicação WebParceiros

Finalmente, foi necessário alterar a *Jsp* de saída da aplicação, para redireccionar o utilizador de volta para a sua área de trabalho no Portal, e não para ecrã de *login* da aplicação, como acontecia inicialmente. Esta alteração consistiu simplesmente em mudar o URL que se encontrava configurado.

FeedbackReports

Esta aplicação desenvolvida de raiz especificamente para o Portal já faz uso das API's de programação fornecidos pelo mesmo. Como tal, os fluxos são relativamente simples:

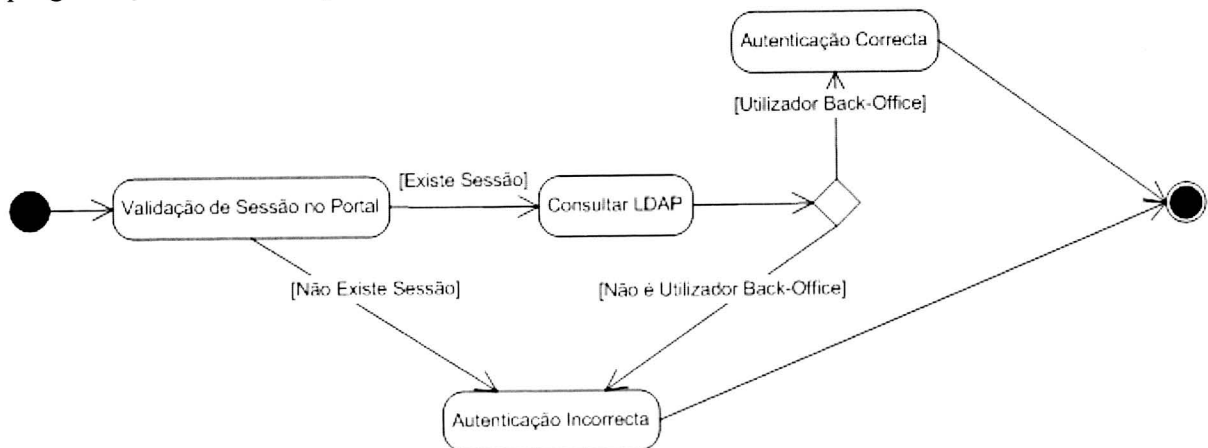


Fig. 4-22 Single Sign On - Diagrama de Estados da Aplicação FeedbackReports

Como se evidencia no diagrama, esta aplicação destina-se apenas aos utilizadores de *Back-Office* do Portal, e funciona apenas dentro do mesmo.

E-Mail

A aplicação de E-Mail já vem na distribuição do iPS. Consiste num serviço *Web* ao qual se acede directamente por uma hiper-ligação HTML. Tal como a aplicação FeedbackReports, o E-Mail só funciona dentro do próprio portal. Efectuar *Single Sign-On* consistiu simplesmente em criar um redireccionamento HTML para esta aplicação.

4.7 Integração do Portal com Aplicações Existentes

Replicação de Utilizadores das Aplicações para o Directório LDAP

Tal como enunciado nos objectivos do projecto, o Portal de Serviços seria um passo na migração dos vários sistemas de armazenamento de dados dispersos, para um sistema centralizado.

A solução que aparenta ser mais robusta e eficaz, aparece na forma de um directório LDAP, que é já por si um componente base do iPS. No entanto, todas as aplicações já existentes apoiam-se nos restantes sistemas disponíveis, e como tal seria impraticável mudá-los radicalmente para fazer uso do LDAP.

É à luz deste problema que surge esta fase de replicação. Numa primeira fase esta replicação diz respeito às aplicações já mencionadas: WebParceiros e SFAWeb. Estas 2 aplicações constituem serviços dominantes prestados pela Vodafone. Enquanto este projecto não for tomado em toda a escala, apenas se podem preparar as fundações para que isso aconteça:

- Migrar todos os utilizadores já existentes para o directório LDAP.
- Criar mecanismos de replicar para o Portal os novos utilizadores das aplicações WebParceiros e SFAWeb.

A aplicação dominante ficou decidida como sendo o WebParceiros. Deste modo, uma criação ou modificação de um utilizador nesta aplicação será forçosamente sucedida da criação deste mesmo utilizador no Portal de Serviços (desde que esse utilizador não exista ainda).

Deste modo quando for necessário adoptar definitivamente a solução do Portal, este já contém toda a informação necessária para funcionar correctamente.

É importante implementar mecanismos para prevenir inconsistências nesta replicação, tais como uma fila de espera para transacções com recolha de erros em caso de falha.

O desenho geral da solução que permite preencher esse directório LDAP será apresentado a seguir.

Arquitectura

Podem-se considerar neste momento 2 componentes principais desta solução:

1. Processo que verifica a fila de espera, que não é mais do que uma tabela em base de dados
2. Replicação para essa fila de espera nas aplicações WebParceiros e SFAWeb

Arquitectura Lógica

A arquitectura lógica da solução é a seguinte:

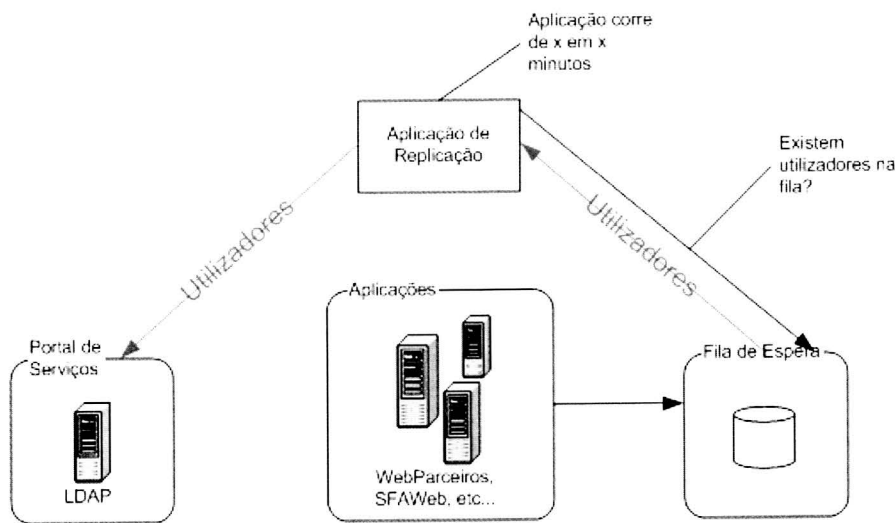


Fig. 4-23 Replicação - Arquitectura Lógica

Descrição de Implementação

Integração Genérica

Para efectuar a integração, existe um programa em JAVA, chamado DbLdapReplicator, que corre como um processo de *background*, que consulta a fila de espera e efectua as mudanças respectivas, caso existam. Estas mudanças podem ser alteração de palavras-chave de utilizadores já existentes ou criação de novos utilizadores.

Este programa opera com o seguinte fluxo:

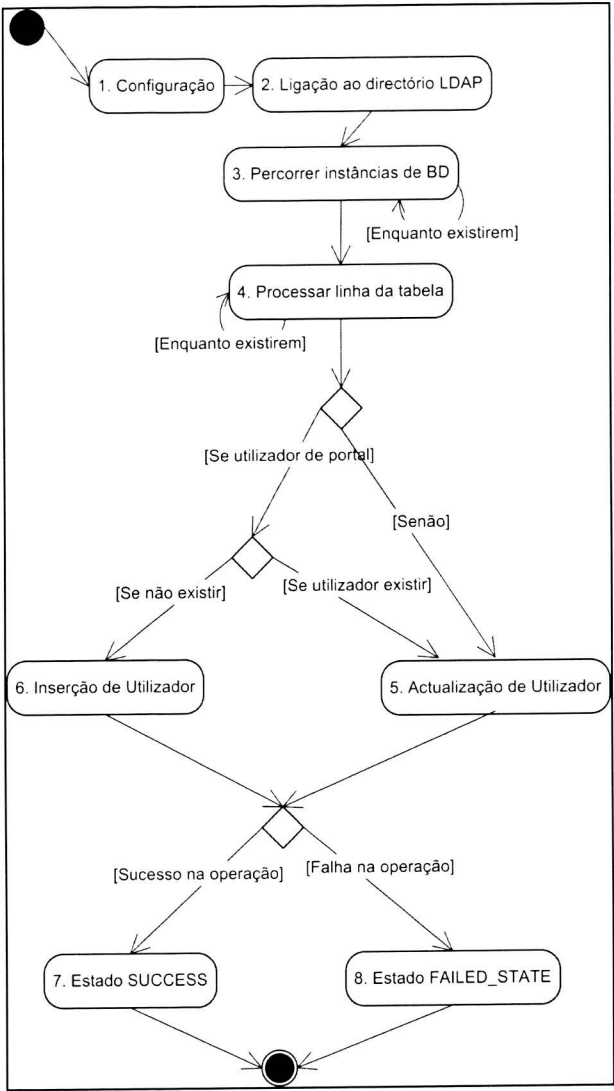


Fig. 4-24 Replicação - Diagrama de Estados do Mecanismo de Replicação

O significado completo de cada um dos estados é explicado a seguir:

1. **Configuração** – Consiste na leitura de um ficheiro de propriedades onde se especificam a lista de bases de dados que contém tabelas **LDAP_REPLICATION**, e aplicações a que se referem. Caso seja à aplicação WebParceiros, os utilizadores especificados são denominados “Utilizadores de Portal”, sendo criados no directório LDAP do *iPlanet Portal Server* se não existirem.
2. **Ligação ao directório LDAP** – Liga-se ao directório LDAP
3. **Percorrer instâncias de BD** – ciclo que percorre todas as instâncias lidas em 1.
4. **Processar linha da tabela** – esta operação lê e interpreta todas as linhas da tabela; passando para várias decisões
 - a. se o utilizador for de portal
 - i. se não existir no directório LDAP passa para 6.
 - ii. se existir passa para 5.

b. se o utilizador não for de portal

i. passa para **5.**

5. Actualização de utilizador – Actualiza um utilizador no directório LDAP

6. Inserção de utilizador – Insere um utilizador no directório LDAP

7. Estado SUCCESS – Replicação bem sucedida, a linha da tabela é apagada

8. Estado FAILED_STATE – Replicação falhou, a linha da tabela guarda o estado FAILED_STATE e é guardado também o erro correspondente

Alteração das Aplicações

As aplicações existentes *SFAWeb* e *WebParceiros* tiveram de ser alteradas, para inserir na fila de espera:

- Alterações de utilizadores
- Criação de novos utilizadores.

Estas alterações foram efectuadas por restantes membros da equipa do projecto.

5 Conclusão

Este capítulo destina-se a focar e comentar 3 pontos essenciais.

Trabalho Realizado

A nível pessoal, todo este projecto e o seu enquadramento foram enriquecedores em numerosos aspectos. A nível técnico foram utilizadas diversas tecnologias, muitas das quais inicialmente desconhecidas ou pouco familiares. Foi necessário todo um processo de aprendizagem, treino e utilização das mesmas que trouxeram bastantes conhecimentos para o autor. Simultaneamente, todo o contacto com os fluxos reais de informação dentro de uma empresa e metodologias de desenvolvimento (escrita de relatórios, actas, documentação de código e processos de maturação dos projectos) foram extremamente enriquecedoras. A componente de trabalho em equipa é uma constante na Vodafone Telecel S.A. Foi necessário desenvolver código de um modo coerente para que novos membros se possam rapidamente adaptar ao trabalho realizado. Existiu também a oportunidade de trabalhar e melhorar aplicações desenvolvidas por outros, sendo necessária uma adaptação ao código existente.

O estágio proposto não foi completamente finalizado nos 6 meses. Existiram atrasos internos que acabaram por dificultar a força de desenvolvimento. No entanto, os principais objectivos foram atingidos, tendo-se criado uma plataforma funcional que permite disponibilizar serviços rapidamente e em tempo eficaz para o mercado.

Contribuições para a Vodafone Telecel S.A.

O piloto construído durante o estágio, prova que a plataforma estudada (*iPlanet Portal Server*) é de facto segura, expansível e funcional. Se a Vodafone quiser lançar um novo serviço para concorrer no mercado, possui uma plataforma que permite rapidamente integrar esse serviço num Portal comprovadamente estruturado e fiável. Isto traz grandes vantagens do ponto de vista de *Time to Market*, aumentando a capacidade de resposta da empresa. Simultaneamente, o trabalho desenvolvido constituiu uma rota de migração de certas aplicações e perfilagem de utilizadores para um ponto centralizado, facilitando a sua gestão e diminuindo os seus custos.

O mecanismo de autenticação por *hardware* desenvolvido para o Portal, acabou por ser aprovado pela direcção de Sistemas Informáticos da Vodafone, tendo sido implementado numa aplicação real disponibilizada a clientes finais: WebParceiros.

Perspectivas Futuras

O piloto encontra-se construído de modo a que seja fácil evolui-lo. O código desenvolvido é modular nesse sentido, já que a adição de um novo serviço ao Portal implica escrever pouco código.

A nível de melhoramentos futuros, pode-se referir a passagem de toda a lógica de apresentação das aplicações para o *software* que corre no Portal. Isto possibilitaria centralizar ainda mais os dados de utilizadores e libertar o esforço de cada aplicação em desenhar a sua própria lógica de apresentação. No entanto isso implicaria uma migração lenta que teria de passar por uma reestruturação mais profunda da perfilagem no directório LDAP e pela

reescrita de muitas partes de aplicações existentes. O núcleo operativo do Portal pode ainda ser melhorado em certos aspectos, como sejam uma maior separação das funcionalidades de cada classe.

A adopção ou não desta plataforma num ambiente de produção real, está ainda pendente de alguns dados, principalmente de testes de *Hacking*. Só então a direcção dos Sistemas de Informação poderá medir custos, investigar opções e decidir o melhor caminho a seguir. Porém, nessa altura poderão saber exactamente o que é o iPlanet Portal Server e o que é capaz de fazer.

Referências e Bibliografia

Documentos Internos Produzidos pela Equipa do Projecto Portal de Serviços

portal_ref_desenvolvimento.doc – Descrição Técnica do Desenvolvimento do Portal de Serviços, Joel Alvim, 2002

single_sign_plan_20020308.doc – Revisão de “Planning And Deploying a Single Sign-On Solution”, Fonte: <http://developer.netscape.com/docs/manuals/security/SSO/sso.htm#1062945>, 2001

admin_guide_20020306 - Revisão de “iPlanet Portal Server 3.0 Administration Guide”, Fonte: <http://docs.iplanet.com/>, 2001

index_iportal_arch_20020304 – Revisão do documento “iPlanet. Portal Server 3.0 Overall Architecture”, Fonte: <http://docs.iplanet.com>, 2001

ldap_overview_20020313 - Visão geral do LDAP, Fontes: <http://www.netscape.com>, <http://www.openldap.org>, 2002

weblogic_intro_20020313 – Revisão de “Introduction to WebLogic Server 5.1”, Fonte: <http://edocs.bea.com>, 2001

wp_gateway_20020315 – Revisão do *White Paper* da componente Gateway do iPlanet Portal Server, Fonte: <http://developer.iplanet.com>, 2001

Documentos Internos Disponíveis

GD_WebParceiros – General Design do projecto “Solução Web para Parceiros”, Miguel Pereira e Benjamim Ferreira, 2001

Sítios Web

VODAFONE TELECEL, S.A. – Páginas do sítio <http://www.vodafone.pt>, Vodafone Telecel S.A., 2002

VODAFONE TELECEL, S.A. Intranet – Páginas do sítio interno da Vodafone, Vodafone Telecel S.A., 2002

BEA SYSTEMS – Páginas do sítio <http://www.bea.com>, Bea Systems, 2002

IPLANET – Páginas do sítio <http://www.iplanet.com>, iPlanet, 2002

SUN MICROSYSTEMS – Páginas do sítio <http://java.sun.com>, Sun Microsystems, 2002

ANEXO A: FeedbackReports, uma Nova Aplicação

Motivação

Esta aplicação foi desenvolvida para provar o quão seguro, integrado e fácil é, desenhar novas aplicações que façam uso directo das funcionalidades do iPS.

Além disto, esta aplicação destina-se a ajudar os operadores das equipas Vodafone 1210, e 1291, a coleccionar e traçar métricas dos problemas encontrados nos acessos dos utilizadores às aplicações SFAWeb e WebParceiros.

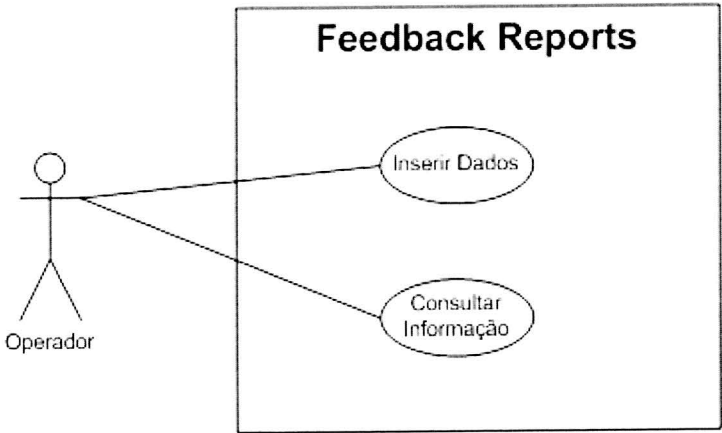


Fig. A-1 FeedbackReports - Diagrama de Casos de Uso

Análise de Requisitos

Inicialmente era requisito a existência de dois tipos de utilizadores: um gestor, e um operador. O operador só poderia consultar informação, e o gestor só a poderia adicionar.

No entanto, tal separação revelou-se inútil, e bastou a existência de apenas um utilizador da aplicação.

Arquitectura Lógica

A arquitectura lógica segue 4 camadas diferenciadas que comunicam entre si do seguinte modo:

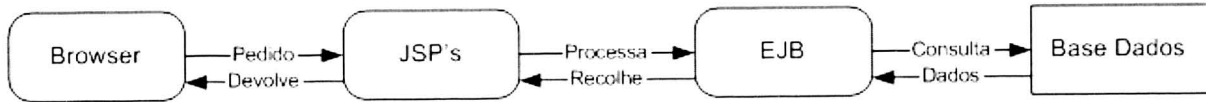


Fig. A-2 FeedbackReports - Diagrama de Fluxos

Este funcionamento é típico numa aplicação J2EE, em que se tem a camada de apresentação (JSP's), a camada de negócio (EJB), e finalmente a camada de bastidores (Base de Dados)²⁶.

²⁶ Para mais informação, consultar figura 2-9

A estrutura em base de dados que irá albergar o núcleo funcional da aplicação, é a seguinte:

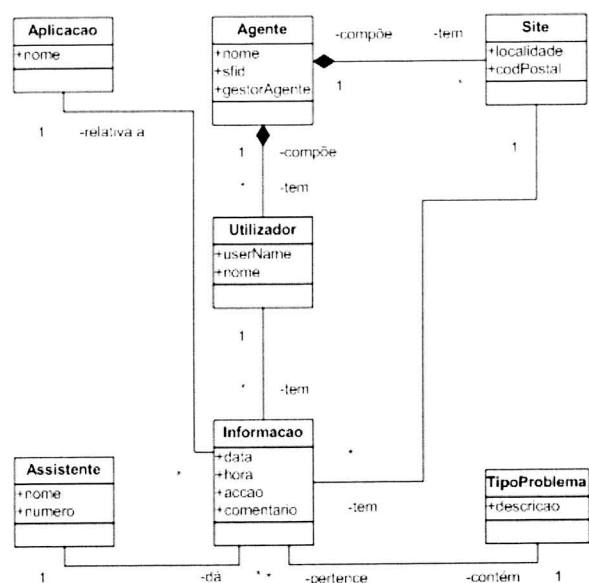


Fig. A-3 FeedbackReports - Diagrama de Classes

O objecto Aplicação, modela a aplicação à qual dada informação se refere. Essa informação consiste num registo que ocorreu numa dada hora, no qual se anota também a acção tomada para resolver o problema respectivo. O operador que ficou a cargo dessa ocorrência, está representado pela classe Assistente, e cataloga a informação numa série de problemas comuns, modelados por TipoProblema.

A informação registada, tem de estar associada a um utilizador, e a uma loja física. Esse utilizador faz parte de um dado Agente Vodafone, que tem várias lojas espalhadas pelo país.

Implementação

A implementação desta aplicação, passa pela criação de *Jsp's* que validem a sessão do utilizador no Portal, e lhe apresentem a interface *Web*. Apoiando esta camada, existe um *EJB* que encapsula os meios de aceder à base de dados e efectuar acções devidas.

Existem 3 ficheiros *JSP* principais. O ficheiro *head.jsp* apresenta as opções no topo do ecrã, e valida a sessão do utilizador no Portal.

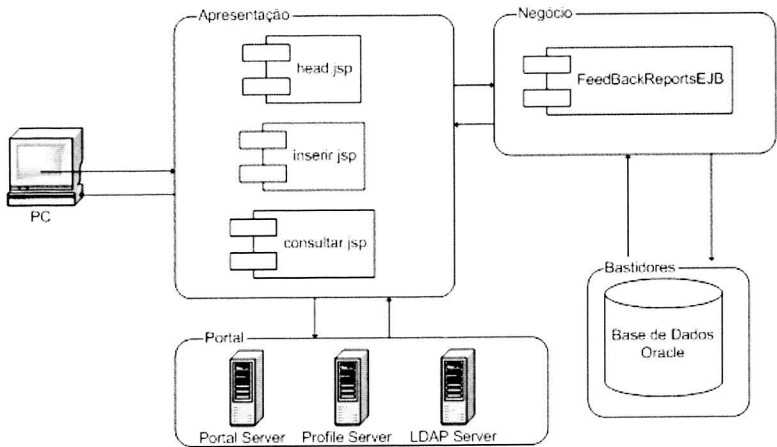


Fig. A-4 FeedbackReports - Diagrama de Componentes

O ficheiro *inserr.jsp* apresenta um formulário que o operador preenche, cujos elementos serão passados ao EJB para inserção na base de dados.

[Submeter Relatório][Estatísticas][Logout]

Agente	Stand Asla
Localização	Amadora 2720
Aplicação	WebParceiros
User	N_FABREU
Tipo de Problema	Conectividade
Questão	
Acção	

Inserir

Fig. A-5 FeedbackReports - Ecrã de Submissão de Dados

O ficheiro *consultar.jsp* apresenta por sua vez um formulário que permite consultar a informação armazenada na base de dados, segundo diversos critérios.

[\[Submeter Relatório \]](#) [\[Estatísticas \]](#) [\[Logout \]](#)



Pesquisa de registos por agente Stand Asla Pesquisar	Pesquisa de registos por utilizador M_AMARQUES Pesquisar
Pesquisa de registos por aplicação SFAWeb Pesquisar	Pesquisa de registos por tipo de problema Conectividade Pesquisar
Top 5 dos Problemas Pesquisar	

Fig. A-6 FeedbackReports - Ecrã de Relatórios

O *EJB* de acesso à base de dados, constitui um invólucro para operações de *select*, *insert*, *delete*, e *update* na base de dados, sendo usado pela camada de apresentação.

ANEXO B: Sítio Web do Estágio

Motivação

Este sítio *Web* destina-se a apresentar o projecto, bem como armazenar e compilar toda a documentação produzida para o mesmo. Deste modo, qualquer colaborador da Vodafone poderá de um modo fácil tomar conhecimento do que é o Portal de Serviços. Por outro lado, poderá ainda tomar conhecimento sobre as tecnologias utilizadas, e outra informação relevante armazenada.

Análise de Requisitos

Foram considerados como requisitos do sítio *Web* os seguintes pontos:

- Qualquer pessoa que entre no sítio, poderá consultar documentação (em forma de ficheiros *PDF*), bem como informação geral que seja apresentada em forma de *HTML*.
- Deverá existir um administrador, que possa inserir novos documentos no sítio

O diagrama de casos de uso que daqui resulta será:

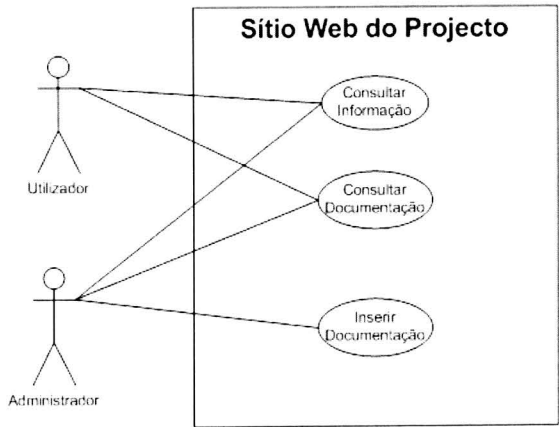


Fig. B-1 Sítio Web Interno - Diagrama de Casos de Uso

Arquitectura Lógica

Tal como na aplicação *FeedbackReports*, esta também se divide em 3 camadas distintas e bem definidas: a camada de apresentação, negócio e bastidores.

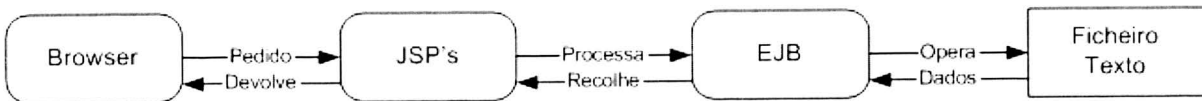


Fig. B-2 Sítio Web Interno - Diagrama de Fluxos

No caso desta aplicação, o acesso e armazenamento de informação, é feito num ficheiro de texto em formato pré-definido. Este passo foi tomado, dado que na altura do desenvolvimento, não era viável esta aplicação aceder a uma base de dados existente.

O diagrama que modela a informação, é bastante simples:

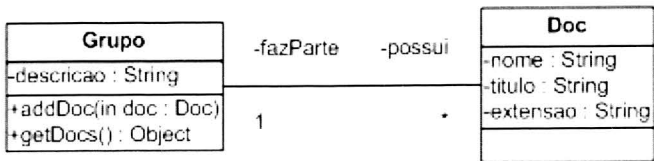


Fig. B-3 Sítio Web Interno - Diagrama de Classes

Um grupo corresponde a um conjunto de documentos que fazem parte do mesmo.

Implementação

Devido à inexistência de uma base de dados para modelar a informação, teve de se recorrer à ajuda de um ficheiro de texto simples. Foi adoptado um formato padrão que seria:

```
*
Nome do Grupo
Nome do Documento1 | Título do Documento 1 | Extensão do Documento 1
Nome do Documento1 | Título do Documento 1 | Extensão do Documento 1
...
**
```

O primeiro asterisco indica que vai começar um novo grupo. A linha imediatamente a seguir contém o nome do mesmo, e as restantes linhas armazenam os dados do documento em si.

Para finalizar o grupo, usa-se o código **. Este ficheiro pode repetir grupos sequencialmente neste formato, tantas vezes quantas as necessárias.

Vejamos agora os componentes que foram utilizados:

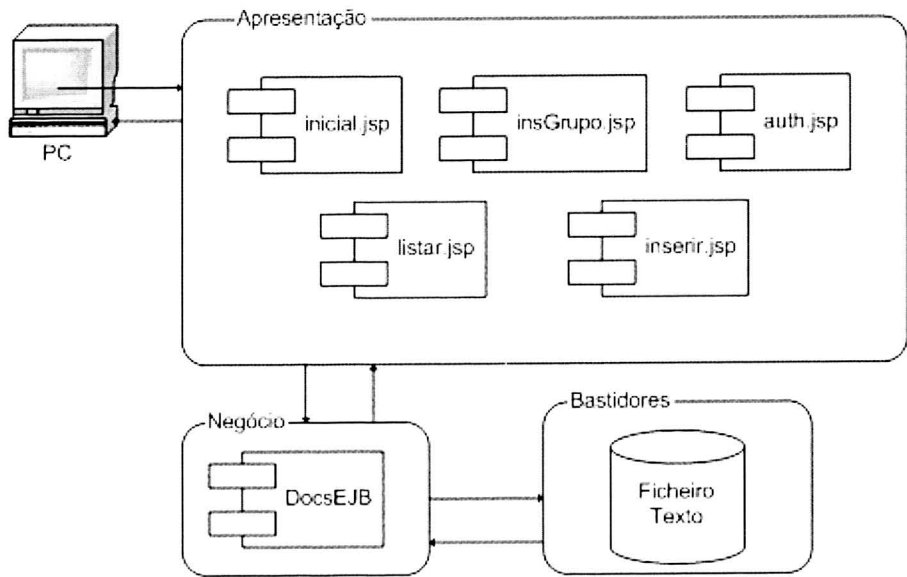


Fig. B-4 Sítio Web Interno Diagrama de Componentes

As JSP's existentes são responsáveis por apresentar a interface com o utilizador.

O ficheiro *incial.jsp* exhibe um ecrã introdutório:

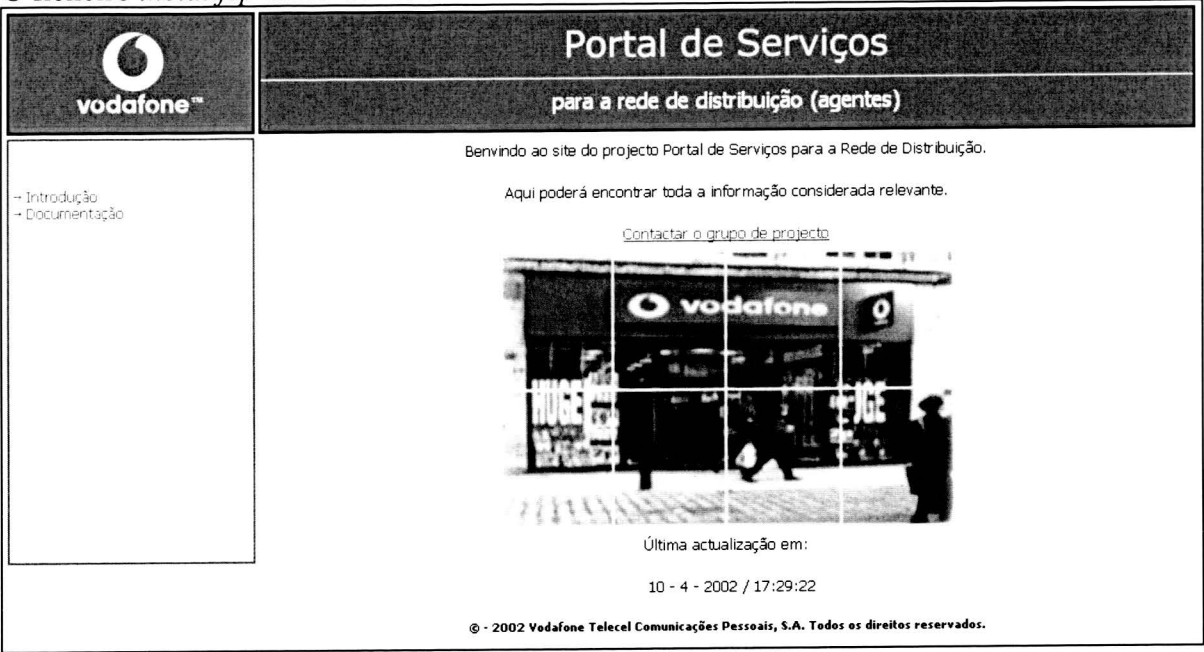


Fig. B-5 Sítio Web Interno - Ecrã de Entrada

No que respeita à documentação, é possível consultá-la através do ficheiro *listar.jsp*. Este simplesmente consulta todos os grupos disponíveis e exhibe os documentos contidos nos mesmos, de uma forma ordenada:

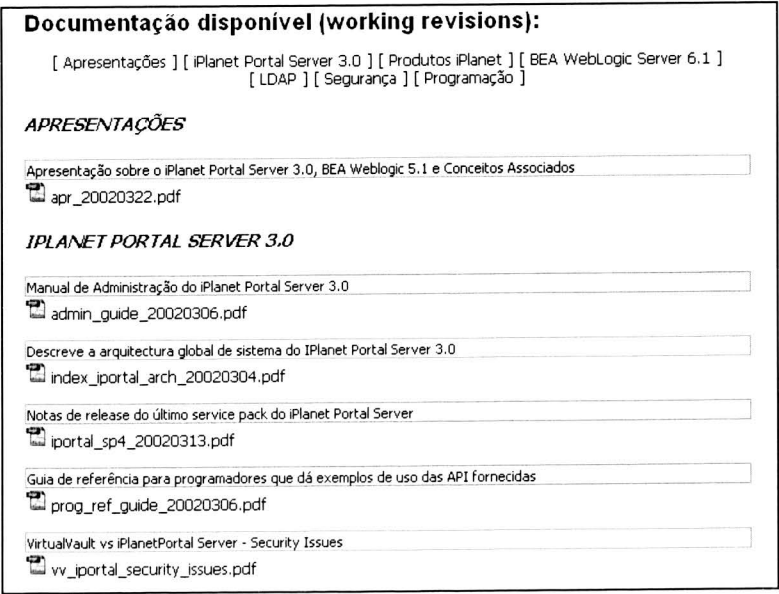


Fig. B-6 Sítio Web Interno - Ecrã de Listagem de Documentos

Para inserir um novo grupo, ou um novo documento através do sítio *Web*, é necessário passar uma fase de autenticação, garantida pelo ficheiro *auth.jsp*. Este possui um formulário com campos para *login* e palavra-chave. Ao serem submetidos, estes campos serão validados como um pedido *HTTP Basic Authentication* ²⁷ no servidor *Web* que dispõe o sítio.

Passada esta fase, é possível inserir novos grupos. Este serviço é disponibilizado pelo ficheiro *insGrupo.jsp*:

Inserir Novo Grupo:

Grupos existentes actualmente:

Apresentações

iPlanet Portal Server 3.0

Produtos iPlanet

BEA WebLogic Server 6.1

LDAP

Segurança

Programação

Apresentações

iPlanet Portal Server 3.0

Produtos iPlanet

BEA WebLogic Server 6.1

LDAP

Segurança

Programação

Apresentações

iPlanet Portal Server 3.0

Produtos iPlanet

BEA WebLogic Server 6.1

LDAP

Segurança

Programação

Nome do Novo Grupo:

OK

Fig. B-7 Sítio Web Interno - Ecrã de Inserção de Grupos

É ainda possível inserir documentos num grupo já existente. Estes documentos são enviados para o servidor através de um pedido *HTTP Post* ²⁸ em formato *MIME* ²⁹.

Inserir Novo Documento:

Ficheiro:

Browse...

Descrição:

Grupo:

Apresentações

Upload

Limpar

Fig. B-8 Sítio Web Interno - Ecrã de Inserção de Novos Documentos

²⁷ Autenticação definida no protocolo HTTP, que valida um *login* e palavra-chave em determinado “reino”.

²⁸ Mecanismo de envio de dados do cliente para o servidor, através do protocolo HTTP.

²⁹ *Multipurpose Internet Mail Extensions*, permitindo mensagens com conteúdo não textual, multi-parte, etc.

Finalmente, está presente na camada de negócio, o *Enterprise Java Bean* DocsEJB, que disponibiliza meios de operar na informação guardada.

ANEXO C: Ramificação ao Projecto do Portal de Serviços

Motivação

Durante a realização do projecto Portal de Serviços, foi proposto ao grupo de trabalho que implementasse a validação por *hardware* utilizada, numa outra aplicação da empresa. A aplicação em questão é o *WebParceiros*.

Visto que a solução implementada no Portal funcionava e era viável, tornou-se oportuno passar esta solução para um caso real de produção, em que possíveis fraudes podem acontecer em qualquer momento. Assim garante-se que os utilizadores desta aplicação, só a poderão usar no seu local de trabalho: a loja do agente para o qual trabalham.

Para efeitos deste relatório, não é relevante conhecer as funcionalidades desta aplicação.

Análise de Requisitos

A aplicação *WebParceiros* possui 3 tipos distintos de utilizadores:

- *Master* : responsável por um agente
- *Normal* : utilizador dentro de um agente
- *Back-Office* : utilizador auxiliar dentro da Vodafone

O requisito principal consiste em validar que um utilizador *Normal* só pode aceder à aplicação dentro das lojas do agente para o qual trabalha.

No caso de ser um utilizador *Master*, ele poderá aceder sem validar o seu PC. Terá então hipótese de registar esse PC na aplicação, para que os seus subordinados possam mais tarde aceder a partir dele. Notar que este registo foi implementado por restantes elementos da equipa do projecto, pelo que não será referido aqui.

Um utilizador de *Back-Office* não necessita de ser validado por *hardware* na aplicação.

Arquitectura

A aplicação inicial apenas validava o utilizador pelo seu *login* e palavra-chave. Neste caso, terá também de validar um número de série presente no PC, que o identifique subsequentemente.

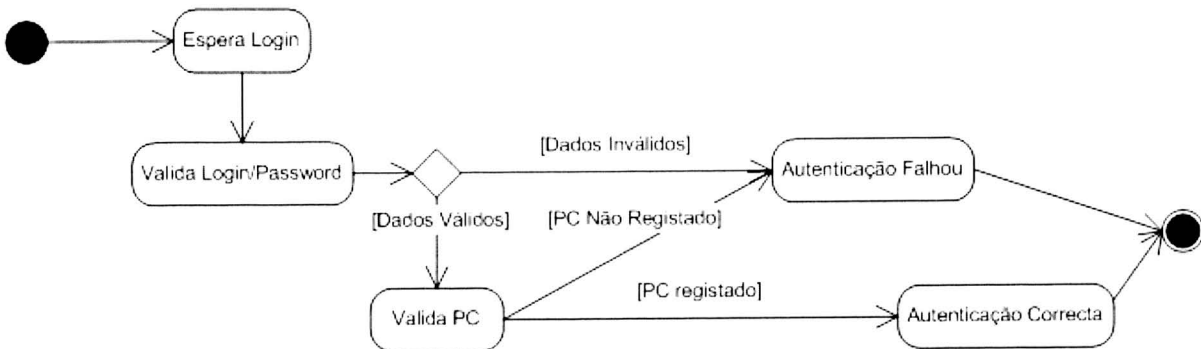


Fig. C-1 WebParceiros - Diagrama de Estados do Novo Login

O modelo de dados existente sofreu ligeiras alterações, de modo a suportar os novos requisitos.

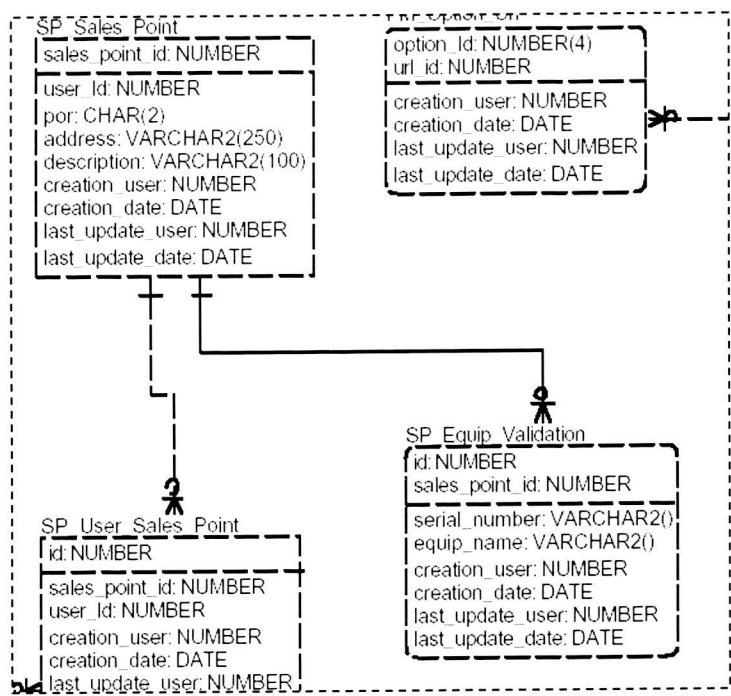


Fig. C-2 WebParceiros - Alterações ao Diagrama de Classes

Tabela: SP_EQUIP_VALIDATION:

Sequência: SP_EQUIP_SEQ

Trigger: SP_EQUIP_VALIDATION_TRIGGER

Chave Estrangeira: FK_SP_EQUIP_VALIDATION

Chave Primária: PK_SP_EQUIP_VALIDATION

A nível aplicacional, a arquitectura lógica apoia-se na estrutura típica contendo hierarquicamente:

- Camada de Apresentação (*JSP's* e *Servlets*)
- Camada de Negócio (*EJB's*)
- Camada de Bastidores (Base de Dados Oracle)

Esta estrutura já foi referida com mais pormenor nos anexos anteriores.

Implementação

Visão Geral dos Componentes de Apresentação

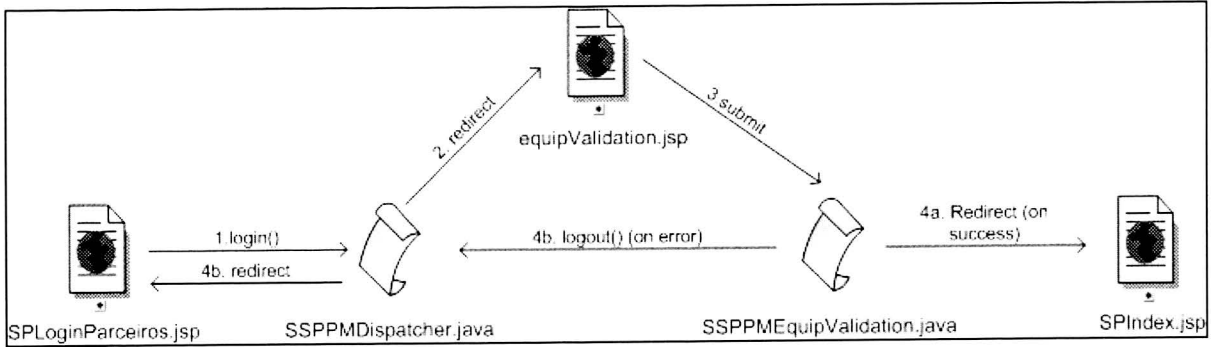


Fig. C-3 WebParceiros - Sequência de Estados do Novo Login

Como se evidencia na imagem, a primeira fase de validação do *login* e palavra chave cabe ao componente SSPPMDispatcher. Caso estes estejam correctos, reenvia o pedido HTTP para uma *JSP* intermédia, que descarrega o ActiveX responsável por ler o número de série do PC. Esta *JSP* envia esse número para um outro componente, o SSPPMEquipValidation. É este que valida o número recebido, e fornece ou não acesso na aplicação. Para essa validação, utiliza um EJB comentado a seguir.

Camada de Negócio

Na camada de negócio foi construído um *Enterprise Java Bean* próprio para auxiliar algumas operações deste fluxo. Esse EJB, denominado PMSPEquipValidationEJB, pode efectuar as seguintes operações:

- Verificar se um dado número de série já se encontra registado na base de dados.
- Listar os nomes dos equipamentos associados a um certo ponto de venda, ou todos os equipamentos associados a um determinado agente.
- Actualizar o nome de um equipamento de determinado ponto de venda.
- Verificar se o equipamento com dado nome já existe em determinada loja.
- Obter o nome do PC e loja correspondente, a partir do seu número de série.
- Activar um equipamento
- Desactivar um equipamento

Para efeitos de autenticação, a operação que interessa mais é a primeira: validar que o número de série do PC encontra-se na base de dados.

Vejamos agora a sequência mais detalhada dos processos que ocorrem numa autenticação correcta e completa de um utilizador *Normal*:

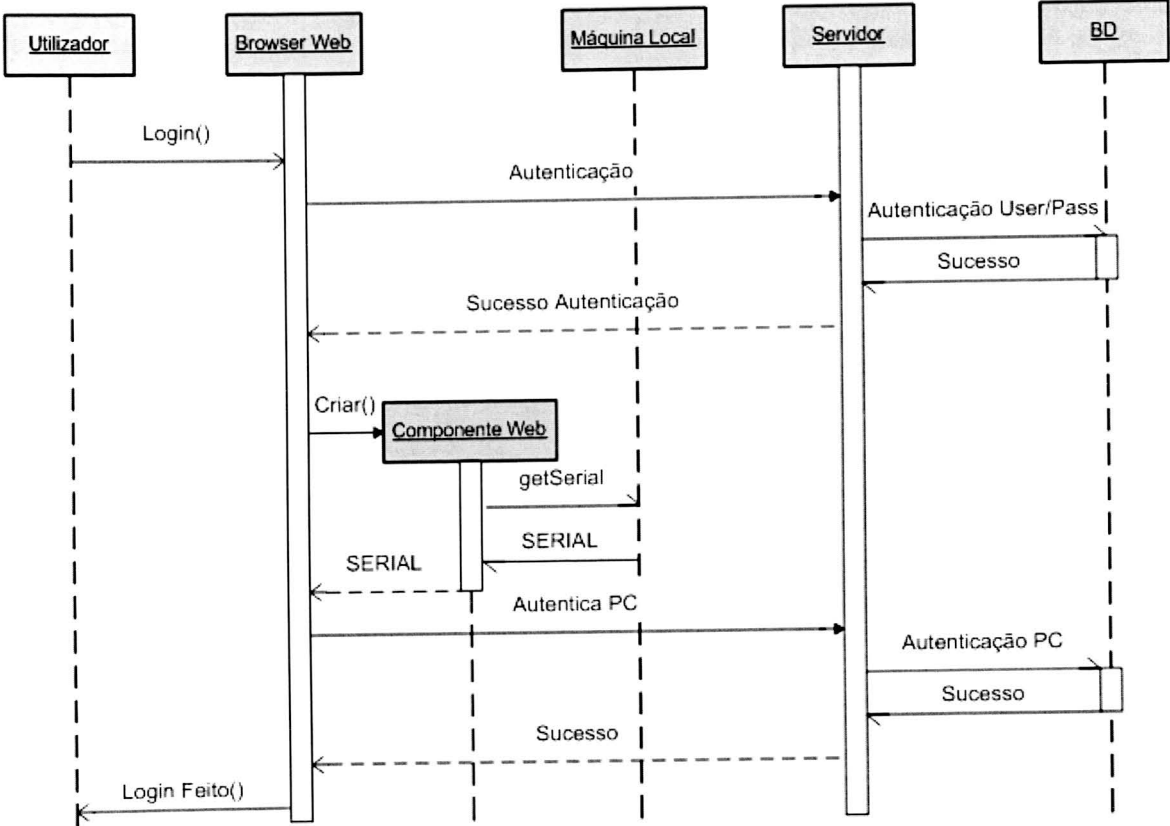


Fig. C-4 WebParceiros - Diagrama de Sequência do Novo Login

Anexo D: Mapa Do Estágio

Este anexo visa fornecer um mapa indicador do trabalho efectuado, incluindo as principais tarefas definidas, e o tempo que cada uma delas durou.

Convém referir que certas tarefas foram efectuadas em paralelo pelo grupo de projecto, ocorrendo no entanto nas mesmas alturas.

Este mapa divide-se em 5 fases essenciais:

- Início do Estágio e Introdução à Vodafone Telecel S.A.
- Desenvolvimento do Projecto Piloto, integrando aplicações e *Single Sign-On* nas mesmas
- Demonstração do Piloto à *board* de Directores, e aperfeiçoamento do mesmo
- Implementação do sistema de segurança física do Piloto numa aplicação real, já em produção
- Migração do Piloto para uma arquitectura possível de produção

1ª Fase

ID	Tarefa	Início	Fim	Duração	Mar 2002																															Abr 2002						
					1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7
1	Documentação Inicial	01-03-2002	01-04-2002	22d																																						
2	Formação Servidores Aplicacionais	15-03-2002	21-03-2002	5d																																						
3	Site do Projecto	22-03-2002	28-03-2002	5d																																						
4	Disponibilização da Seghex	15-03-2002	04-04-2002	15d																																						
5	Reunião Com Orientador da FEUP	05-04-2002	05-04-2002	0d																																						

2ª Fase

[illegible]

3ª Fase

ID	Tarefa	Início	Fim	Duração
1	Aperfeiçoamento do Piloto	05-06-2002	14-06-2002	8d
2	Renovação de Documentação	10-06-2002	14-06-2002	5d
3	Aperfeiçoamento da Apilet	06-06-2002	07-06-2002	2d
4	Aperfeiçoamento Sison WebParceiros	10-06-2002	14-06-2002	5d
5	Anúncio da Integração da Solução no WebParceiros de Produção	14-06-2002	14-06-2002	0d

4ª Fase

[illegible]

5ª Fase

ID	Tarefa	Início	Fim	Duração
1	Preparar a Nova Máquina para o Portal (Lisboa)	01-08-2002	01-08-2002	1d
2	Criação de Classes Gerais de Apoio Ao Single Sign On	02-08-2002	20-08-2002	13d
3	Single Sign On SFAWeb	15-08-2002	21-08-2002	5d
4	Nova Máquina Disponibilizada	13-08-2002	13-08-2002	0d
5	Preparação para Migração Nexenter	21-08-2002	23-08-2002	3d
6	Migração	26-08-2002	30-08-2002	5d





FACULDADE DE ENGENHARIA

UNIVERSIDADE DO PORTO

BIBLIOTECA



000065324

004
EIC50