Departamento de Engenharia Informática

Acesso Remoto e Móvel a Serviços Seguros

TRABALHO FINAL DE CURSO

do Curso de
LICENCIATURA EM ENGENHARIA INFORMÁTICA E DE COMPUTADORES (LEIC)

Ano Lectivo 2004 /2005

N.º da Proposta: 01

Título: Acesso Remoto e Móvel a Serviços Seguros

Professor Orientador:

Prof. Alberto Manuel Ramos da Cunha _____(assinatura)_____

Co-Orientador:

Eng. Miguel Filipe Leitão Pardal _____(assinatura)_____

Alunos:

nº 48317, Miguel Pina Xavier _____(assinatura)_____

nº 49716, Nelson Filipe Carvalho Santos _____(assinatura)_____

Agradecimentos

Acreditamos que o trabalho não teria corrido tão bem sem algum do apoio recebido. Agradecemos em especial ao Prof. Alberto Cunha e ao Eng. Miguel Pardal. Ao Prof. Alberto Cunha pelo esforço inicial em nos fazer investigar o problema numa perspectiva de engenheiro. Ao Eng. Miguel Pardal por ter estado sempre presente, pelo excelente apoio dado no âmbito do trabalho, e pelo contributo na nossa formação como engenheiros, tanto através de conselhos, como de metodologia de trabalho.

Agradecemos também todo o esforço efectuado pela Link Consulting, e respectivos elementos do departamento ISN. Em particular agradecemos ao Eng. João Almeida, Eng. Jorge Mendes, Eng. Paulo Barreto, Eng. Pedro Mota, Eng. Vítor Galveia . Sem o material disponibilizado e sem o apoio recebido, não nos teria sido possível realizar o trabalho adequadamente.

Resumo

O mercado dos dispositivos móveis está em evolução constante, estando este tipo de dispositivos bastante divulgado na actualidade. A contínua evolução da tecnologia móvel e do seu mercado criam imensas oportunidades na área dos serviços disponibilizados através deste género de tecnologia. Mas, associados a estas oportunidades surgem grandes desafios.

Este trabalho surge para tentar estudar e resolver alguns destes desafios, entre os quais se destacam a segurança adequada ao valor do serviço, a ser prestado, e a reutilização de tecnologia.

De modo a aprofundar o estudo nesta área estudou-se um caso prático que culminou na realização dum protótipo.

Palavras Chave

Segurança, mobilidade, valor do serviço, ambiente desligado, reutilização de tecnologia, cartões inteligentes, comunicação sem fios.

Índice

Lista de Figuras

Lista de Tabelas

Lista de Siglas

AWT	Abstract Window Toolkit
AML	Área Metropolitana de Lisboa
API	Application Program Interface
CDC	Connected Device Configuration
CLDC	Connected Limited Device Configuration
DAO	Data Access Object
DB4O	Data Base for Objects
EE	Execução Exame
GC	Gestão Central
GS	Gestão Sala
GTML	Generic Transport Mask Light
IC	Circuito Integrado
JAR	Java Archive File
JDBC	Java Database Connectivity
JNI	Java Native Interface
JSR	Java Specification Request
JVM	Máquina Virtual Java
J2ME	Java Micro Edition
J2SE	Java Standand Edition
MAC	Message Authentication Code
OBEX	Object Exchange
OO	Orientado a Objectos
ORB	Object Request Broker
PDA	Personal Digital Assistant
RMI	Remote Method Invocation
SAM	Secure Access Mode
SGBD	Sistema de Gestão de Base de Dados
SO	Sistema Operativo
SQL	Structured Query Language
VM	Máquina Virtual
XML	Extensible Markup Language

1 Introdução

Actualmente assiste-se a uma grande divulgação dos dispositivos móveis, especialmente dos telefones móveis, existindo similarmente grande evolução na utilização dos PDA (Personal Digital Assistant), sendo que estes dispositivos atingem actualmente capacidades próximas das que até há pouco tempo eram exclusivas de computadores de secretária. A massificação dos dispositivos, bem como das tecnologias a estes associadas, levou a uma explosão na oferta de serviços à sociedade e na sua consequente utilização através dos referidos equipamentos móveis, existindo ainda grande margem de expansão no que respeita a serviços oferecidos através de dispositivos móveis.

A expansão da oferta de serviços através destes dispositivos dependerá, em grande parte, da segurança por eles oferecida, da tecnologia e infra-estrutura necessária, e da capacidade de reutilização de tecnologia [27].

Tendo como base o contexto descrito, o trabalho consistiu numa pesquisa na área da mobilidade e segurança, que foi realizada para se perceberem quais as dificuldades existentes, e que aspectos podem estabelecer a diferença competitiva de um serviço neste nicho de mercado.

Após a fase inicial de pesquisa, optou-se por utilizar um caso de estudo para validar os pontos identificados num caso prático. Escolheu-se a realização de exames universitários em formato electrónico por ser um caso que necessita de confiança na disponibilização do serviço, i.e., tem requisitos de segurança elevados, aos quais se acrescentam requisitos de mobilidade, robustez e independência de infra-estruturas.

Depois de escolhido o caso de uso, a fase seguinte consistiu no levantamento de requisitos, que foi efectuado usando um modelo estruturado de representação dos requisitos [6].

Depois da fase de levantamento de requisitos, abordou-se a solução através da construção de protótipos iniciais que serviram principalmente para validar a adaptação e testar a tecnologia a ser usada na solução a desenvolver.

Com os protótipos iniciais desenvolvidos e respectivos pontos validados, completou-se a especificação da solução, que já tinha sido iniciada durante a fase de levantamento de requisitos.

Após todas estas fases, foi possível desenvolver um protótipo de implementação do caso de estudo. O protótipo, que foi desenvolvido de modo a ser extensível, teve um desenvolvimento relativamente rápido, o que se deveu em grande parte à metodologia seguida durante todo o ano. As fases de levantamento de requisitos, especificação e prototipagem foram cruciais para um desenvolvimento objectivo e eficaz.

A figura seguinte resume a metodologia de desenvolvimento.

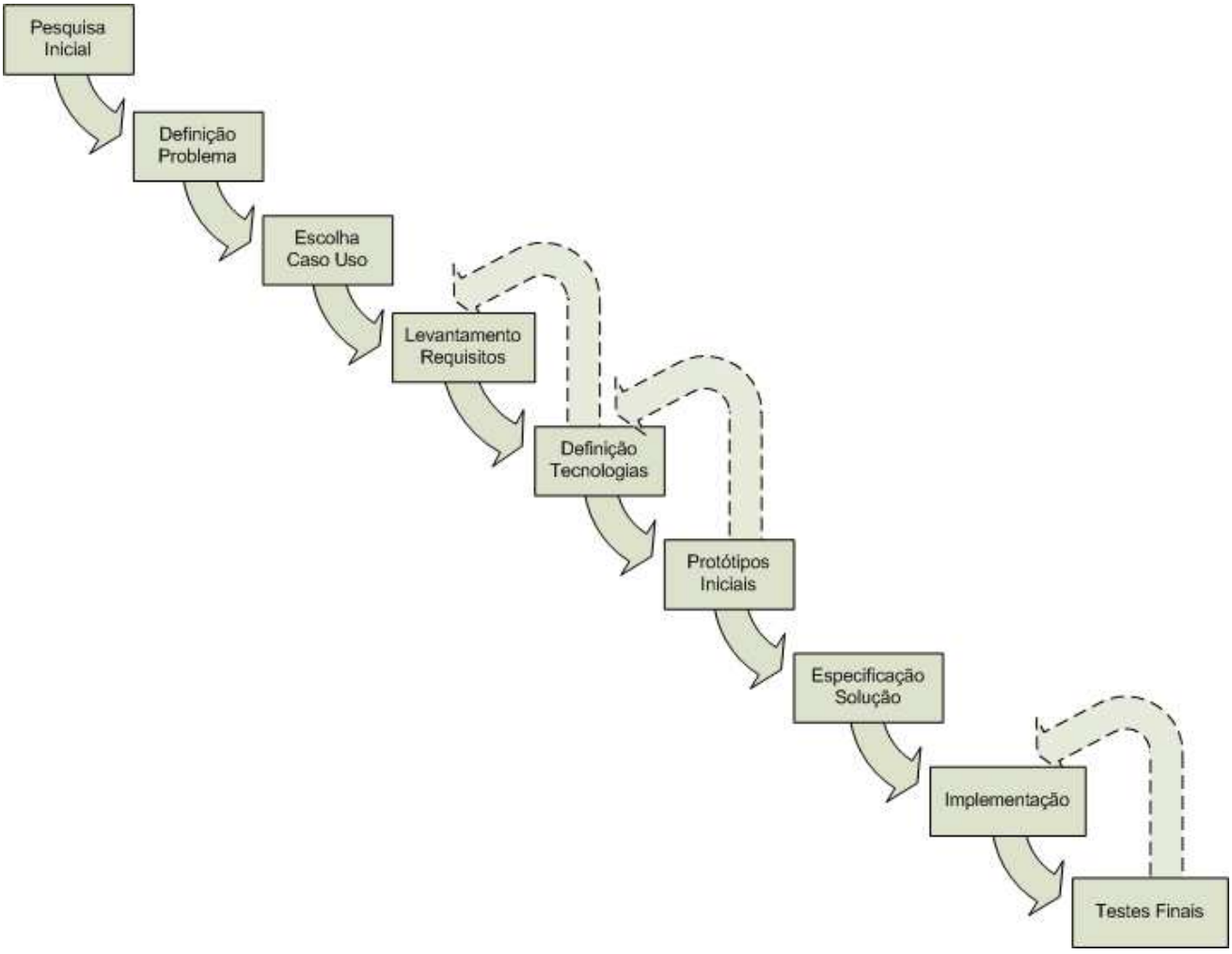


Figura 1.1 – Metodologia de desenvolvimento.

A solução obtida é bastante rica em termos tecnológicos e apresenta ainda alguma complexidade, em parte devido às inúmeras tecnologias envolvidas. O protótipo construído demonstra as principais virtudes da solução proposta.

2 Descrição do problema

Os dispositivos móveis e as tecnologias de redes sem fios criam inúmeras oportunidades de negócio, em variadas áreas de actividade económica. Mas, actualmente a segurança é um factor inibidor na adopção de tecnologias sem fios [27], pelo que, para a viabilização destas oportunidades é necessário que exista protecção adequada ao valor do serviço.

A segurança é, portanto, um ponto fulcral para garantir a confiança nos serviços prestados usando redes sem fios. No entanto, os riscos fazem parte da realidade das redes sem fios. Alguns riscos são similares àqueles que se encontram nas redes com fios. Outro tipo de riscos, tais como, acesso ilegítimo à rede, ataques de negação de serviço[1], entre outros, estão potenciados. Este aumento de riscos, associados às redes sem fios, tem a sua causa no facto de não existirem limitações físicas no acesso à rede [28] dentro de um perímetro que pode variar desde alguns metros até às centenas de metros.

O problema abordado neste trabalho relaciona-se directamente com as dificuldades em utilizar dispositivos móveis e tecnologias de redes sem fios para conseguir uma solução com segurança para viabilizar um serviço num ambiente distribuído, desligado e sem infra-estrutura permanentemente dedicada.

Os pontos principais que o caso aborda, contribuem para a discussão de possíveis soluções nesta área, com segurança e reutilização de tecnologia, pois estes são alguns dos aspectos que se forem resolvidos poderão garantir vantagens competitivas neste mercado.

[1] Em inglês Denial of Service (DOS)

3 Caso de estudo

Após o período inicial de pesquisas definiu-se um domínio que permitiu aprofundar de um modo prático os conceitos estudados, de modo a tornar possível a compreensão das dificuldades inerentes ao problema, que não surgiriam num estudo apenas teórico.

Na selecção de um caso de estudo procurou-se escolher um que se enquadrasse no problema em causa.

3.1 Descrição sumária do caso de estudo

O domínio escolhido foi o dos exames universitários electrónicos, que consiste na realização de exames universitários com recurso a dispositivos computacionais móveis.

O serviço para realização de exame, tem como principal diferença em relação ao modelo actualmente utilizado, o facto dos exames serem realizados em dispositivos computacionais em vez de ser em papel. A entrega do enunciado de exame bem como a entrega da resolução do exame por parte do aluno são feitas electronicamente.

Para definir o domínio teve-se sempre em conta o modelo actual de realização de exames, tendo como principal fonte de inspiração o Instituto Superior Técnico (IST), sendo de salientar que o domínio escolhido não pressupõe a existência de salas previamente equipadas para a realização do exame nem a proximidade física entre as salas, algo que é relevante no IST devido à sua realidade com múltiplos pavilhões e com dois pólos (Alameda e Tagusparque).

Vantagens em relação ao modelo actual

- **Logística**
    - Com este modelo de realização de exames poderá existir maior comodidade em termos logísticos para o corpo docente, abrindo-se portas para funcionalidades tais como a entrega e recolha automática de exames, correcção automática e diversificação automática de perguntas.

- **Segurança e fiabilidade**
    - Capacidade de oferecer maior confiança no serviço, em termos de garantias de entrega da resolução do exame e de realização de modo integro e confidencial, tanto por parte do docente, como do aluno.
- **Dados para gestão e apoio à decisão**[2]
    - Esta forma de realizar exames torna mais fácil eventuais processos de armazenamento[3] e extracção[4] de dados, evitando o preenchimento manual dos dados e consequente perda de informação daí decorrente, quer pela ocorrência de erros, quer pela necessária omissão de dados.

3.2 Enquadramento face ao problema

O caso de estudo da realização de exames escolares usando dispositivos móveis tem requisitos que se enquadram naqueles inerentes ao problema em estudo, nomeadamente, requisitos de segurança, fiabilidade e mobilidade, e cuja satisfação simultânea e integral se enquadra nas dificuldades descritas anteriormente.

Segurança

A realização de exames envolve questões de segurança, uma vez que este serviço tem um valor que tem que ser preservado, que se pode definir, não por medidas quantitativas, mas por medidas como a credibilidade da instituição perante a sociedade.

Para proteger o valor associado à realização electrónica de exames é, portanto, necessário responder a requisitos de segurança, nomeadamente:

- **Autorização**
    - É necessário garantir que apenas as entidades que têm permissão para executar determinada função ou aceder a informação no sistema o possam fazer. No

[2] Em inglês, *Business Intelligence*

[3] Em inglês, *Data Warehousing*

[4] Em inglês, *Data Mining*

presente caso de estudo, são exemplos disso, o acesso às funções que permitam a distribuição de enunciados e recolha das resoluções.

- **Integridade e autenticidade da informação trocada entre os participantes**
    - De modo a proteger o valor associado a este serviço, tem que existir, tanto por parte dos professores como dos alunos confiança na integridade, e autenticidade da informação que é trocada entre as partes, como o enunciado ou a resolução do exame;
    - No modelo actual a autenticação dos alunos é efectuada pelos professores através da fotografia presente no documento de identificação. Por parte dos alunos, existe confiança implícita nos professores.
- **Confidencialidade**
    - Para além da integridade e autenticidade da informação trocada entre os participantes no serviço, é necessário, em determinadas fases do serviço a confidencialidade dessa informação, de modo a proteger o valor do serviço. No caso de estudo, a confidencialidade aplica-se, por exemplo à informação relativa à resolução do aluno.

Fiabilidade e disponibilidade

É necessário garantir o correcto funcionamento dos processos associados à prestação do serviço, bem como garantir a durabilidade e disponibilidade da informação necessária à sua realização.

Distribuição e escalabilidade

A realização de um mesmo exame escolar, ocorre, muitas vezes, distribuída por diferentes salas, distanciadas no espaço.

O sistema tem momentos de utilização intensiva como a entrega de enunciados e recolha de provas, por exemplo, que exigem que o sistema suporte essa escala de utilização.

Mobilidade

Para garantir a mobilidade do serviço é necessário a utilização de dispositivos móveis com capacidade de estabelecer comunicação sem fios com outros dispositivos.

3.3 Análise de requisitos

Na análise requisitos usou-se o Modelo de Serviços Distribuídos [6], uma metodologia para especificar o sistema, representando requisitos não funcionais e a forma como influenciam a especificação do sistema.

A modelação é constituída por diferentes perspectivas: negócio, tecnologia, requisitos adicionais e especificação do sistema.

- **Perspectiva de negócio** define uma visão do serviço do ponto de vista do negócio, evidenciando o valor e os participantes na estrutura organizacional;
- **Perspectiva tecnológica** descreve as várias tecnologias utilizadas na concretização do sistema;
- **Perspectiva de requisitos adicionais** define requisitos não funcionais, derivados do negócio e da tecnologia, que têm que ser equilibrados entre si na implementação. Estes requisitos influenciam a qualidade da prestação de serviço;
- **Especificação do sistema** especifica os objectos do sistema em linguagem UML, representando aspectos estruturais e comportamentais, bem como os fluxos de dados e os fluxos de controlo. A definição dos dados e funcionalidades dos objectos dependem das perspectivas do negócio, dos requisitos adicionais e da tecnologia utilizada na sua implementação.

A criação e a evolução do sistema é ditada pelos requisitos de negócio, requisitos adicionais e pela tecnologia. Os requisitos influenciam-se uns aos outros, causando a evolução iterativa de si mesmos e da especificação.

Na análise de requisitos optou-se por usar este modelo porque proporciona uma metodologia em que os requisitos são estudados sobre diferentes perspectivas permitindo compreender de que forma os requisitos não funcionais, interferem com a especificação do sistema.

A metodologia usada, consistiu, na identificação do serviço, dos participantes, dos requisitos sobre diversas perspectivas, segurança, fiabilidade e tolerância a faltas, ergonomia e desempenho. Estes requisitos foram elaborados a partir do levantamento dos casos de uso principais, bem como dos casos de fraude. Finalmente, elaborou-se uma política de gestão de valor que tenta conciliar os diferentes requisitos, alguns antagónicos entre si.

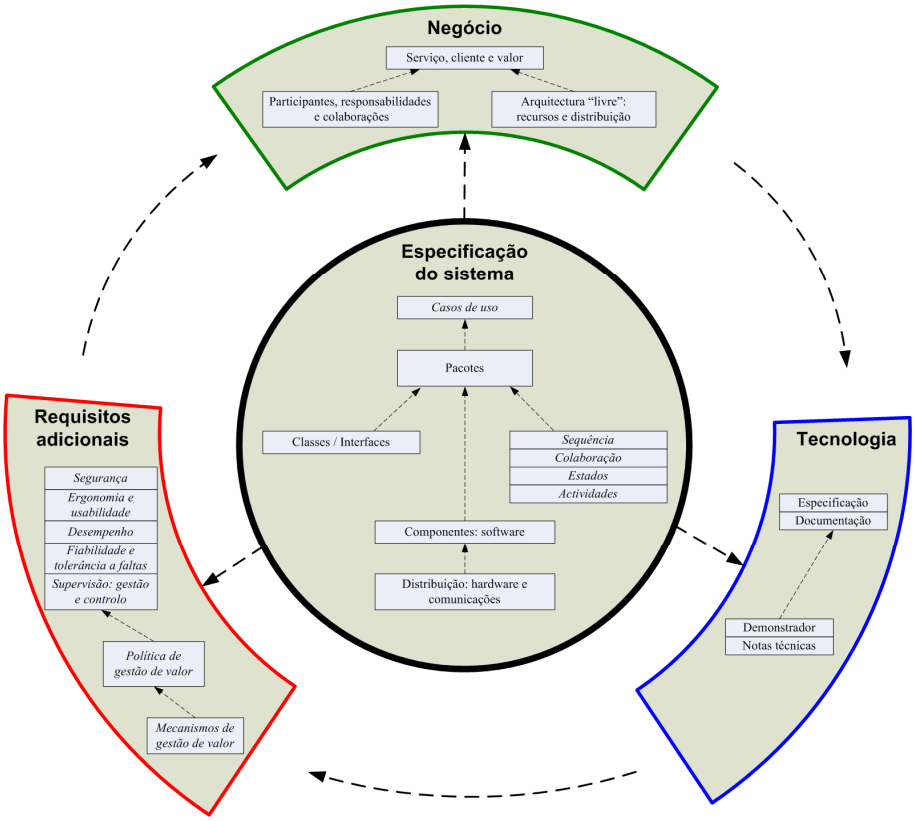


Figura 3.1 Modelo de serviços distribuídos[5]

[5] "A - - -> B" representa que "A depende de B". Os itens em itálico representam aspectos comportamentais do sistema e os outros representam aspectos estruturais.

3.3.1 Identificação do Serviço

Permitir a realização de exames através do uso de um identificador (ID) que proporcione o acesso a exames escolares.

3.3.2 Arquitectura Livre

A arquitectura livre permite identificar os principais recursos necessários, a sua distribuição e as ligações necessárias. Pretende transmitir a visão intuitiva da concretização do sistema, para explorar as potencialidades da tecnologia.

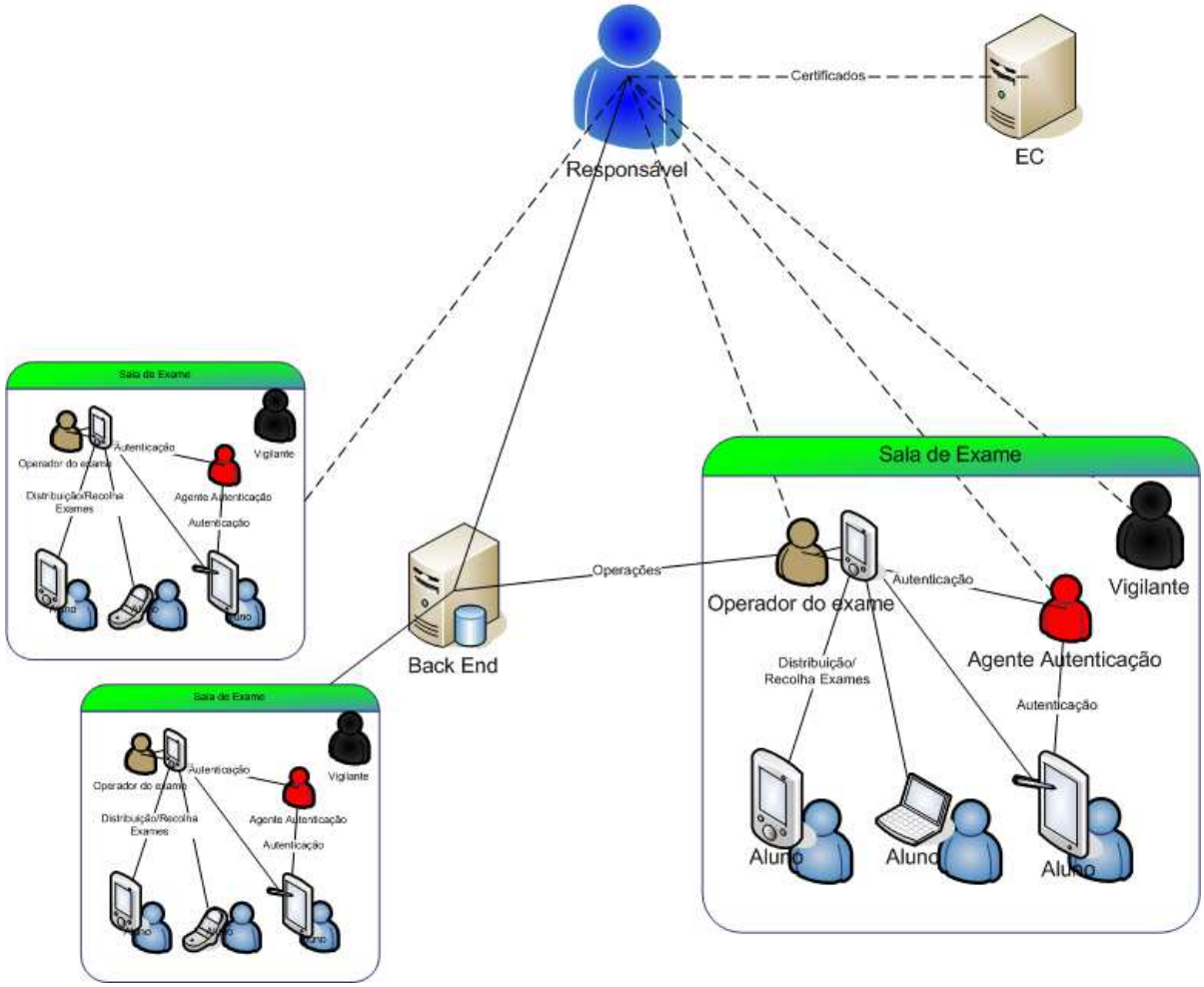


Figura 3.2 Diagrama que representa a arquitectura livre do serviço evidenciando a distribuição do serviço.

3.3.3 Proposições de Valor

- Permitir ao aluno realizar exames na instituição com recurso a equipamentos móveis;
- Oferecer uma melhor alternativa à realização clássica de exames;
- Reduzir tempo dispendido na logística associada aos exames;
- Permitir a realização distribuída do exame por diversas salas, com possíveis desfasamentos temporais na execução do serviço;
- Garantir a integridade do processo de armazenamento dos exames electrónicos e do processo de correcção dos exames.

3.3.4 Participantes, responsabilidades e colaborações

A gestão do valor do sistema é realizada pelas diferentes entidades. O valor do sistema tem que ser protegido de acordo com a política de gestão de valor. As entidades e as suas colaborações descrevem onde o valor pode ser criado, por quem, e como se move entre entidades no sistema.

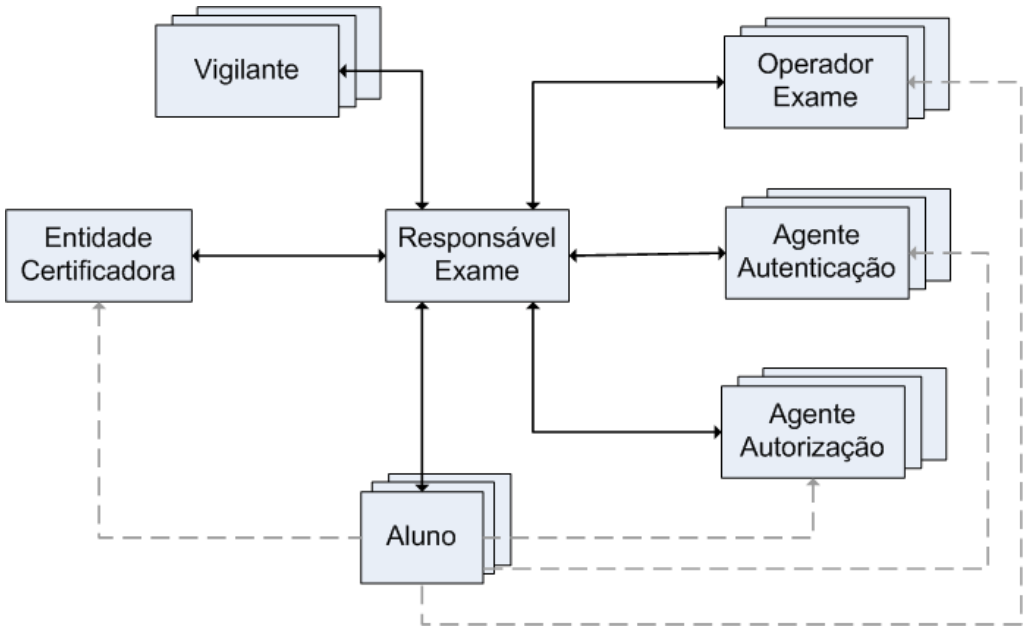


Figura 3.3 – Resumo das colaborações entre participantes na prestação de serviço

Tabela 3.1 – Responsabilidades e colaborações de cada participante.

Responsabilidades	Colaborações
Entidade Certificadora	
Emissão de identificadores (IDs)	
Responsável pelo Exame	
Coordenação da realização de exame	Entidade Certificadora Operador de Exame Agente Autorização Agente Autenticação Aluno
Aluno	
Realização do exame	Entidade Certificadora Responsável pelo Exame Operador de Exame Agente Autenticação Agente Autorização
Agente Autorização	
Autorização do aluno no acesso ao serviço	Entidade Certificadora Aluno
Agente Autenticação	
Autenticação do aluno no serviço	Entidade Certificadora Aluno
Operador de Exame	
Controlo do fluxo de operações do exame	Responsável do exame Aluno
Vigilante	
Vigilância do exame	Operador de Exame

3.3.5 Casos de Uso

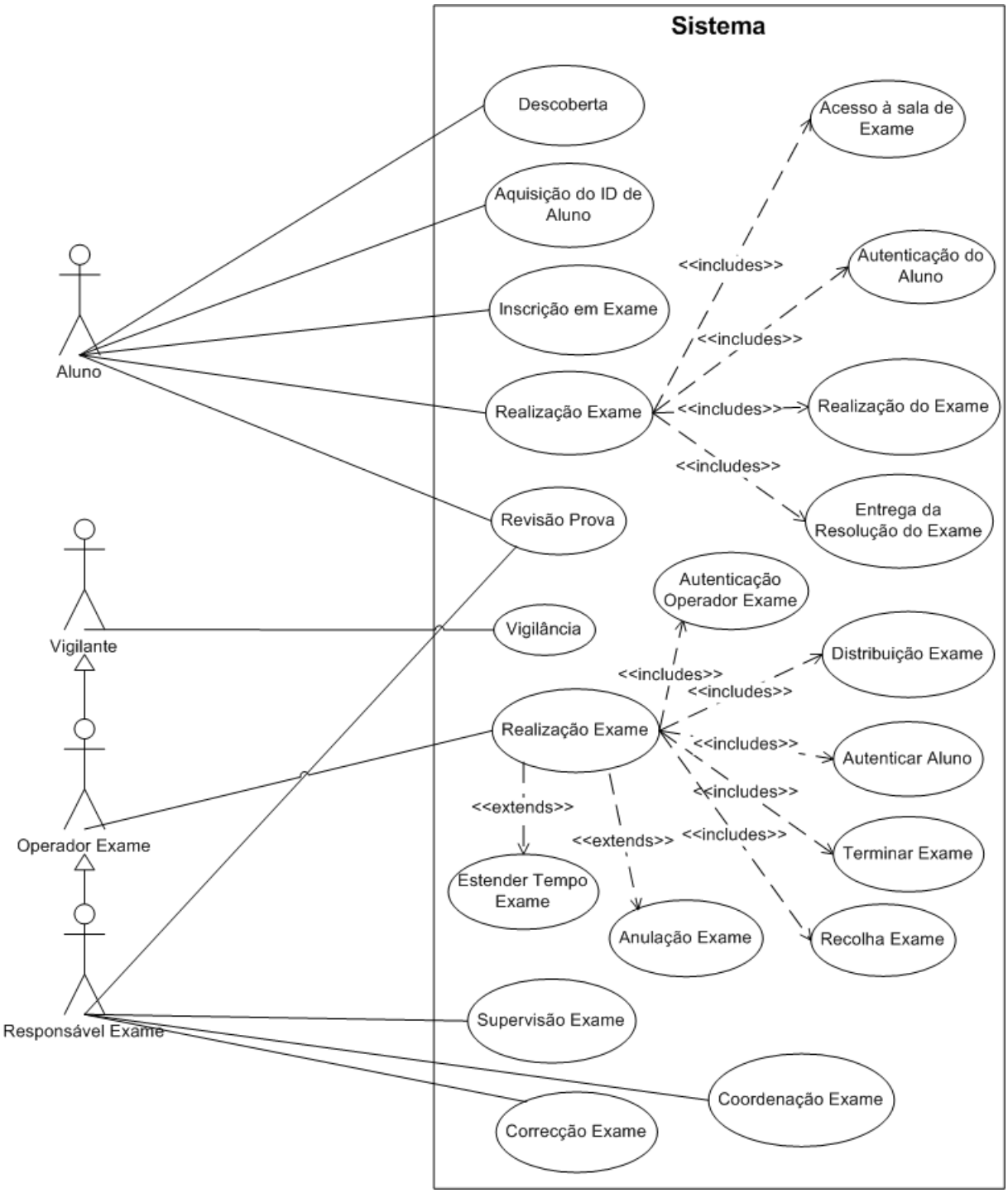


Figura 3.4 – Diagrama de casos de uso

3.3.6 Requisitos

Segurança: disponibilidade, integridade e confidencialidade

- A autenticação dos alunos tem que ser feita de forma presencial;
- Garantir privacidade e isolamento do aluno, de modo a que só o próprio consiga ter conhecimento da sua resolução durante a realização do exame;
- Garantir a integridade e autenticidade das informações trocadas entre as diferentes entidades participantes no prestação do serviço.
- Garantir que os exames são recolhidos de forma íntegra e privada;
- Garantir que o aluno não pode obter informação por meios não autorizados;

Fiabilidade e tolerância a faltas

- Garantir a fiabilidade do processo de distribuição e recolha das respostas;
- Garantir os requisitos de segurança, mesmo ocorrendo faltas relativas às comunicações;
- Garantir a durabilidade da informação associada à resolução do exame;
- Garantir o acesso aos dados necessários à autenticação mesmo num ambiente desligado[6];

Ergonomia e desempenho

- Usar suportes para os IDs que não sejam maiores do que o formato dos cartões de estudante actuais (dimensões equivalentes a cartão de crédito);
- A interface de resolução do exame deve ser fácil de usar;
- Disponibilizar informação sobre exames facilmente acessível e compreensível;
- Permitir facilidade e rapidez nas inscrições em exames;
- Permitir a recolha automática de exames;

[6] Off-line

- Permitir a realização do exame em diversas salas sem limitações de distância;
- Permitir a realização assíncrona do exame nas diversas salas;
- Permitir a alunos não inscritos realizar o exame;
- Suportar uma variedade de equipamentos;
- Permitir ao aluno ter acesso à sua resolução para efeitos de revisão de provas;
- O processo de autenticação, autorização, distribuição do exame e recolha do exame em cada sala devem ter um desempenho suficiente para que não haja atrasos significativos relativamente à hora estipulada para o início do exame, assim como para o instante em que o aluno pode abandonar a sala;
- A distribuição, recolha de exames, autenticação e autorização devem ter um desempenho suficiente para que todos os alunos, de uma determinada sala, usufruam do mesmo tempo para realizar o exame.

3.3.7 Política de Gestão de Valor

- O valor do serviço está centrado na autenticidade do aluno, integridade, confidencialidade, e garantia de entrega da prova de exame por ele efectuada;
- O ID do aluno prova a autenticidade da relação entre o aluno e a instituição de ensino, devido aos seus mecanismos de protecção. A gestão do ID de aluno deve ser íntegra garantindo o não repudiamento da identidade do aluno quando utilizada;
- O sistema terá que garantir que os exames não se perdem, garantindo a entrega nas condições referidas nos requisitos de fiabilidade e tolerância a faltas;
- O serviço realização de exame tem que garantir o cumprimento das regras da instituição na qual se realiza;
- Apenas e quando o cumprimento dos pontos anteriores não estiver em causa, deve tentar-se maximizar a satisfação dos requisitos de ergonomia e desempenho, pois são importantes para a aceitação generalizada do serviço como forma de fazer exames;
- Deve optar-se por utilizar tecnologia existente sempre que possível, desde que não comprometam os pontos anteriores desta política.

4 Arquitectura da solução

A arquitectura da solução é apresentada através de um diagrama de módulos de alto nível, com a descrição das funcionalidades de cada módulo, definição do modelo de domínio para os módulos principais, e foco nas operações principais da solução. Após as especificações, apresenta-se a tecnologia usada na implementação do protótipo.

4.1 Diagrama de Módulos

Após a análise de requisitos definiram-se os principais módulos [3] do sistema, estruturados da seguinte forma:

- Gestão Exame:
    - Gestão Central;
    - Gestão Sala;
    - Execução Exame;
- Gestão ID;
- Gestão Cadeira.

O diagrama de módulos seguinte representa a funcionalidade global do sistema.

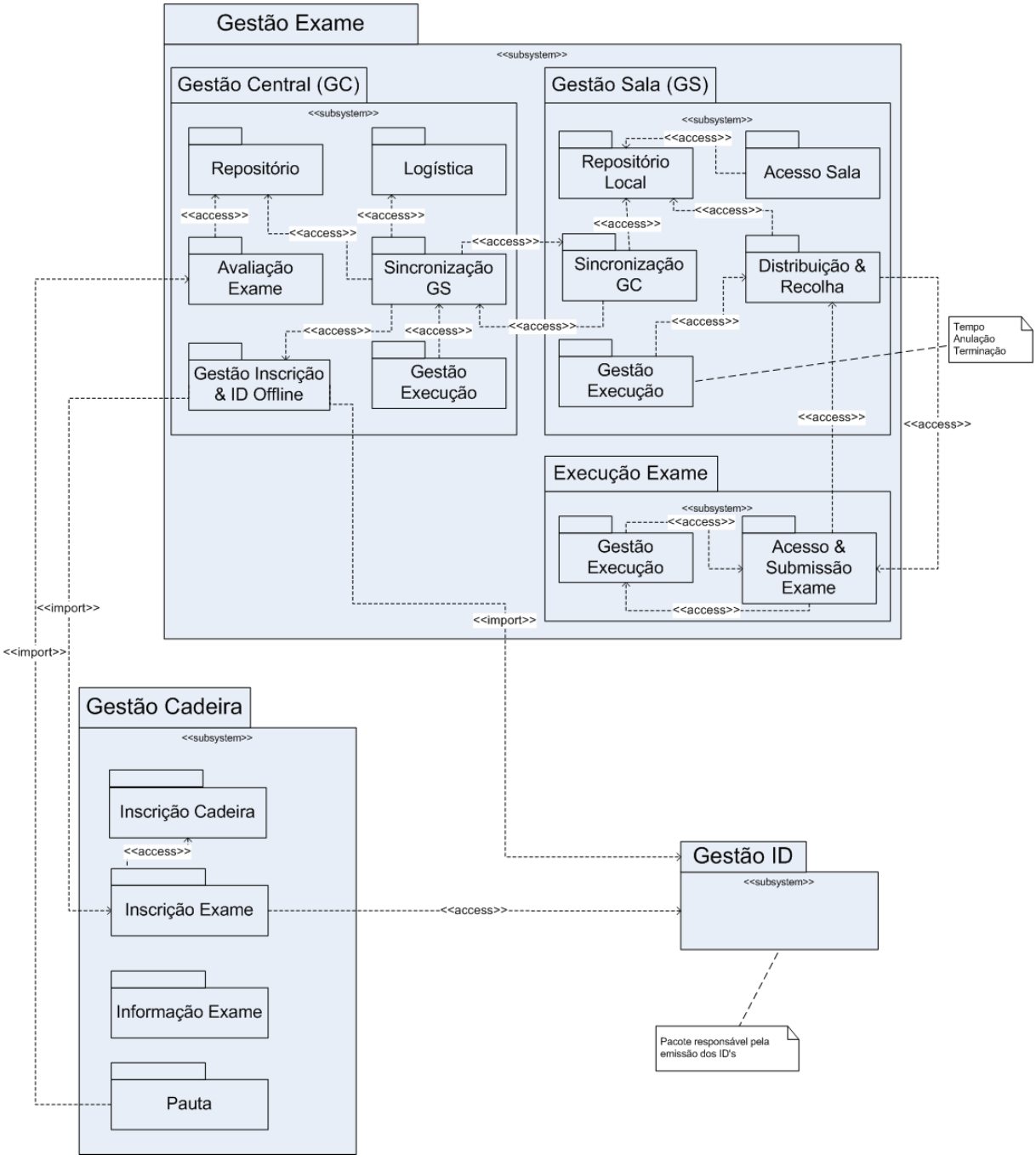


Figura 4.1 – Diagrama de módulos do sistema de Exames Electrónicos.

Gestão Exame

O módulo gestão exame é onde se encontra a principal funcionalidade da aplicação. Este é decomposto nos seguintes principais sub módulos:

- Gestão Central;
- Gestão Sala;
- Execução Exame.

Gestão Central (GC)

O módulo gestão central é onde se encontram as funcionalidades acessíveis tipicamente ao responsável pela cadeira/exame, necessárias antes da data do exame ou do início do exame, e ainda as acções efectuadas após a conclusão do exame.

Este é decomposto nos seguintes sub módulos:

- Repositório;
- Sincronização GS (Gestão Sala);
- Gestão Inscrição & ID Offline;
- Logística;
- Gestão Execução;
- Avaliação Exame.

Repositório

Módulo responsável pelo armazenamento dos exames e das respostas aos exames após a sua realização. Garante a integridade e autenticidade da informação guardada.

Sincronização GS

Módulo responsável pela distribuição e recolha dos dados necessários para a realização do exame pelas salas, nomeadamente os IDs dos alunos de cada sala, o exame e informação associada à segurança e ao exame.

Gestão Inscrição & ID Offline

Módulo responsável por permitir a execução do exame de modo normal independentemente de o sistema ter uma ligação on-line com o módulo Gestão ID. Este módulo carrega todos os dados necessários para a validação das identidades envolvidas na execução do exame (alunos, vigilantes, agentes autenticação, agentes autorização). Todos os dados acedidos/carregados por este módulo não lhe pertencem e não podem ser modificados.

Logística

Módulo que implementa as funcionalidades relacionadas com a logística necessária à resolução do exame, nomeadamente a distribuição de alunos por salas ou o carregamento dessa informação.

Gestão Execução

Módulo que implementa operações complementares à execução do exame, como por exemplo, monitorização da execução do exame, e outras funcionalidades de supervisão.

Avaliação Exame

Módulo responsável pela correcção dos exames e a respectiva atribuição de nota. A correcção pode ser automática ou não. Disponibiliza a avaliação final (a pauta) ao módulo Pauta (sub módulo de Gestão Cadeira) para a sua divulgação.

Gestão Sala (GS)

Módulo responsável pelas funcionalidades necessárias à execução do exame numa determinada sala.

Este é decomposto nos seguintes sub módulos:

- Distribuição & Recolha;
- Sincronização GC;
- Repositório Local;
- Gestão Execução;
- Acesso Sala.

Distribuição & Recolha

Módulo responsável pela gestão da distribuição e recolha dos exames pelos alunos numa determinada sala. A segurança é implementada e/ou verificada por este módulo: autenticação, assinaturas e chaves necessárias. Este módulo é quem sabe como comunicar com os alunos, em termos de segurança necessária para satisfazer pedidos.

Sincronização GC

Módulo responsável pela comunicação com o módulo Gestão Central, nomeadamente, acesso aos dados necessários para a realização do exame numa determinada sala e pela submissão das resoluções dos exames e a respectiva informação associada às resoluções dos exames.

Repositório Local

Módulo responsável pelo armazenamento dos exames, da resolução dos exames após a realização do exame e dos IDs de alunos numa determinada sala.

Gestão Execução

Módulo que implementa operações complementares à execução do exame. Tais como, ordem de início do exame, extensão do tempo de exame, anulação dum exame, finalização do exame.

Acesso Sala

Módulo que permite validar o acesso dum aluno a uma sala através do seu ID. Também responsável por controlar o acesso aos equipamentos.

Execução Exame (EE)

Módulo responsável pelas funcionalidades necessárias à execução do exame num dispositivo dum aluno. Este é decomposto nos seguintes sub módulos:

- Gestão Execução;
- Acesso/Submissão Exame.

Gestão Execução

Módulo onde se encontra toda a funcionalidade existente para a realização do exame por parte do aluno.

Acesso & Submissão Exame

Módulo responsável por comunicar com o módulo Gestão Sala, nomeadamente, o módulo Distribuição & Recolha, tem que saber satisfazer os requisitos de segurança impostos pelo módulo Distribuição & Recolha.

Gestão ID

Módulo associado a funções de secretaria, responsável pela gestão das identidades electrónicas (emissão, revogação, etc.), tanto de alunos como de professores e outros. Gerir as identidades electrónicas consiste também em gerir os certificados associados às identidades electrónicas. Este módulo irá aceitar pedidos de dados acerca das identidades electrónicas, apenas poderá satisfazer pedidos de dados públicos (isto pode mudar dependendo das condições físicas) e apenas em casos que os requisitos de segurança sejam satisfeitos (autenticação do docente que faz o pedido).

Neste módulo encontra-se a base de confiança do sistema. Existe a necessidade de caucionamento de chaves para permitir recuperar cartões perdidos.

Este módulo é o dono de todo o tipo de dados que contenham informação confiada sobre a identidade electrónica das entidades envolvidas na execução dum exame (alunos, professores, vigilantes).

Gestão Cadeira

O módulo gestão da cadeira é um módulo cujas responsabilidades se encontram actualmente satisfeitas pelo sistema Fénix [25]. Por este motivo não é neste módulo que se encontra o cerne deste trabalho, mas este módulo é necessário para descrever algumas funcionalidades adicionais (ao Fénix), tais como a comunicação com o módulo Gestão Exame. De qualquer modo terá que se implementar este módulo com as funcionalidades necessárias para demonstrar o funcionamento do módulo Gestão Exame.

4.2 Modelo Domínio

O modelo de domínio é definido com diferentes vistas para facilitar a compreensão do mesmo. Define-se o domínio apenas para os módulos principais da solução, os sub módulos do Gestão Exame, nomeadamente o Gestão Central, Gestão Sala e Execução Exame.

O domínio do GS e EE são sub domínios do GC, pois partilham o domínio, apresentando no entanto simplificações.

4.2.1 Gestão Central

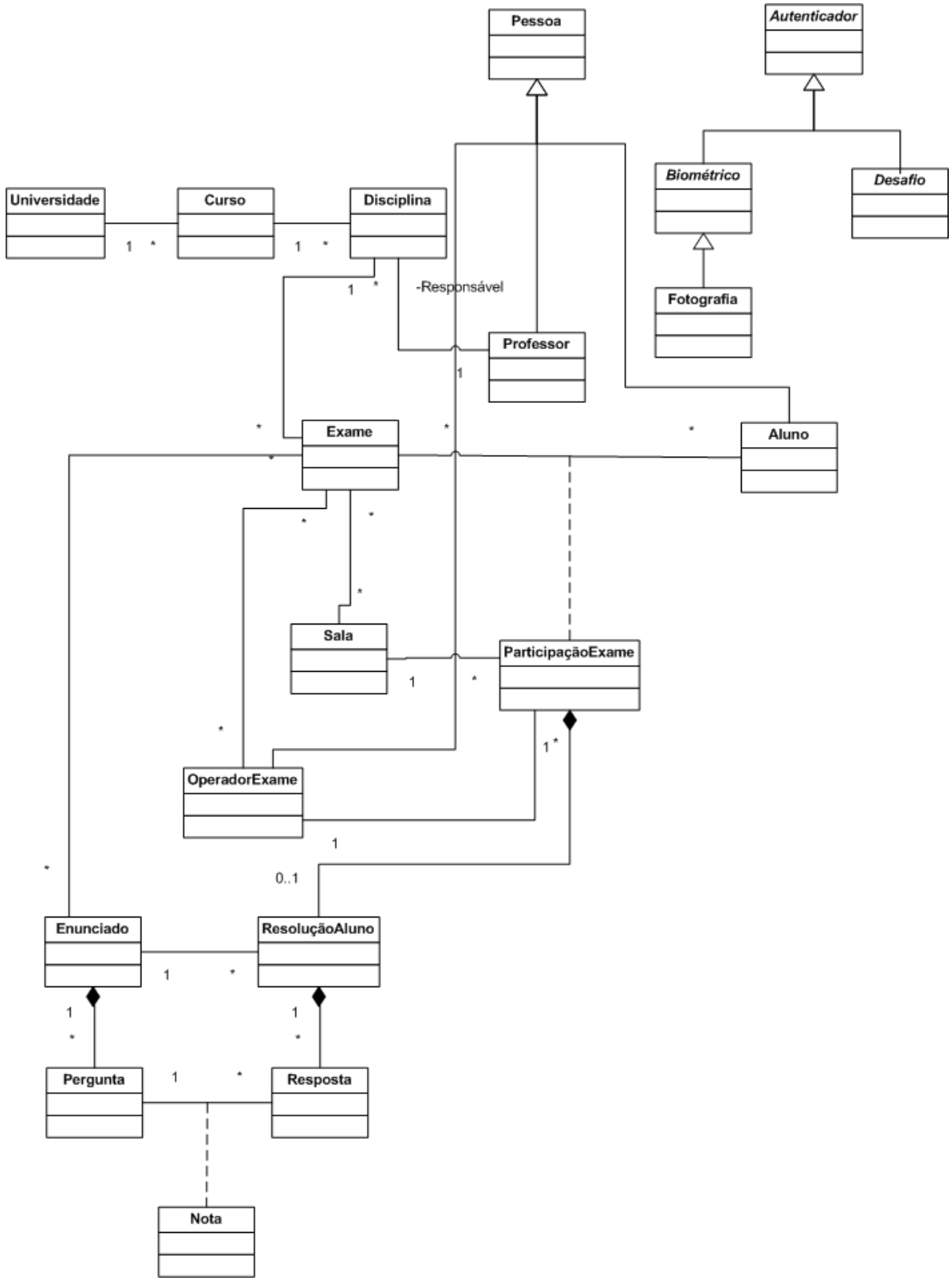


Figura 4.2 – Modelo Domínio GC.

4.2.1.1 Vistas

Neste ponto apresenta-se o domínio em diferentes vistas de modo a mostrar mais detalhe.

Modelo Centrado no Exame

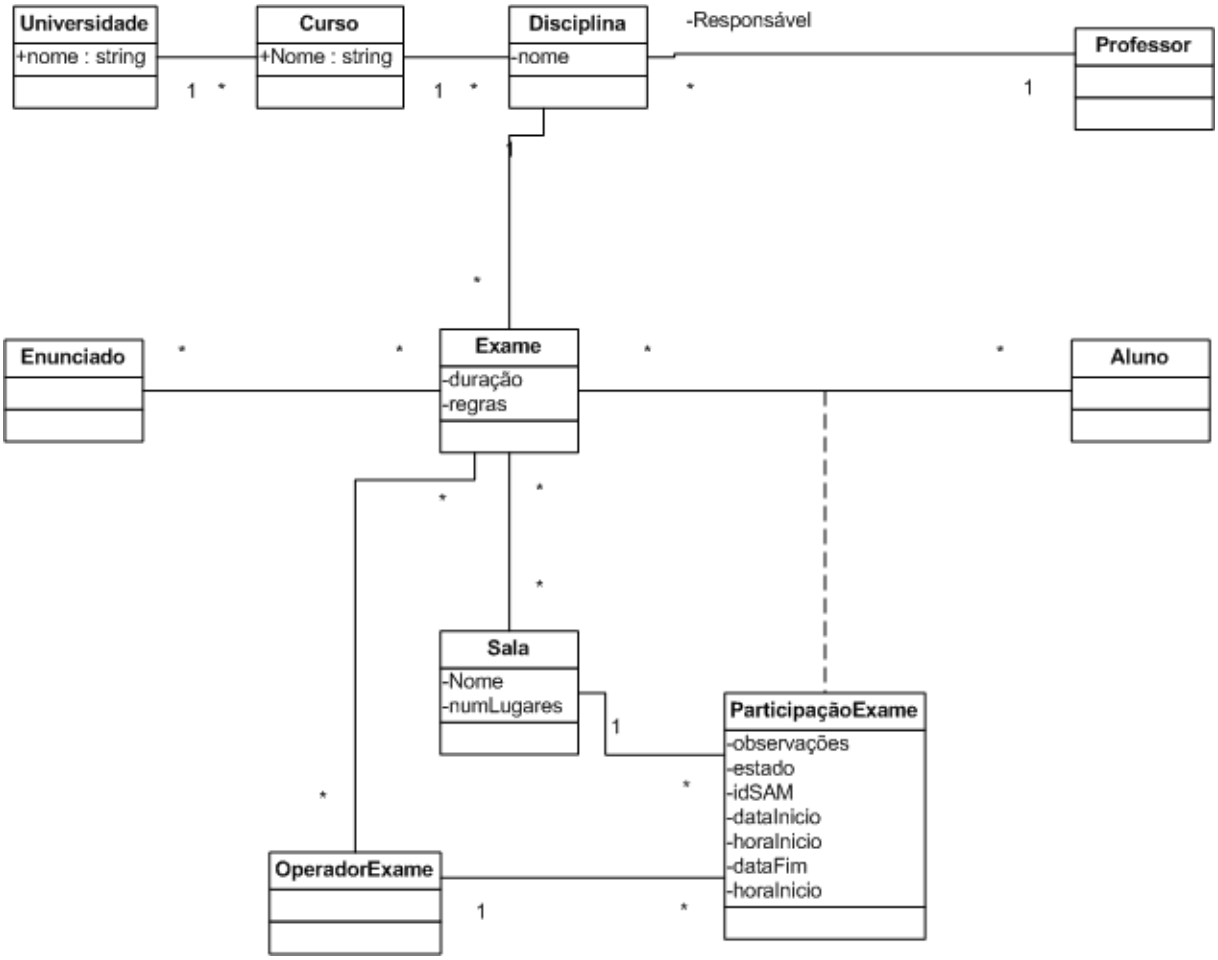


Figura 4.3 – Modelo Domínio centrado no Exame.

Modelo Centrado nos Participantes

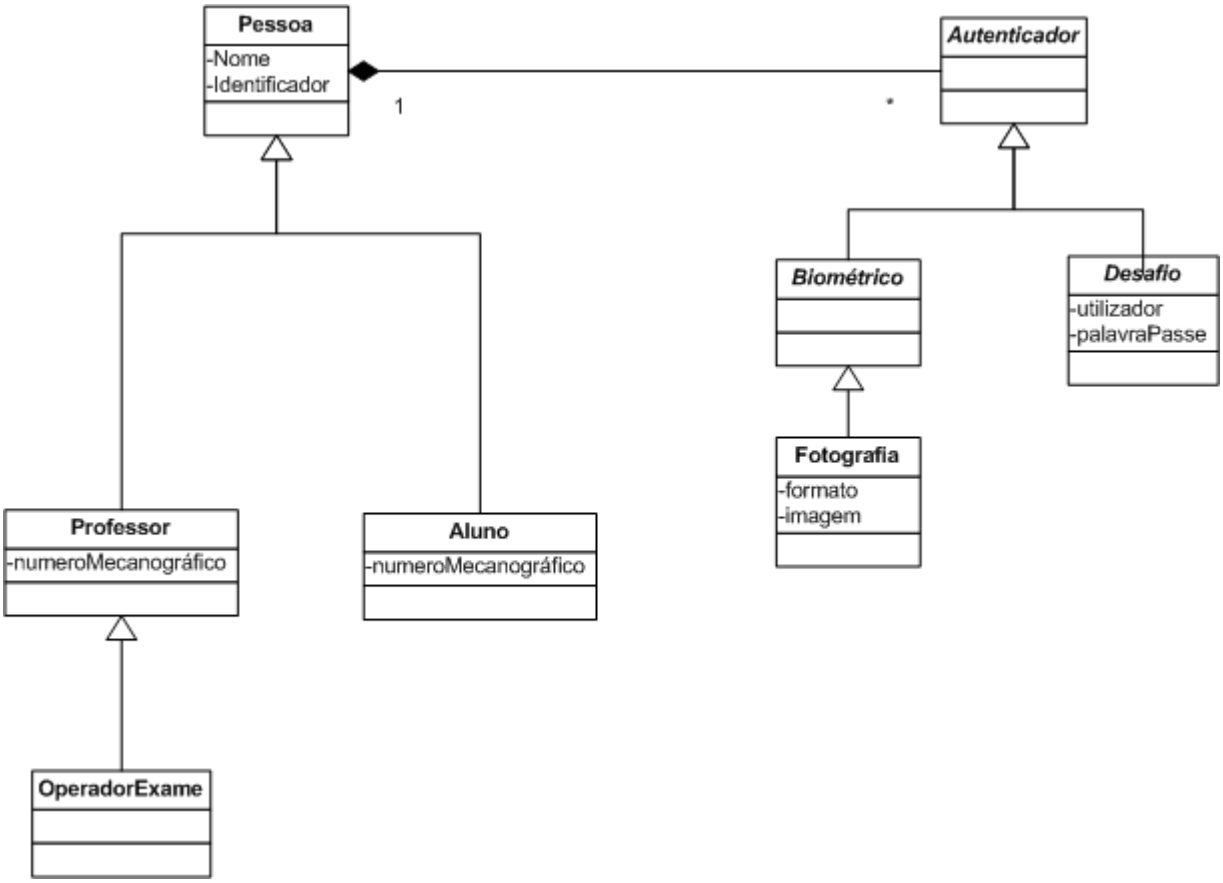


Figura 4.4 – Modelo Domínio centrado nos Participantes.

Modelo Centrado na Participação Exame

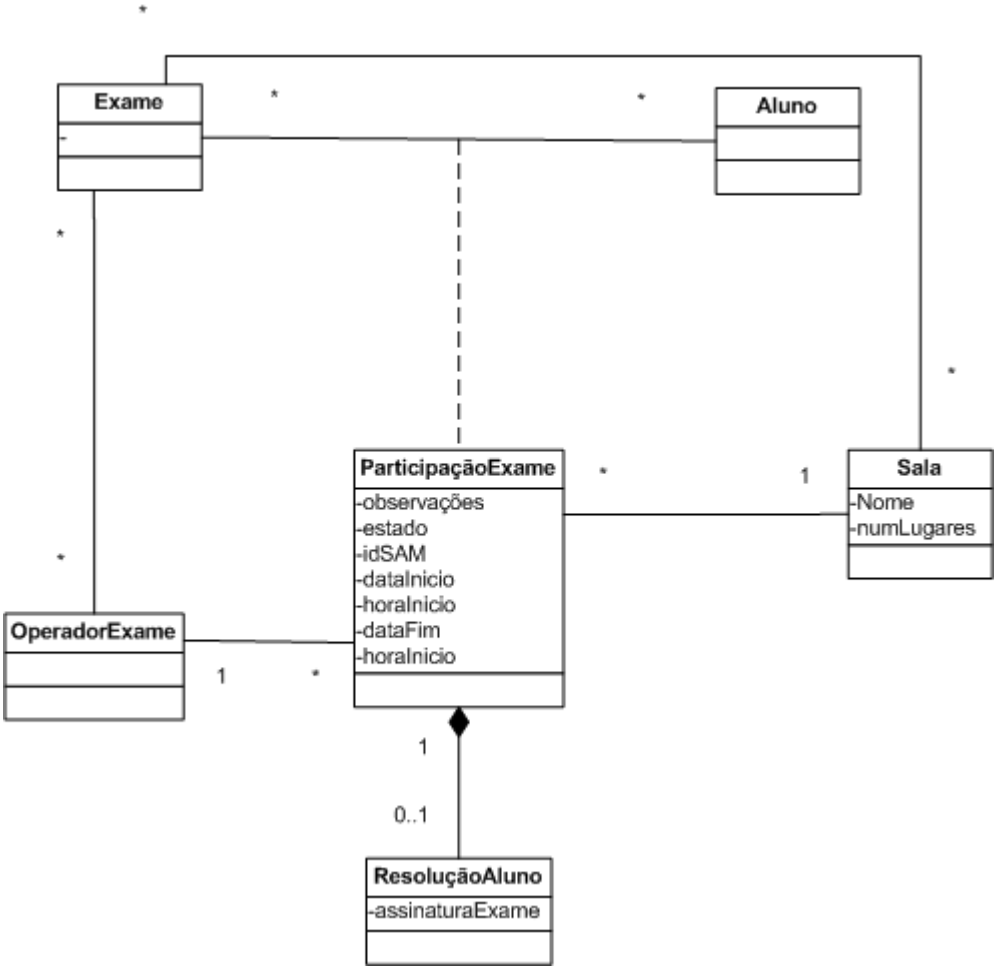


Figura 4.5 – Modelo Domínio centrado na Participação Exame.

Modelo Centrado na Resolução Aluno

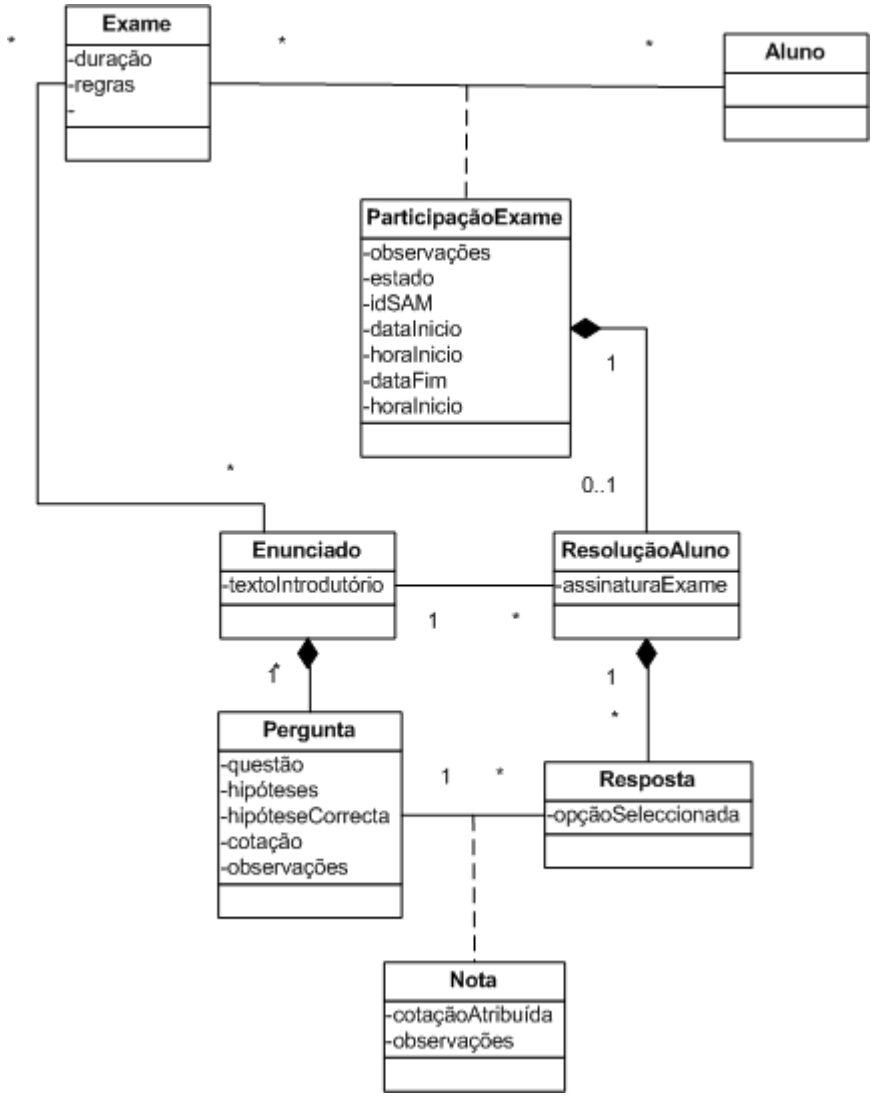


Figura 4.6 – Modelo Domínio centrado na Resolução Aluno.

4.2.2 Gestão Sala

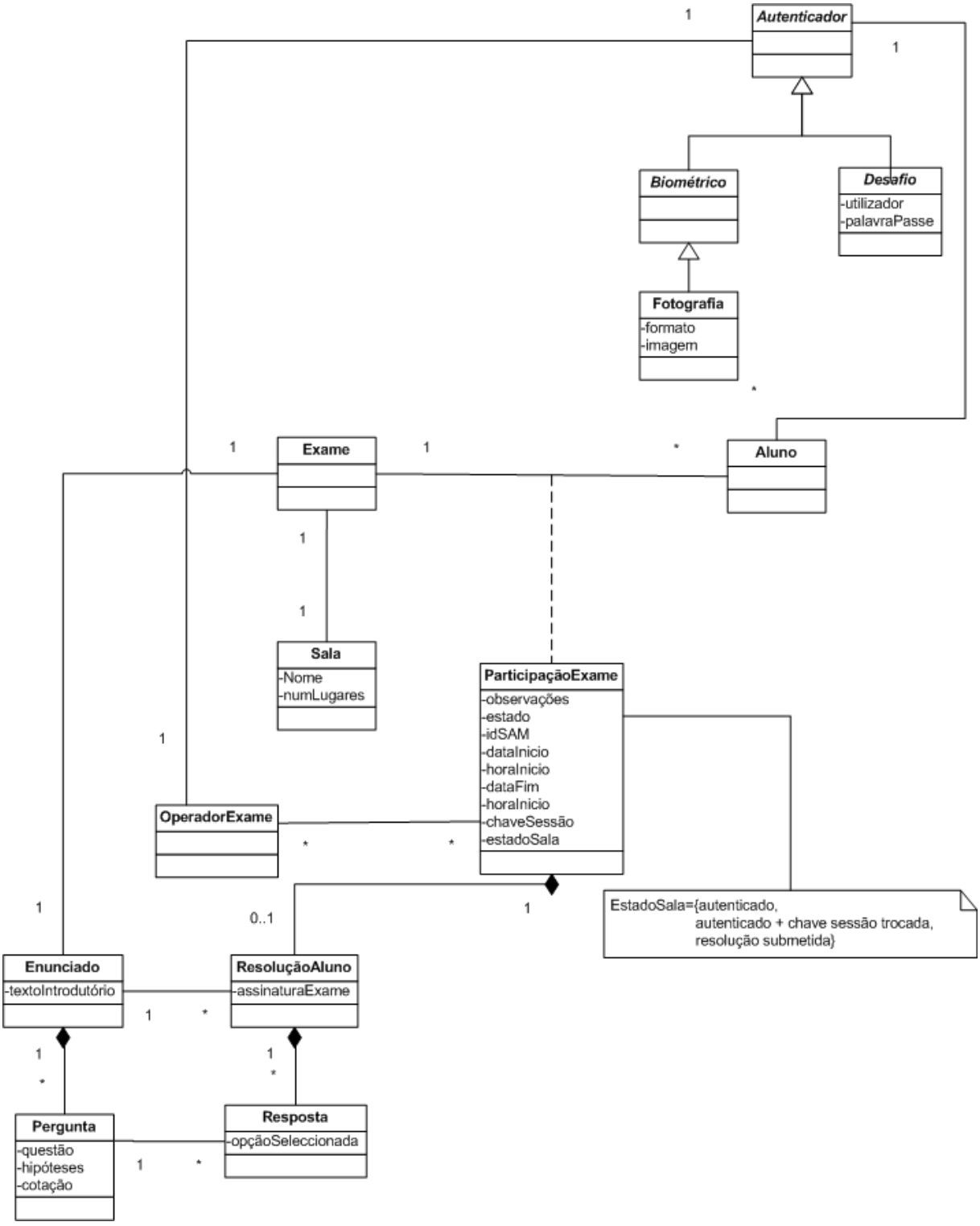


Figura 4.7 – Modelo Domínio GS.

4.2.3 Execução Exame

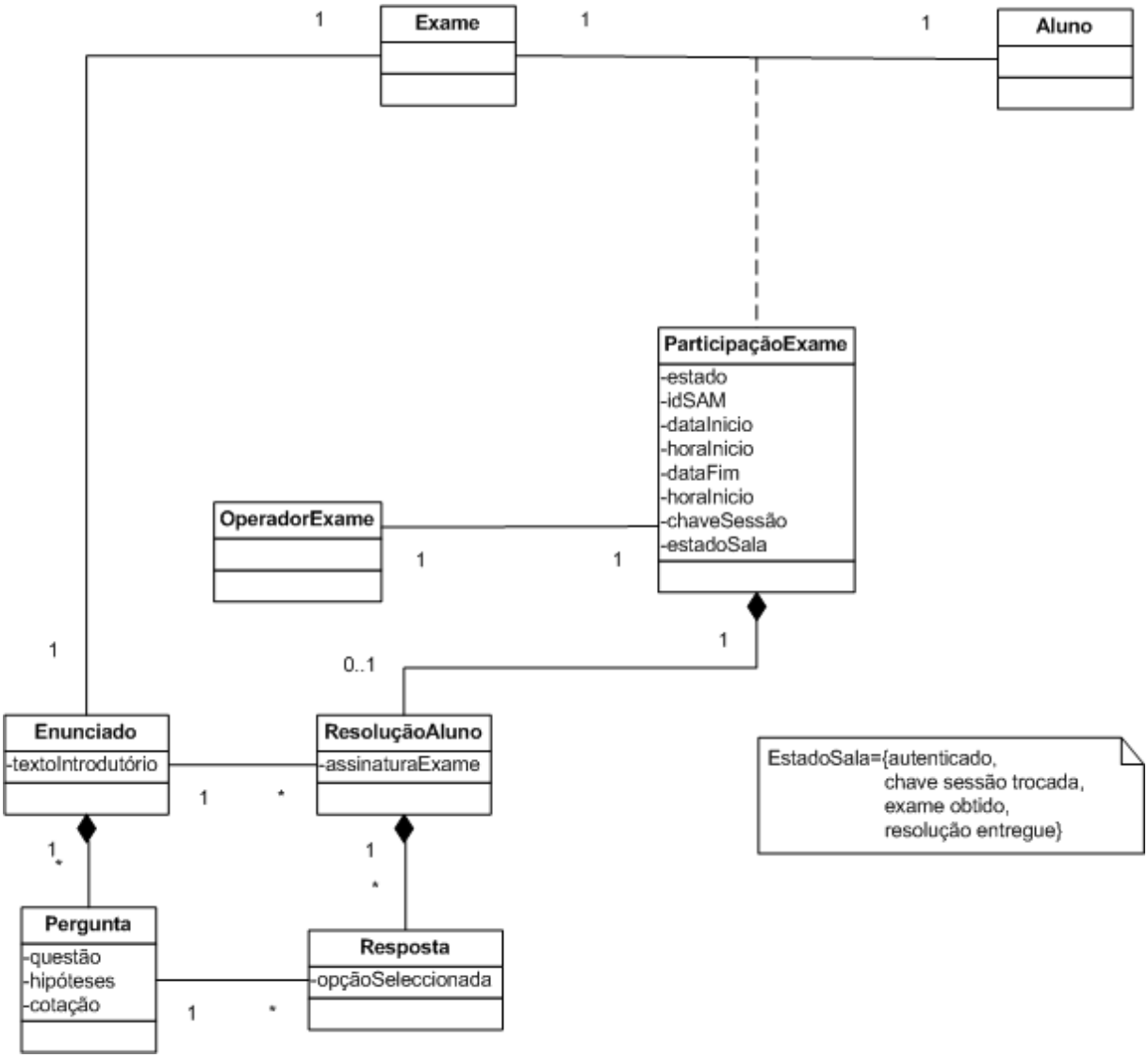


Figura 4.8 – Modelo Domínio EE.

4.3 Actividades Realização Exame

O diagrama seguinte apresenta dum modo global os passos (actividades) a executar no âmbito da realização dum exame genérico. Não são visíveis, neste diagrama, todas as operações que a solução suporta e nem todas as actividades são suportadas, no entanto o diagrama é um bom ponto de partida para a compreensão dos processos que decorrem no modelo descrito de realização de exames.

A implementação da solução foca-se nas interacções do GS com o EE, i.e., no momento de realização do exame por parte do aluno, onde a confiança do serviço é essencial para ambas as partes envolvidas.

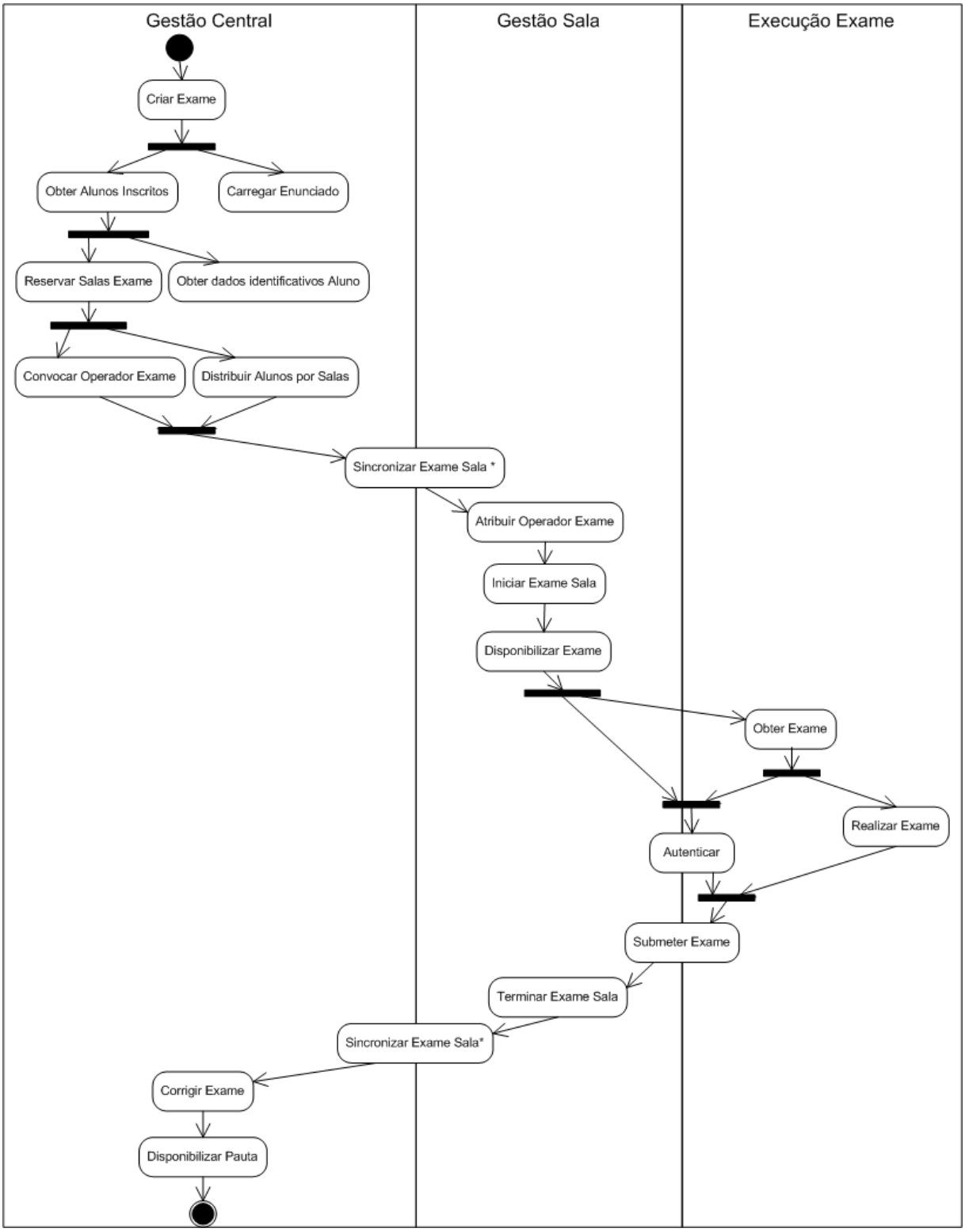


Figura 4.9 – Diagrama global de actividades.

4.4 Descrição Detalhada dos Casos de Uso

Neste ponto descrevem-se os casos de uso da aplicação, os casos de uso principais, tal como já referido, decorrem nas interacções entre os módulos GS e EE. Também são descritos os casos de uso entre os módulos GC e GS, designados casos de uso complementares.

4.4.1 Casos de Uso Principais, GS – EE

Neste ponto descrevem-se os casos de uso principais, que são totalmente suportados pela implementação do protótipo. Cada caso de uso é acompanhado de diagramas de sequência exemplificativos das interacções ocorridas.

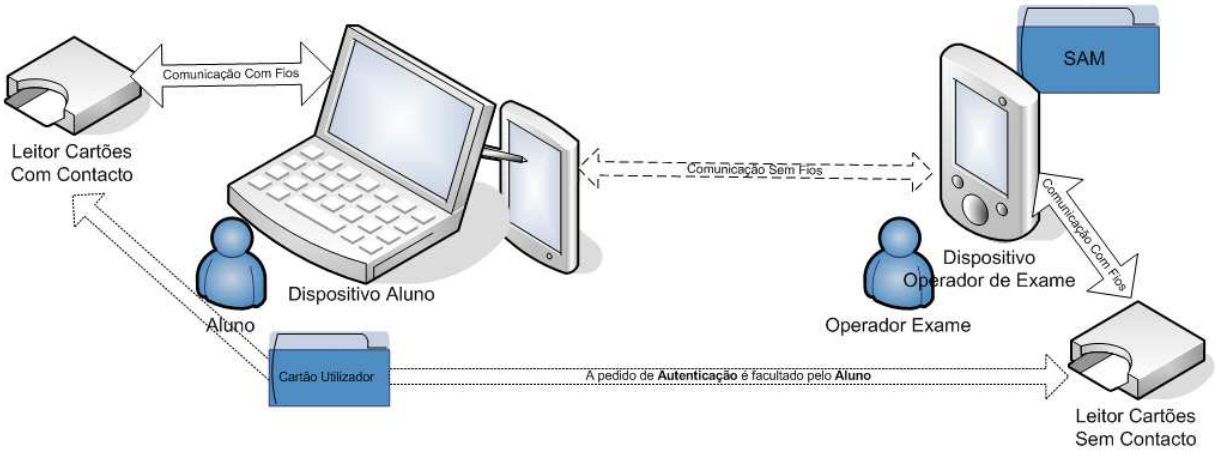


Figura 4.10 – Arquitectura de utilização da solução.

Autenticação e Estabelecimento de Chave de Sessão

1. Fornecimento de cartão utilizador por parte do aluno;
    a. O Aluno entrega o cartão directamente ao Operador de Exame, o dispositivo do Operador de Exame consegue comunicar com o cartão do Aluno;
2. Operador de exame obtém dados identificativos do Aluno através duma comunicação com o cartão de Aluno, de forma íntegra;
    a. Integridade do identificador de Aluno obtido, é garantida através das propriedades de segurança associadas à emissão do cartão de Aluno;

3. Garantir que o identificador de Aluno e outros dados relevantes são guardados para a posterior entrega da resolução do exame;
4. Operador de Exame consegue autenticar o aluno através do identificador de Aluno obtido;
    a. Através do identificador de Aluno obtido, acede a informação biométrica (fotografia) previamente carregada no PDA do operador de exame;
    b. Compara a informação biométrica carregada com o respectivo dado biométrico do Aluno (compara fotografia com a presença do Aluno);
5. Operador de Exame, através do seu dispositivo, escreve uma chave de sessão no cartão do Aluno, uma chave que está associada ao número de Aluno, para futuras interacções. A escrita da chave no cartão de Aluno deve ser íntegra e confidencial;
    a. Escrita directa no cartão do aluno, a integridade e confidencialidade estão garantidas pelo tecnologia do próprio cartão.

O diagrama de sequência simples apresenta a interacção do ponto de vista das entidades e objectos físicos que participam: o operador de exame, o aluno e o PDA. Os diagramas detalhados já mostram os módulos da aplicação.

Diagrama de sequência Autenticação Aluno (simples)

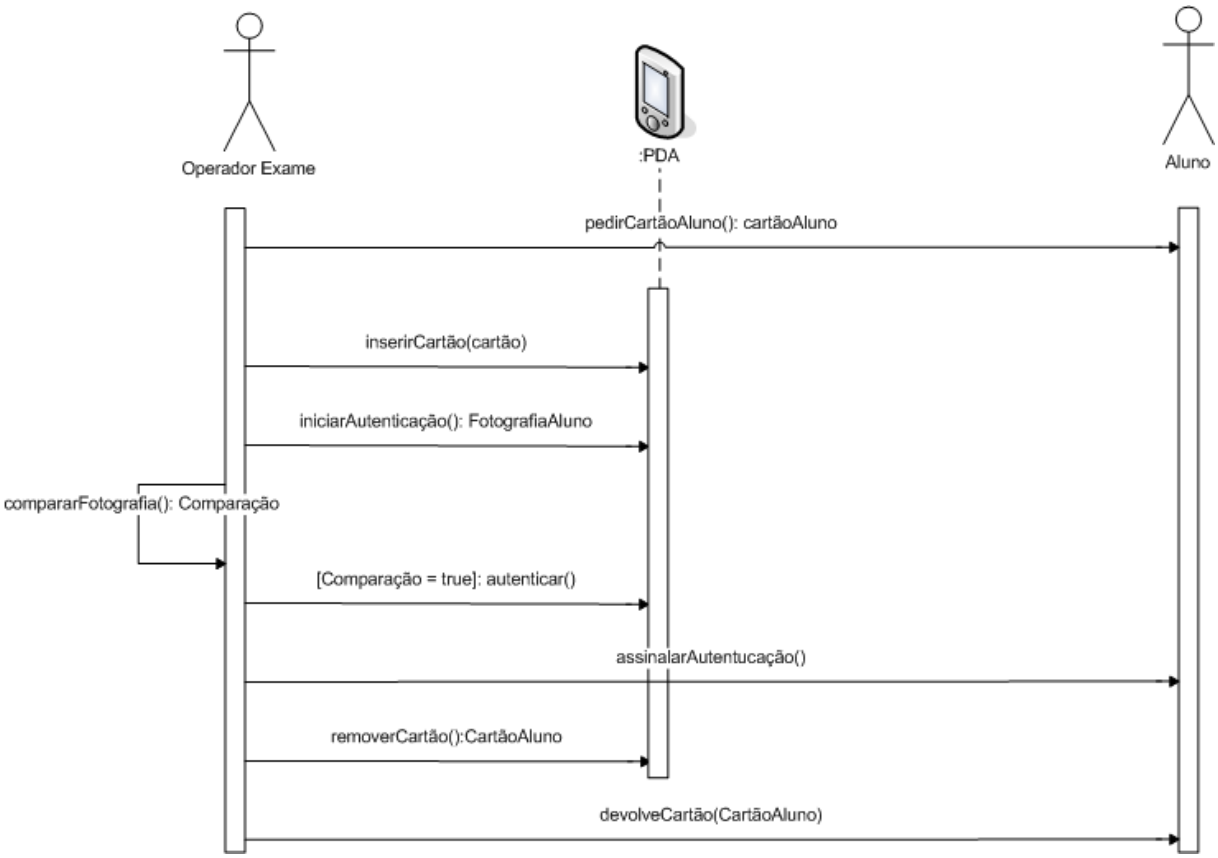


Figura 4.11 – Diagrama Sequência Autenticação Aluno (simples).

Diagrama de sequência Autenticação Aluno (detalhado)

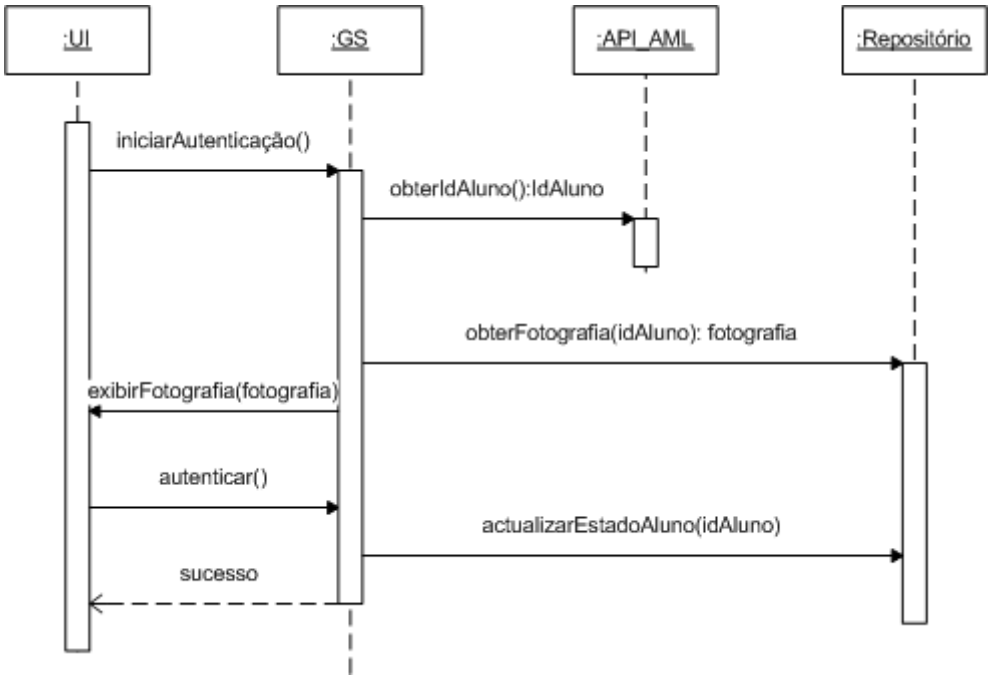


Figura 4.12 – Diagrama de sequência Autenticação Aluno (detalhado).

Diagrama de sequência Estabelecimento Chave Sessão (detalhado)

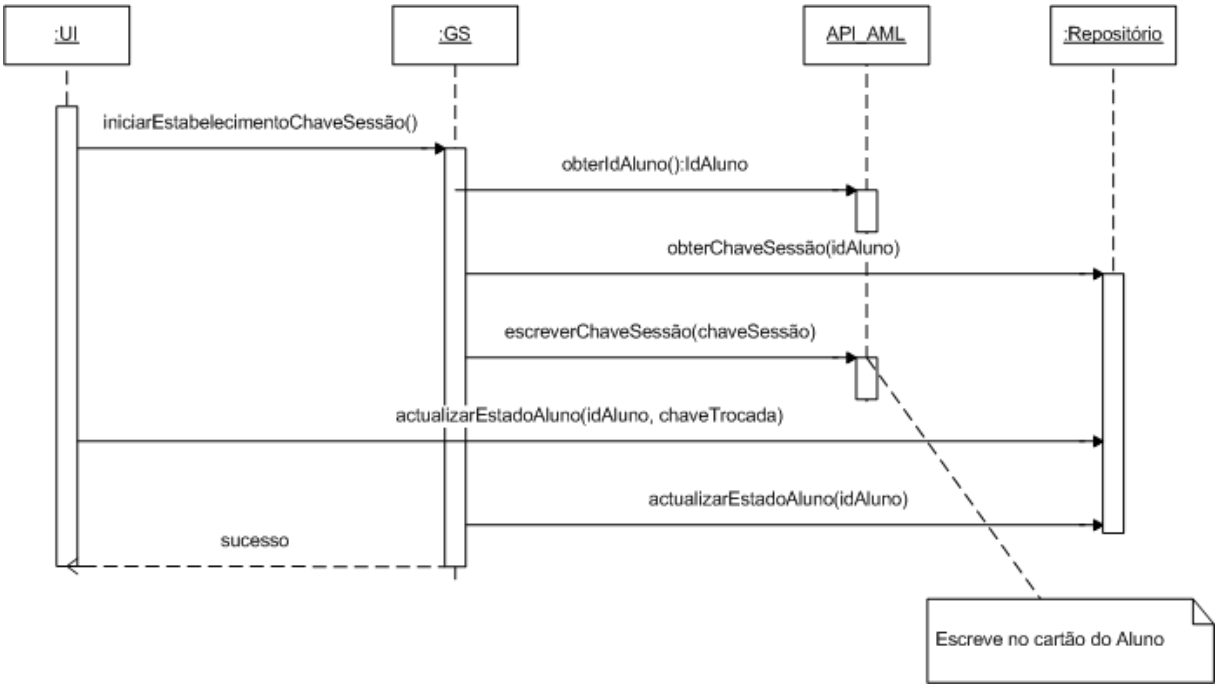


Figura 4.13 – Diagrama de sequência Estabelecimento de Chave de Sessão (detalhado).

Embora se tenha descrito nos diagramas acima apresentados a autenticação separada do estabelecimento da chave de sessão, estas duas operações são, tal como já descrito textualmente, efectuadas em simultâneo, isto apenas por conveniência, pois os passos iniciais são os mesmos e ambos os cenários necessitam da presença física do cartão do aluno junto do operador de exame.

Distribuição do Enunciado

1. O Operador de Exame, através do seu dispositivo envia por comunicação sem fios o enunciado do exame;
2. Enunciado passa em claro na comunicação, mas acompanhado duma assinatura através da chave de sessão previamente negociada;
3. O aluno, lê a chave de sessão previamente negociada do seu cartão utilizador, e valida a assinatura do enunciado do exame recebido.

Diagrama de sequência Distribuição Enunciado

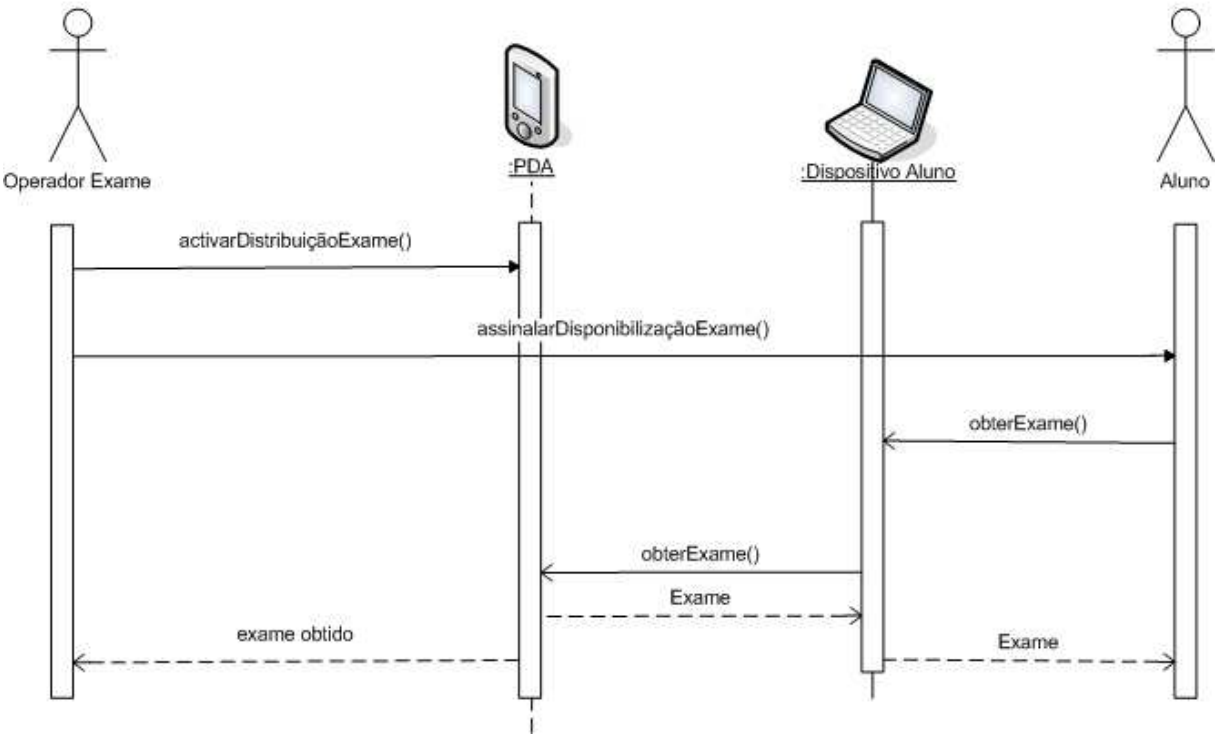


Figura 4.14 – Diagrama de sequência Distribuição Enunciado.

Entrega da Resolução do Exame

1. O Aluno, usa comunicação sem fios, para enviar a resolução do exame ao dispositivo do operador de exame;
2. A resolução do exame é fornecida num formato conhecido;
3. A resolução do exame é enviada para o dispositivo do Operador de Exame, cifrada com a chave de sessão negociada previamente, garantindo confidencialidade;
4. Em conjunto com a resolução do exame, vão dados identificativos do Aluno que submeteu a resolução, estes dados não podem ir cifrados para o Operador de Exame descobrir imediatamente a respectiva chave de sessão que lhe permite aceder à resolução do exame;
5. O operador de exame consegue através do número de Aluno decifrar a resolução do exame, e guarda a resolução associada com o número do Aluno;
6. O operador de exame, gera uma assinatura da resolução do exame, usando o SAM acoplado ao seu dispositivo. Esta assinatura é gerada com a chave do SAM, que não é passível de ser lida, nem modificada. O número do cartão de Aluno é usado como diversificante da assinatura;
7. A assinatura da resolução do exame gerada, é guardada pelo dispositivo do Operador de Exame, e é enviada ao dispositivo do Aluno, por sua vez também assinada com a chave de sessão negociada previamente (para validar que foi gerado pelo professor), servindo para efeitos de prova de entrega do exame.

Diagrama de sequência Entrega Resolução Exame (simples)

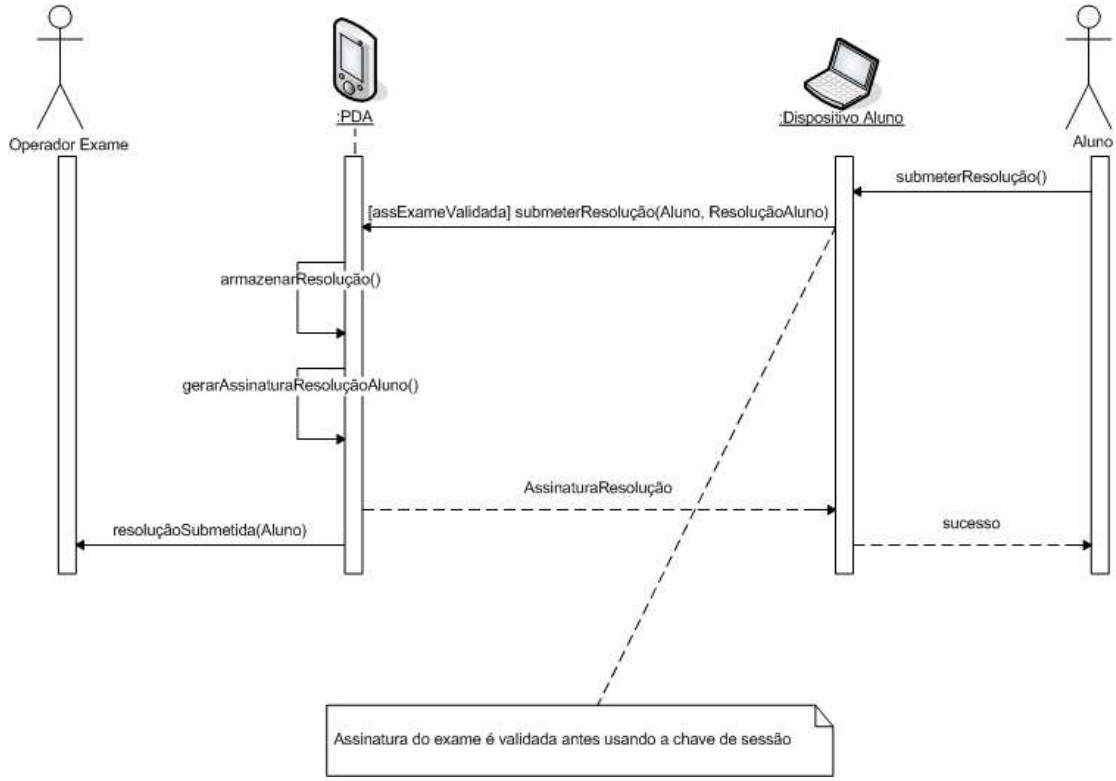


Figura 4.15 – Diagrama de sequência Entrega Resolução Exame.

Diagrama de sequência Entrega Resolução Exame (detalhado)

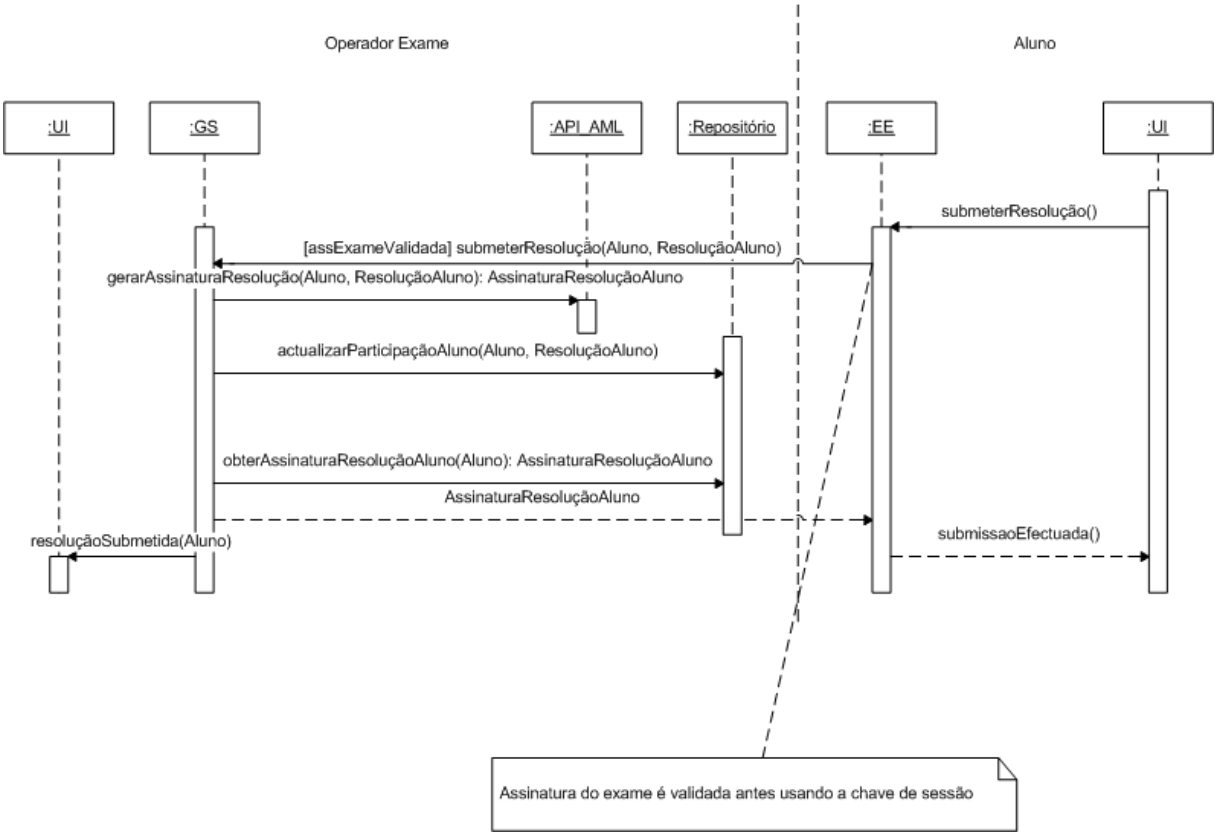


Figura 4.16 – Diagrama de sequência Entrega Resolução Exame (detalhado).

4.4.2 Casos de Uso Complementares, GC – GS

Neste ponto descrevem-se textualmente os casos de uso complementares.

Casos de Uso Pré Exame[7]

1. Inserção do Enunciado do Exame no sistema por parte do docente responsável;
2. Obtenção de informação sobre quais os alunos inscritos no exame, na cadeira em questão. Esta informação poderia ser obtida do sistema Fénix;
3. Obtenção dos dados pessoais dos alunos inscritos no exame necessários para a realização: dado biométrico (fotografia), número, nome;

[7] Acontecem antes da realização do exame

4. Obter informação sobre as salas disponíveis para a realização do exame;
5. Efectuar o controlo logístico do exame:
    a. Associar os alunos às salas disponíveis para o exame;
    b. Associar os docentes convocados para participação no exame.
6. Distribuir por cada dispositivo de Operadores de Exame, em função da sala, a informação necessária à realização do exame:
    a. Enunciado do Exame;
    b. Informação associada à sala em que o Operador de Exame irá realizar o exame;
        i. Quais os alunos que efectuam o exame na sala em questão;
        ii. Dados biométricos dos alunos, para autenticação dos mesmos (fotografia);
    c. Pré geração de chaves e associação a alunos.

Casos de Uso Pós Exame[8]

1. Recolher de cada dispositivo de Operador de Exame, a informação gerada em cada sala:
    a. Todas as resoluções, em que cada uma possui associada a identificação do Aluno, o MAC gerado pela entrega;
    b. Associar às resoluções o Operador de Exame responsável pela recolha dos exames, e quais os SAM envolvidos em cada sala (para efeitos de controlo de geração de MACs);
2. Tratar esta informação e armazená-la de modo persistente;
3. Validar se está tudo correcto e completo;
4. Avaliar as provas (este processo pode ser automático em alguns casos);
5. Disponibilização das pautas ao sistema Fénix, para disponibilização ao "público".

[8] Acontecem após a realização do exame

4.5 Arquitectura Tecnológica

A arquitectura tecnológica da aplicação é apresentada na figura seguinte. É possível observar uma estrutura tecnológica bastante rica e organizada, sendo bastante semelhante em ambas as aplicações, GS e EE, embora a aplicação GS seja um pouco mais complexa, pois tem requisitos superiores em termos de funcionalidade e em termos de persistência.

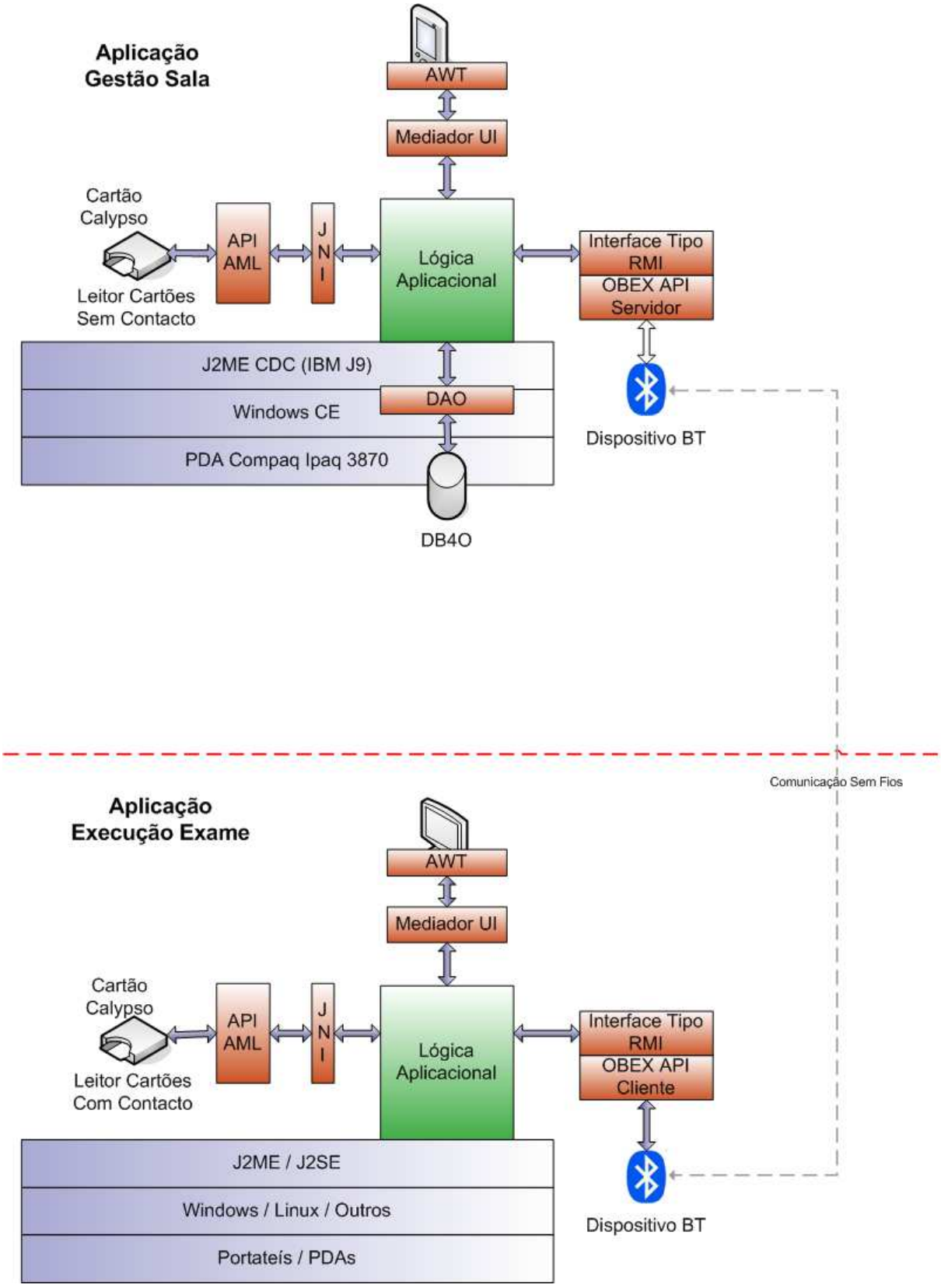


Figura 4.17 – Arquitectura Tecnológica das aplicações GS e EE.

Aplicação Gestão Sala

A lógica da aplicação disponibiliza e usa as seguintes interfaces: comunicação, *Bluetooth* e Cartões Inteligentes *Calypso*; persistência; e controlo de apresentação. É a lógica da aplicação que, através duma metodologia de serviços, contém a inteligência da aplicação, podendo estes ser desencadeados pelo controlo da apresentação, ou por pedidos *Bluetooth*, decorrendo num ambiente multi-tarefa.

Aplicação Execução Exame

Esta aplicação não necessita dum ambiente multi-tarefa, pois todas as acções são desencadeadas pelo mesmo utilizador. As acções são direccionadas para o serviço correcto na lógica da aplicação pelo controlo da apresentação, os serviços podem originar pedidos *Bluetooth* à aplicação GS, escritas ou leituras dos Cartões Inteligentes *Calypso*, ou apenas serviços simples, i.e., sem comunicações externas. Esta aplicação não tem persistência por simplificação, mas a funcionalidade principal, senão a única, da persistência nesta aplicação seria permitir à aplicação poder ser reiniciada com o mesmo estado.

4.5.1 Arquitectura Calypso

4.5.1.1 Cartões Inteligentes

Um Cartão Inteligente é um cartão de plástico com um *chip* embutido. Nesse *chip* existe um microprocessador, e alguns tipos de memória volátil e não volátil.

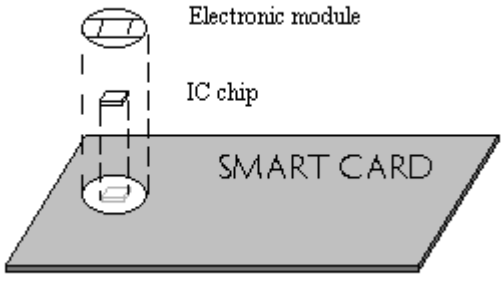


Figura 4.18 – Estrutura dum cartão inteligente (extraído de [7]).

Os cartões inteligentes estão divididos em grupos por:

- Função:
    - Cartões de memória;
    - Cartões com microprocessador.
- Interface Física / Mecânica:
    - Cartões com contacto;
    - Cartões sem contacto.

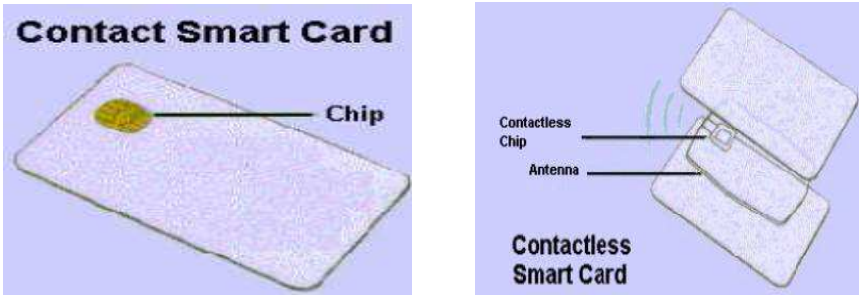


Figura 4.19 – Cartão Com Contacto e Sem Contacto (extraído de [7]).

Alguns cartões estão munidos dos dois mecanismos de acesso, podendo ser lidos com leitores para cartões com e sem contacto, como é o caso do cartão *Calypso* (também designado cartão Lisboa Viva devido à sua aplicação mais divulgada entre nós [26]).

4.5.1.2 Arquitectura Segurança

A arquitectura *Calypso* foi desenhada para o modelo dos transportes, onde está efectivamente a ser usada. A arquitectura de segurança é baseada numa hierarquia de cartões com diferentes permissões, sendo diferentes chaves criptográficas atribuídas a cada nível.

Existem dois tipos de cartões, o cartão utilizador (GTML) e o cartão SAM (*Secure Access Mode*) e ambos armazenam chaves criptográficas de modo seguro. O último implementa a hierarquia, já referida, que designa diferentes funcionalidades a cada nível. Existem 4 tipos de SAM, para cada um deles está bem definido qual o seu papel no sistema com base nas permissões de escrita e leitura nos outros cartões:

- SAM-CV (SAM_VALID) é normalmente usado nos equipamentos de validação e fiscalização. É o que tem menores privilégios, não permitindo carregar valor, apenas descontar;
- SAM-CL (SAM_RELOAD) é normalmente usado nos equipamentos de venda/carregamento. Para além dos privilégios do SAM-CV, permite carregar valor;
- SAM-CP (SAM_ISSUER) é normalmente usado nos equipamentos de personalização eléctrica e gráfica dos cartões. É o que tem maiores privilégios, para além dos de SAM-CV e SAM-CL, permite escrever na área do cartão reservada a área de dados pessoais (por exemplo, perfis do cliente) e outras informações importantes da rede de transporte;
- SAM-CPP (SAM_PROD) é um SAM usado nos equipamentos de iniciação e pré-personalização dos cartões. Este SAM é usado pelos fornecedores de cartões, para carregar as chaves nos cartões. Este é na realidade o SAM com maiores privilégios. Ele normalmente está na posse da entidade reguladora do serviço, sendo cedido temporariamente aos fornecedores de cartões quando há necessidade de encomendar cartões. Este SAM tem um mecanismo que permite auditar o número de cartões emitidos para evitar que os fornecedores produzam mais cartões do que os encomendados.

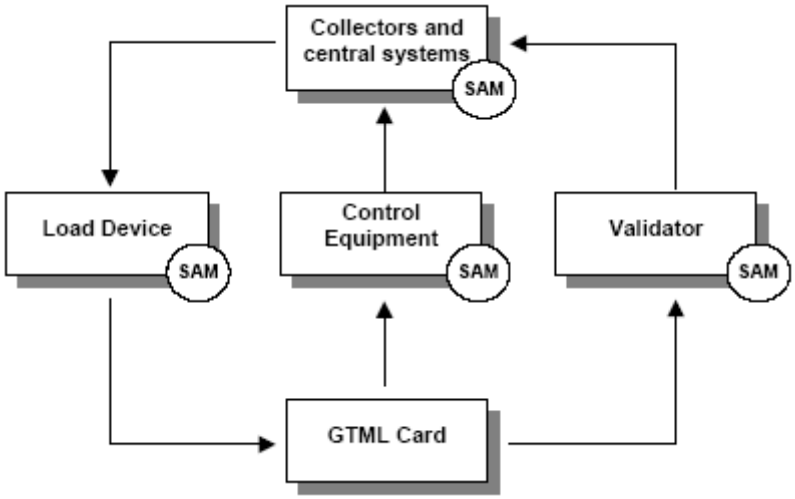


Figura 4.20 – Esquema interacções no modelo dos transportes[9].

Protocolo de sessão segura

Todas as interacções que envolvam escritas são feitas no âmbito duma sessão segura. A sessão segura garante:

- Autenticação do cartão participante;
- Autenticação do terminal participante;
- Integridade e autenticidade dos dados trocados.

A sessão é também transaccional, na medida em que as alterações ao cartão apenas se tornam efectivas se a sessão for fechada com sucesso.

É garantida a autenticação mútua através da geração e verificação dum MAC (*Message Authentication Code*) gerado no âmbito de cada sessão segura. O terminal (SAM) envia parte do MAC (os 4 bytes mais significativos) e o cartão, se o MAC recebido estiver correcto, envia para o terminal os 4 bytes menos significativos.

Durante a sessão, todos os comandos e dados são tidos em conta na assinatura (no MAC) final de 8 bytes.

[9] As setas na figura representam fluxos de informação.

Reload Ceilings

Para a realização de operações de controlo do uso dos cartões, por parte do operador de exame, realizaram-se testes para validar que tipo de controlo é possível efectuar através dos mecanismos disponibilizados pela arquitectura dos cartões *Calypso*.

Na arquitectura *Calypso* existe, tal como já foi referido, uma hierarquia de cartões, em que cada um tem o seu conjunto de funcionalidades e permissões associadas.

Os cartões SAM de *Reload*, tem as chaves necessárias para ler e escrever nos cartões utilizador, estas chaves tem associadas um contador, que por sua vez tem associado um *ceiling* que limita o número de operações possíveis de efectuar que façam uso da chave. Este mecanismo além de permitir um controlo *offline* do número de operações efectuadas por um operador, permite limitar à partida o número e tipo de operações efectuadas fazendo uso dum cartão SAM de *Reload*. Estas permissões, como é natural, são carregadas por SAMs com permissões superiores ao de *Reload*, o *Issuer*.

4.5.1.3 Arquitectura *Calypso* nos Exames Electrónicos

A reutilização desta tecnologia serviu para garantir as propriedades de segurança que se exigiam ao modelo de execução dos exames electrónicos. O cartão utilizador foi adaptado para o cartão de identificação do aluno, pois este está preparado para receber informação de personalização. As garantias de escrita íntegra e autêntica no cartão (usando o mecanismo já descrito das sessões seguras), permitem usar o cartão como elemento de armazenamento de dados autenticadores (a informação de personalização) e para troca de dados sensíveis durante a realização dum exame (a chave de sessão).

O uso das características de segurança da arquitectura *Calypso* permitiu garantir uma confiança acrescida ao serviço de realização de exames.

4.5.2 API AML

A API AML foi desenvolvida, pela *Link Consulting*, no âmbito do projecto Lisboa Viva. Trata-se de uma aplicação modular concebida para poder ser portável para qualquer sistema. A aplicação disponibiliza uma interface para interagir com cartões inteligentes através de leitores com e sem contacto.

A aplicação está dividida em várias camadas, sendo que a camada de baixo nível se refere à comunicação com os leitores. Para cada tipo de leitor (com ou sem contacto) existe um módulo próprio designado como *coupler* que implementa a interface disponibilizada.

Esta aplicação é uma peça fundamental no nosso trabalho, pois através desta conseguimos interagir com os cartões inteligentes para realizar todas as operações necessárias.

4.5.3 Java Native Interface

A interface nativa do Java (JNI), é uma ferramenta da plataforma Java que permite a integração da linguagem Java com outras linguagens, tal como C ou C++. Permite a aplicações escritas em Java a utilização de código nativo doutras linguagens de programação. Permite também a utilização de código Java noutras linguagens de programação, i.e., o inverso do descrito anteriormente.

Por fazer parte da plataforma Java, problemas como a heterogeneidade de plataformas estão tratados, libertando o programador da tarefa de resolver os eventuais problemas causados pela heterogeneidade.

A JNI pode ser usada para escrever métodos nativos (implementados na linguagem de programação nativa) que, permitem às aplicações Java chamar funções implementadas em bibliotecas nativas. As aplicações Java chamam os métodos nativos da mesma forma que o fazem em métodos implementados na linguagem Java.

A JNI disponibiliza às aplicações nativas a utilização da máquina virtual Java (JVM), pelo que estas podem usar uma biblioteca nativa que implementa a JVM, e através dela podem usar software escrito na linguagem de programação Java.

A JNI foi usada no projecto para aceder à API AML através da linguagem Java. Já existia um *wrapper* Java com JNI que permitia invocar a API AML, mas este tinha alguns problemas relacionados com versões diferentes da API AML a usar. Por este motivo houve necessidade de efectuar alterações e de adicionar funcionalidades ao *wrapper*.

4.5.4 J2ME (IBM J9)

4.5.4.1 Plataforma J2ME

A plataforma J2ME [8], uma componente mais leve do Java, foi expressamente criada para satisfazer as necessidades que o desenvolvimento de software para dispositivos mais limitados impõe. Mas, mesmo dentro desta categoria continuam a existir dispositivos com grandes diferenças entre si, por isso, esta plataforma não é uma especificação, é um conjunto de especificações que se dividem em:

- Configurações;
- Perfis;
- Pacotes opcionais.

Uma configuração define uma plataforma Java para um segmento horizontal de dispositivos com capacidades semelhantes a nível de processamento e memória. Cria um ambiente base de desenvolvimento e de execução através da definição de um conjunto de bibliotecas e capacidades da máquina virtual. A noção de ambiente base existe porque as configurações não devem cobrir as especificidades dos dispositivos para que possam ser reutilizadas em muitas áreas.

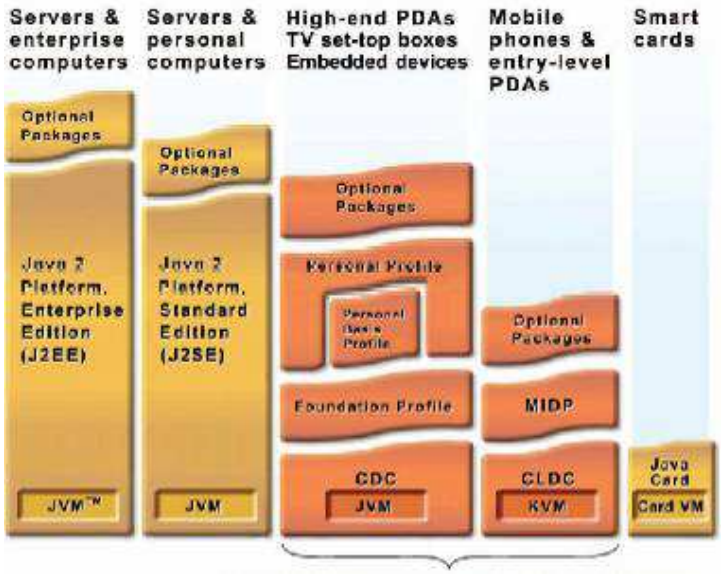


Figura 4.21 – Máquinas Virtuais Java, configurações da plataforma J2ME.

Existem duas configurações em J2ME:

- *Connected Limited Device Configuration* (CLDC) – ambiente baseado numa máquina virtual com capacidades muito reduzidas, dirigido para dispositivos com poucas capacidades, telemóveis, p*agers*, PDAs (gama baixa);
- *Connected Device Configuration* (CDC) – ambiente baseado na máquina virtual Java clássica e que define um ambiente similar ao de um PC de secretária, dirigido para dispositivos mais poderosos como s*et-top boxes*, *gateways*, PDAs (gama alta), etc.

Como as configurações tentam definir um ambiente Java para uma gama de dispositivos o mais abrangente possível, as especificidades dos grupos de dispositivos de uma destas famílias passam a ser tratadas por perfis. Um perfil define um conjunto de APIs que satisfazem as necessidades de um segmento vertical dentro de uma configuração.

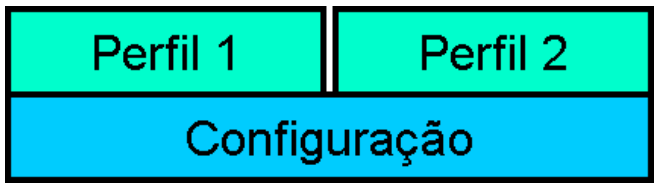


Figura 4.22 – Perfis sobre configurações na plataforma J2ME.

Pacotes opcionais, são como o nome indica, funcionalidades acrescidas que um determinado vendedor poderá incorporar no seu equipamento, servindo para colmatar lacunas de funcionalidade específicas.

4.5.4.2 IBM J9

A J9 [9] é uma máquina virtual propriedade da IBM, que implementa a configuração CDC e alguns perfis. Esta foi a máquina virtual (VM) utilizada na aplicação GS, com a configuração *Personal Profile*, pois é uma configuração bastante completa e rica em funcionalidades, quase que permitindo uma total abstracção das limitações do dispositivo em uso.

A escolha desta configuração deveu-se sobretudo a:

- Necessidade de existência de bibliotecas gráficas;
- Complexidade da aplicação a desenvolver;

- Suporte por parte da VM à plataforma de hardware disponível (PDA *Compaq Ipaq 3870*);
- Plataforma de hardware com recursos suficientes para suportar a VM;
- Ferramenta reconhecida no mercado dos dispositivos móveis, com suporte de implementação completo;

O ambiente de desenvolvimento, ainda se acresce de alguns pacotes opcionais para suportar a persistência, as comunicações por Bluetooth, as comunicações com os cartões inteligentes, a escrita e leitura de ficheiros XML e o uso de algoritmos criptográficos.

4.5.5 Windows Mobile

É um sistema operativo (SO) [10] desenhado para computadores com capacidades reduzidas, tipicamente, PDAs. A versão 2002 foi usada no projecto como SO de suporte, pois é usado pelo *Compaq Ipaq 3870*, dispositivo onde corre a aplicação GS.

4.5.6 Abstract Window Toolkit

O *Abstract Window Toolkit* (AWT) [11] é uma biblioteca da *Sun* para criar interfaces gráficas. Esta providencia um conjunto de classes, interfaces e excepções que permitem criar objectos gráficos.

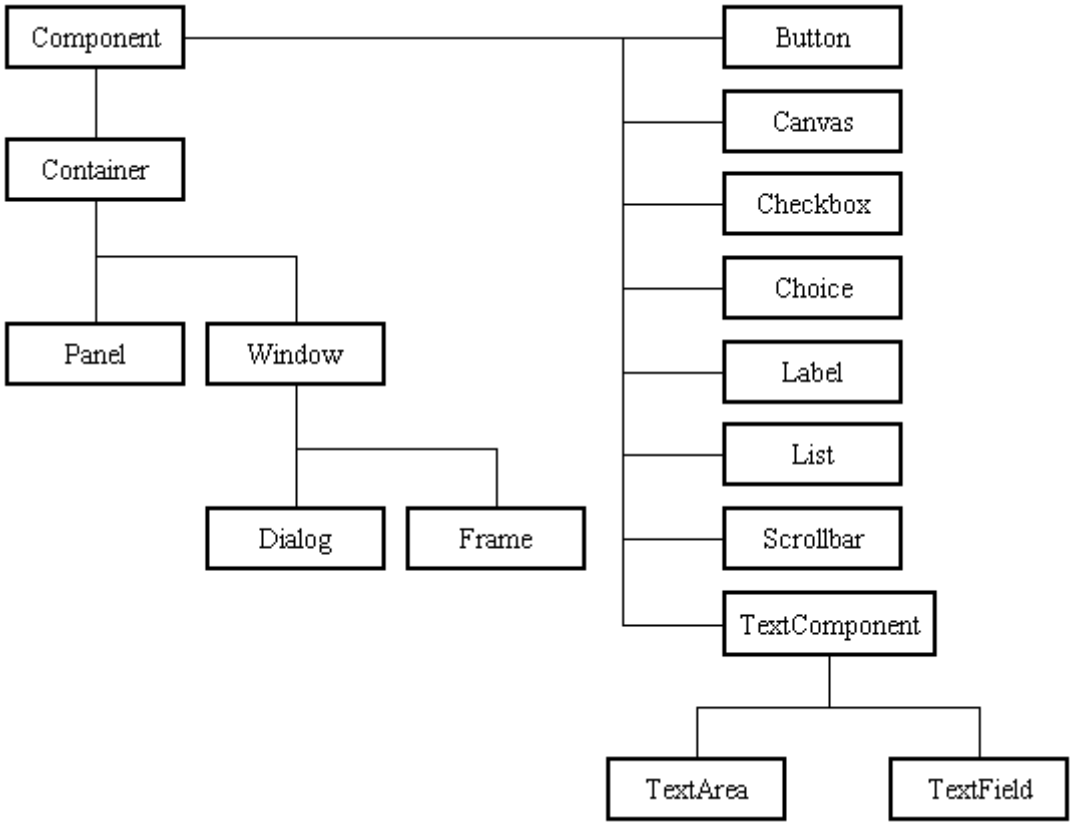


Figura 4.23 – Tipos de componentes na biblioteca AWT (extraído de [12]).

Um C*omponent* (componente) é um objecto que tem uma representação gráfica e pode ser visualizado num ecrã desde que englobado num contentor, define alguns atributos de visualização e permitem ao utilizador interagir com a interface, através da geração de eventos para a aplicação. Um *Container* (contentor) é um componente que pode conter vários componentes. Através da composição de contentores e componentes conseguem-se criar objectos gráficos complexos e ricos.

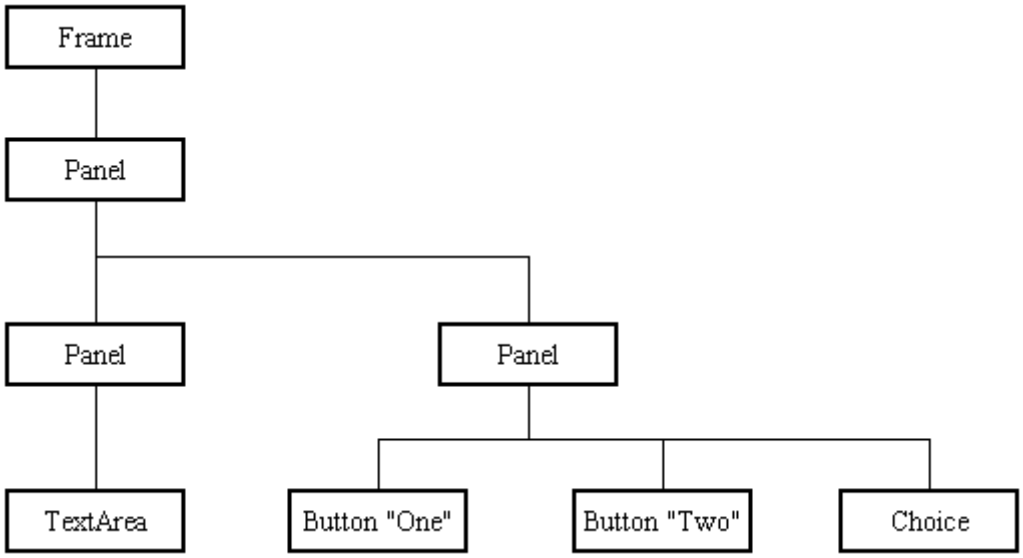


Figura 4.24 – Árvore de componentes numa interface gráfica (extraído de [12]).

Esta biblioteca foi usada no projecto nas aplicações GS e EE. Apesar de não ser uma biblioteca tão rica como, por exemplo, a SWING, possui no entanto toda a funcionalidade necessária para criar interfaces gráficas com uma lógica complexa associada, e é suportável em termos de desempenho por dispositivos com capacidades limitadas. Esta foi um imperativo para a aplicação GS e permite também à aplicação EE correr num ambiente com capacidades limitadas.

4.5.7 Bluetooth OBEX

4.5.7.1 Comunicações Bluetooth

Bluetooth é uma tecnologia para comunicação sem fios entre dispositivos, baseia-se numa ligação de rádio de curto alcance (pode ir até 100 metros), e baixo custo. Esta tem vindo a ser cada vez mais utilizada nos equipamentos móveis, o que se deve ao facto de ser uma norma aceite mundialmente.

Como é natural, a plataforma Java, através dos JSR, criou uma especificação para permitir a aplicações escritas em Java poderem aceder às funcionalidades dum dispositivo *Bluetooth*, o *Java Specification Request* (JSR) 82 [14].

4.5.7.2 APIs Bluetooth

A especificação JSR 82, tem como objectivo providenciar as seguintes capacidades:

- Registo de serviços;
- Descoberta de dispositivos e respectivos serviços;
- Estabelecimento de ligações entre dispositivos usando os protocolos RFCOMM, L2CAP e OBEX;
- Usar as ligações mencionadas para enviar e receber dados;
- Gerir e controlar as ligações;
- Providenciar segurança para estas actividades.

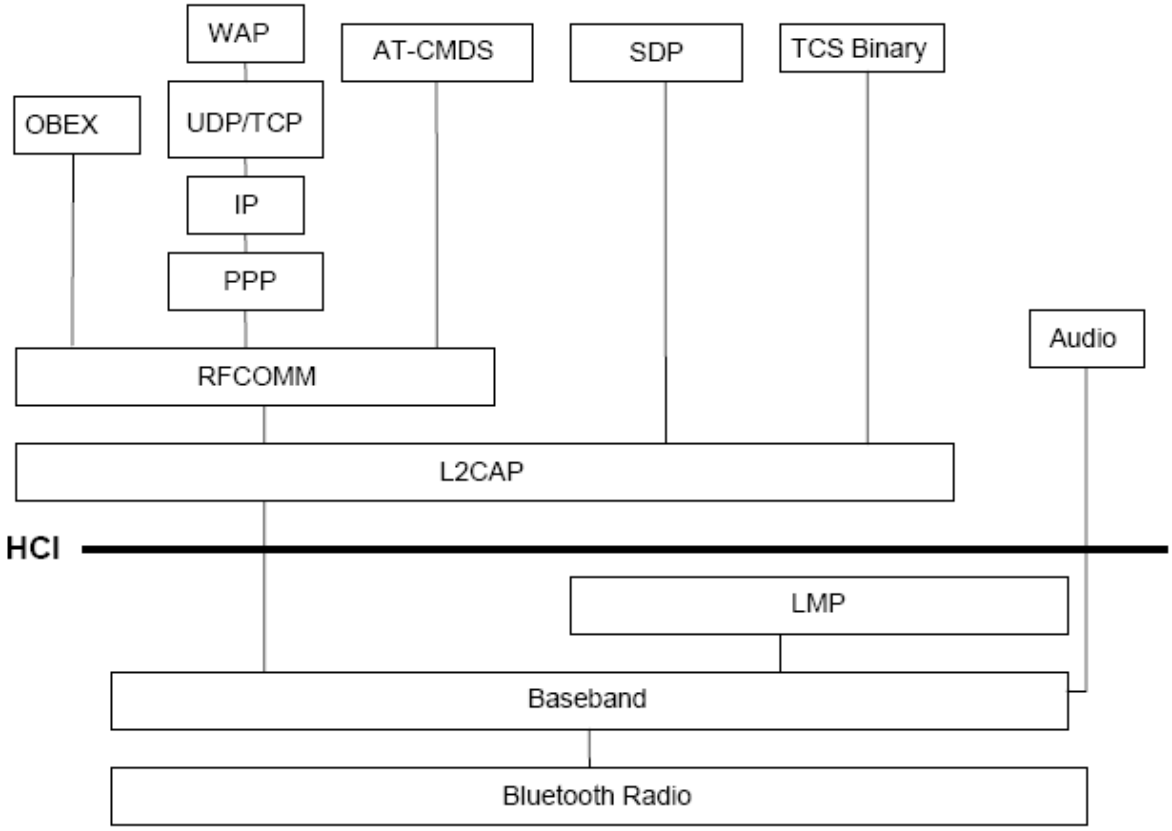


Figura 4.25 – Pilha de protocolos Bluetooth [14].

A API usada no projecto é a correspondente ao protocolo *Object Exchange* (OBEX), sendo a API de mais alto nível que apresenta as funcionalidades necessárias, a troca de dados (objectos) pelas aplicações GS e EE num modelo Cliente – Servidor.

4.5.7.3 Níveis de abstracção sobre API OBEX

A API OBEX, embora seja a de mais alto nível na especificação JSR 82, apenas permite trocar e receber bytes. Por este motivo, e para privilegiar a independência das aplicações GS e EE, implementou-se sobre a API OBEX níveis de abstracção para isolar o modo como os dados são transferidos entre as aplicações, garantido assim uma utilização estruturada, i.e., com alguma independência do transporte dos dados. Os níveis encontram-se listados na figura seguinte.

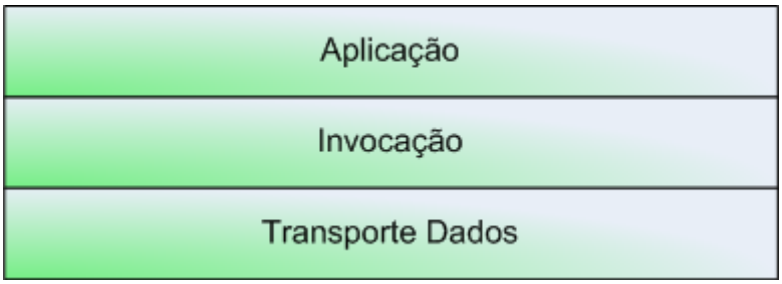


Figura 4.26 – Níveis Bluetooth sobre API OBEX.

Deste modo foi possível à lógica da aplicação apenas conhecer uma interface Java, existindo o lado cliente e o respectivo servidor. O cliente invoca uma interface semelhante ao J*ava Remote Method Invocation* (RMI) [15], enquanto que o servidor a implementa. A fase de descoberta, que é necessária no lado do cliente, o estabelecimento da ligação, e o modo como os dados são trocados, i.e., os detalhes de como a ligação OBEX funciona é escondido da lógica da aplicação.

4.5.8 DB4O – Persistência na aplicação Gestão Sala

A aplicação GS necessita de persistência, tendo em conta as limitações do dispositivo fez-se uma análise das tecnologias existentes.

4.5.8.1 Requisitos necessários

Os requisitos a nível do Sistema de Gestão de Base de Dados (SGBD), na aplicação GS são mínimos:

- Não existem situações em que se façam leituras de objectos que possam estar a ser escritos numa outra transacção;
- As transacções de negócio são elementares, podendo facilmente ser realizadas em transacções simples do SGBD;
- No entanto alguns requisitos poderão impor-se, nomeadamente:
    - Segurança: necessidade de proteger o acesso aos dados presentes na base de dados.

4.5.8.2 Comparação de alternativas

Para escolher o SGBD optou-se por comparar diferentes alternativas existentes no mercado. No mundo dos SGBDs, para dispositivos de poucos recursos, existe uma panóplia imensa de produtos. Para comparar escolheram-se quatro alternativas representativas de diferentes segmentos. Comparam-se SGBDs relacionais (R) com os Orientados a Objectos (OO) e produtos comerciais com produtos *Open Source.* As alternativas estudadas foram as seguintes:

- PointBase Micro [16];
- DB4O [17];
- IBM DB2 EveryPlace [18];
- HSQLDB [19].

Tabela 4.1 – Comparação de SGBDs de acordo com características relacionadas com a integração em dispositivos móveis e de baixos recursos.

SGBD	Tipo	Integração com J2ME	Portabilidade	Requisitos de memória	
				RAM footprint (mínimo)	DB library footprint
PointBase (micro)	R	J2ME (CDC/CLDC) Através de JDBC	Independente da plataforma SGBD Java	~40 KB	~45 KB(CLCD/MIDP) ~90 KB(CDC)
DB4O	OO	Existe versão para J2ME (CDC) c/ reflection Disponibiliza objectos à aplicação Java	Independente da plataforma SGBD Java	~1 MB Objectos partilhados com a aplicação	~350 KB
HSQLDB	R	JDBC	Base de dados Java		
DB2 Everyplace	R	Através de JDBC	Um conjunto vasto SO e plataformas		~200KB

Tabela 4.2 – Comparação de SGBDs de acordo com características de sincronização e de segurança.

SGBD	Sincronização	Segurança
PointBase	Dispõe de soluções para sincronização com diversos SGBDs para servidor	Cifra ao nível das tabelas
DB4O	Não dispõe de soluções built-in.	DB FS protegido por palavra passe DB FS protegido por mecanismos de cifra
HSQLDB	Não dispõe de soluções built-in.	Tabelas protegidas por password
DB2	Dispõe de soluções para sincronização com diversos SGBDs para servidor	Cifra ao nível das tabelas

4.5.8.2.1 Comparação de SGBDs comerciais com Open Source

Dos SGBDs comparados, os comerciais são aqueles que apresentam soluções com mais funcionalidades acessórias como é o caso da sincronização com SGBDs para servidores ou possibilidade de sincronização usando serviços remotos. Essas funcionalidades, especialmente a da sincronização poderiam ser úteis na realização deste projecto, no entanto, optou-se por uma solução *ad-hoc.*

4.5.8.2.2 Comparação SGBDs relacionais com Orientados a Objectos

As bases de dados relacionais têm uma maior quota de mercado do que as orientadas a objectos, estando os produtos deste género bastante amadurecidos e com normas estabelecidas, especialmente pela existência da linguagem Structured Query Language (SQL).

Uma abordagem OO está muito mais próxima da abordagem inerente ao modelo de programação por objectos. No entanto existem diversas soluções para contornar essa desvantagem, por parte dos SGBDs relacionais, através de tecnologias como o JDBC [20] que já permite alguma aproximação ao modelo orientado a objectos, ou, outras mais avançadas como os Object Request Brokers (ORB) de que são exemplos o Hibernate [21] e o OJB [22]. No entanto, a utilização de ORBs deste género requer bastante quantidade de memória volátil, requisito que não pode ser satisfeito numa aplicação para plataformas de baixos recursos como é o caso. A opção JDBC seria uma opção bastante razoável dado o nível de requisitos da aplicação.

4.5.8.3 DB4O

Face à alternativa SGBD Relacional + JDBC existia a alternativa SGBD OO.

A alternativa OO estudada, DB4O, apresenta as seguintes vantagens:

- Facilidade de distribuição: o SGBD pode ser distribuído com a aplicação bastando incluir na aplicação um ficheiro JAR (Java Archive File) contendo toda a aplicação SGBD. Não existe, portanto necessidade prévia de instalação da base de dados nos equipamentos alvo. O SGBD HSQLDB bem como o PointBase Micro também dispõem desta facilidade.
- Facilidade de integração com a aplicação: esta solução, por ser orientada a objectos permite a escrita e leitura implícita de objectos a partir da base de dados, não existindo a necessidade de reconstruir objectos a partir do modelo relacional, nem de explicitar quaisquer regras de conversão de objectos presentes na base de dados para a aplicação ou vice-versa.

Em termos de desempenho, as pesquisas efectuadas sobre comparações com outros SGBDs concorrentes, bem como alguns testes efectuados indicam que o desempenho do SGBD DB4O é satisfatório.

Pelo facto da alternativa OO estudada, DB4O apresentar uma maior facilidade na integração com uma aplicação OO, e pelo facto dos autores deste trabalho quererem experimentar uma abordagem diferente daquela a que estavam habituados, i.e., SGBDs relacionais, e pela pouca necessidade a nível de requisitos optou-se por usar o SGBD DB4O.

De modo a permitir a independência da aplicação face a um SGBD específico, usou-se o padrão de desenho DAO (Data Access Object) [23].

4.5.9 XML

É um formato de texto simples, que permite guardar informação de modo independente de plataformas, devido a ter uma estrutura bem definida e completamente extensível de acordo com as necessidades.

Existem diversas ferramentas, denominados *parsers*, disponíveis para a linguagem Java, mas muitos destes para tornar a tarefa do programador mais simples, tornam-se pesados em termos computacionais, nomeadamente em termos de memória necessária, tendo em conta as limitações da plataforma da aplicação GS. Após uma pequena pesquisa, escolheu-se o *parser* kXML [24], uma implementação directamente orientada a dispositivos com poucas capacidades.

O XML no projecto é importante na aplicação GS sendo usado para tornar o protótipo extensível e configurável. A aplicação GS necessita de dados para correr a aplicação, como por exemplo, uma lista de alunos, o exame que se vai realizar, entre outros, estes dados são obtidos dum formato XML conhecido pela aplicação. A aplicação torna-se configurável porque facilmente se podem mudar os dados da aplicação a realizar, e extensível porque outra aplicação pode criar os dados necessários à aplicação GS, sem ser necessário alterar a última. A aplicação que tem como objectivo criar os dados encontra-se identificada e especificada na análise de requisitos, seria a aplicação Gestão Central (GC). A aplicação GS além de ler os dados necessários para a execução da aplicação, também escreve os dados relevantes gerados na realização do exame para um ficheiro, obviamente também em XML, mais uma vez tendo em vista a extensão da aplicação.

5 Demonstração da solução

Para demonstrar as funcionalidades do protótipo da aplicação, apresenta-se um caso de uso prático completo.

O caso fictício apresentado respeita à realização do exame da disciplina de "Sistemas de Informação" da Licenciatura em Engenharia Informática e de Computadores do IST. O *Prof. Fonseca* é o responsável da disciplina. Existem cerca de 10 alunos inscritos para o exame. Foram convocados dois vigilantes, o *Prof. Meireles* e o *Eng. Barbosa*. Entre os alunos estão o *Nelson* e o *Miguel*.

5.1 Antes do exame

5.1.1 Actividades do Docente responsável pela cadeira

1. O docente responsável pela cadeira de "Sistemas de Informação", *Prof. Fonseca*, produz o enunciado do exame:
    a. O professor dirige-se ao computador central da cadeira;
    b. Executa a aplicação de Gestão de Exames;
    c. Selecciona a opção "Criar Novo Exame";
    d. Preenche os dados necessários;
    e. Cria perguntas, e adiciona-lhes opções de resposta;
    f. Guarda o enunciado de exame;
2. O *Prof. Fonseca* distribui os alunos inscritos por salas:
    a. Executa os mesmos passos descritos em 1.a e 1.b;
    b. Selecciona o exame sobre o qual pretende distribuir alunos por salas;
    c. Obtém a lista de alunos a partir do Sistema Fénix;
    d. Obtém a lista de salas disponíveis para o exame a partir do Sistema Fénix;
    e. Executa a opção "Distribuir alunos por salas automaticamente";
    f. Uma lista com a distribuição de alunos por salas é exibida no ecrã;

i. Na lista consta que o *Miguel* irá realizar o exame na sala GA1, e que o *Nelson* irá realizar o exame na sala FA3.

g. O Docente responsável exporta a lista para a página da cadeira.

3. O *Prof. Fonseca* disponibiliza o exame para sincronização com os PDAs dos operadores de exame:

   a. Executa a mesma sequência de passos de 2.a e 2.b;

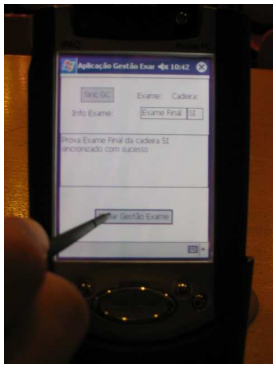
   b. Escolhe a opção “Disponibilizar Exame para sincronização”.

5.1.2 Actividades do Operador do Exame

1. O Operador de Exame *Eng. Barbosa,* sincroniza-se com o computador da Gestão Central:

   a. O *Eng. Barbosa* dirige-se ao computador central da cadeira e liga o PDA ao referido computador;

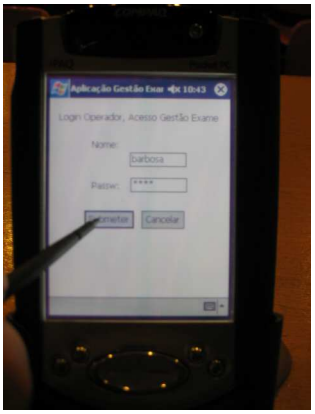
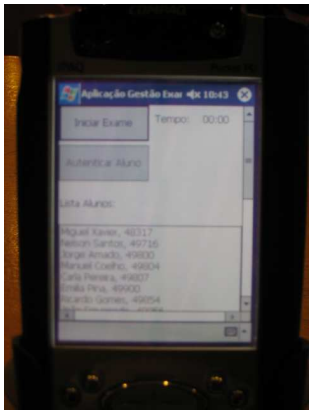
   b. Após estabelecida a ligação entre os dois dispositivos, o *Eng.* pressiona o botão “Sincronizar GC”.



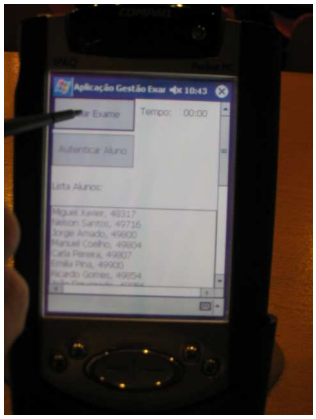
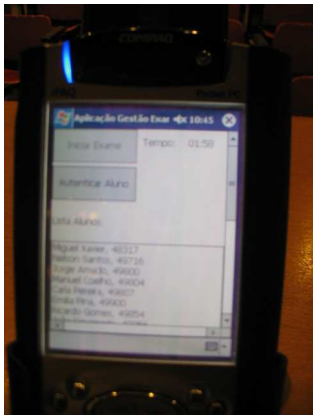
5.2 Durante o exame (sala)

5.2.1 Actividade do Operador do Exame

1. O operador do Exame, *Eng. Barbosa*, autentica-se para acesso à gestão sala:
    a. Introduz a palavra passe para ter acesso à gestão do exame;
    b. Pressiona Submeter e tem acesso ao ecrã principal.

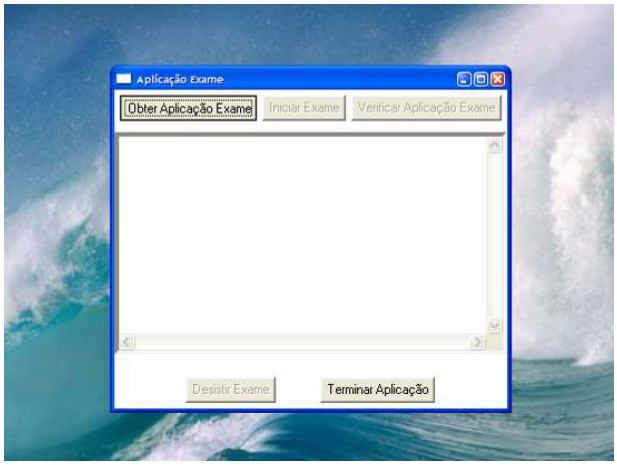



2. Inicia o exame, disponibilizando os enunciados aos alunos:
    a. Pressiona o botão iniciar exame;
    b. A luz indicadora do *Bluetooth* a funcionar acende no PDA.

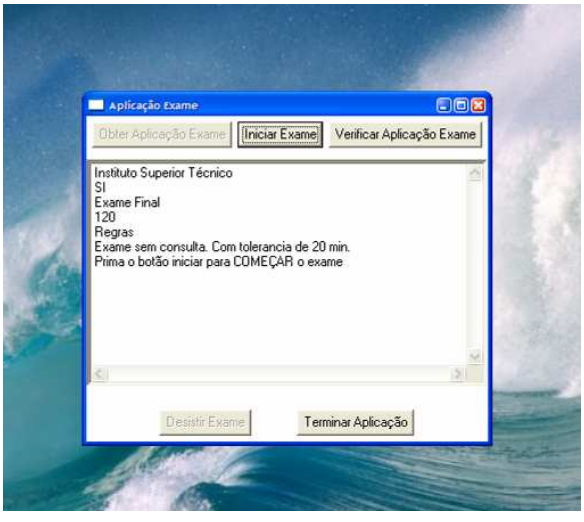
3. Informa os alunos que podem obter o enunciado do exame:
    a. O aluno *Miguel*, insere no leitor de cartões, acoplado ao seu portátil, o seu cartão de aluno;


b. O *Miguel* pressiona o botão obter exame:



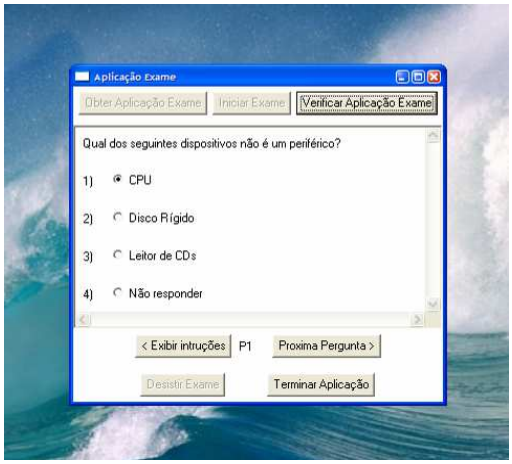
4. O Miguel começa a resolver o exame

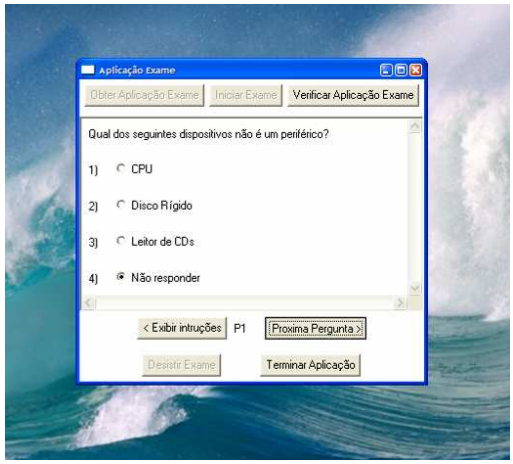
a. As regras do exame são exibidas



b. Miguel inicia o exame e surge a primeira pergunta

c. O *Miguel* escolhe uma opção, e avança de pergunta

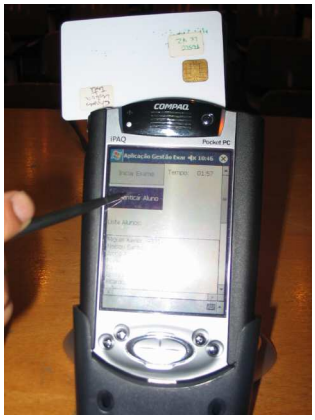




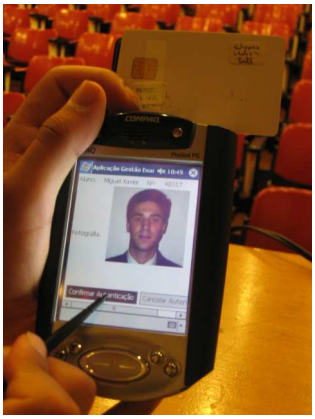
5. Durante a prova, e depois de já ter autenticados outros alunos, o *Eng. Barbosa* vai autenticar o *Miguel*:

   a. Aproxima-se do aluno e este voluntariamente entrega-lhe o seu cartão de aluno;


b. Aproxima o cartão no seu PDA e pressiona o botão autenticar aluno;

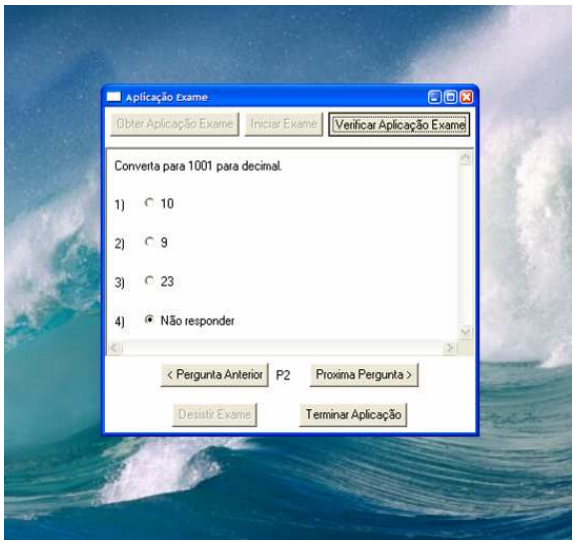


c. É exibida a fotografia do aluno e o *Eng.* confirma se a foto corresponde ao esperado, pressionando o botão Autenticar para registar a autenticação realizada;

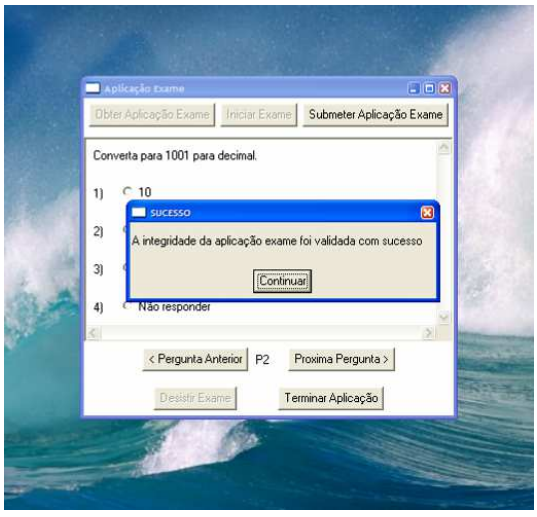


d. O *Eng.* devolve o cartão de aluno ao *Miguel* e dirige-se ao próximo aluno.

6. O *Miguel* decide validar se o exame que recebeu está integro:
    a. Miguel certifica-se de que o seu cartão de aluno está inserido no leitor de cartões que está acoplado ao seu portátil;
    b. O *Miguel* pressiona o botão “Verificar validade do enunciado”;

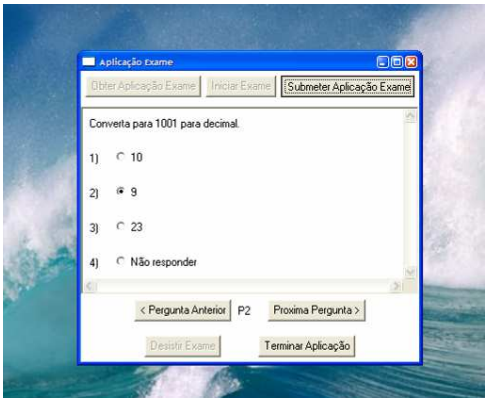


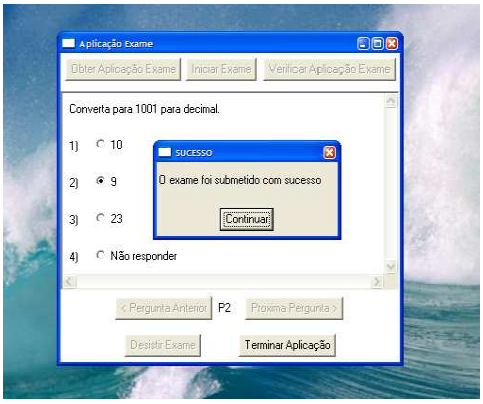
    c. É exibida uma mensagem indicando que o enunciado que o *Miguel* tem presente é integro;



7. Depois de concluir a prova, o *Miguel* decide entregar a sua resolução do exame;
    a. Certifica-se que o seu cartão de aluno está inserido no leitor de cartões que está acoplado ao seu portátil;
    b. O aluno submete o exame pressionando no botão “Submeter Resolução”;

c. É exibida uma mensagem de sucesso na aplicação do aluno *Miguel*;

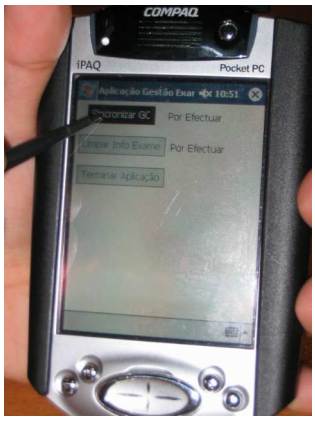


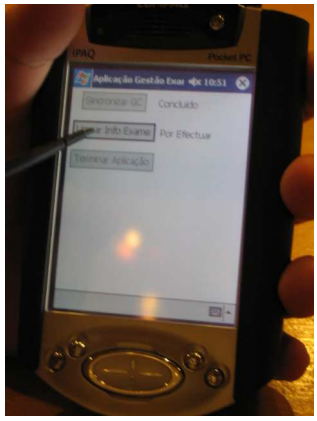


5.3 Após o exame

5.3.1 Actividades do Operador de Exame

1. O exame terminou, o Operador de Exame da sala FA3, *Eng. Barbosa* certifica-se que todos os alunos, entre os quais está o *Miguel*, entregaram as resoluções do exame, ou desistiram.
2. As resoluções dos exames dos alunos estão guardadas no PDA do *Eng.*;
3. O *Eng.* decide sincronizar as resoluções recolhidas na sala com o computador principal da cadeira, Gestão Central (GC):
    a. O *Eng.* dirige-se ao computador central da cadeira e liga-lhe o PDA;
    b. Após estabelecida a ligação entre os dois dispositivos, pressiona o botão "Sincronizar GC";





c. As resoluções e informação associada ficam convenientemente guardadas no computador da Gestão Central;

5.3.2 Actividades do Docente responsável

1. O docente responsável insere a correcção do exame no computador da Gestão Central:
    a. Acede ao computador da GC;
    b. Escolhe o exame sobre o qual pretende inserir a correcção;
    c. Por cada pergunta o docente responsável selecciona a opção de resposta correcta;
    d. Guarda a correcção.
2. O Docente responsável pela cadeira decide avaliar as resoluções dos alunos:
    a. Acede ao computador da GC;
    b. Escolhe o exame sobre o qual pretende avaliar as resoluções dos alunos;
    c. Executa a opção “Correcção automática”
    d. É exibida no ecrã do computador a pauta resultante da correcção automática;
3. O Docente responsável decide publicar a pauta de um exame:
    a. Acede ao computador da GC;
    b. Escolhe o exame sobre o qual pretende pauta;
    c. Escolha a opção “publicar pauta”;
    d. A pauta, onde se destaca a nota do aluno *Nelson*, que obteve 20 valores, é exibida;
    e. O Docente exporta a pauta para a página da cadeira, no sistema Fénix.

6 Avaliação de resultados

Neste ponto realiza-se uma avaliação crítica da solução desenvolvida, do ponto de vista dos requisitos de negócio e da utilização da tecnologia.

6.1 Avaliação dos requisitos de negócio

A avaliação de requisitos de negócio é efectuada com base nos requisitos descritos no ponto 3.3.6. São avaliados com três cores diferentes que significam o seguinte:

-  – Requisito totalmente satisfeito;
-  – Requisito parcialmente satisfeito;
-  – Requisito não satisfeito.

A avaliação dos requisitos é acompanhada por uma breve justificação.

Tabela 6.1 – Requisitos Segurança, nível de satisfação.

Requisitos de Segurança: disponibilidade, integridade e confidencialidade	**Satisfação**	**Comentários**
A autenticação dos alunos tem que ser feita de forma presencial.		Aluno encontra-se a uns metros do professor.
Garantir privacidade e isolamento do aluno, de modo a que só o próprio consiga ter conhecimento da sua resolução durante a realização do exame.		Excepto se o Aluno espreitar o dispositivo do aluno do lado.
Garantir a integridade e autenticidade das informações trocadas entre as diferentes entidades participantes no prestação do serviço.		Dados trocados são assinados digitalmente.
Garantir que os exames são recolhidos de forma íntegra e privada.		Dados cifrados através de chave simétrica partilhada por cada aluno com operador de exame.
Garantir que o aluno não pode obter informação por meios não autorizados.		Envolve bloquear as restantes aplicações no dispositivo do aluno, não realizado.

Tabela 6.2 – Requisitos Fiabilidade e Tolerância a Faltas, nível de satisfação.

Requisitos de Fiabilidade e Tolerância a Faltas	**Satisfação**	**Comentários**
Garantir a fiabilidade do processo de distribuição e recolha das respostas.		Apesar de implementado, não foram realizados testes de uso com vários utilizadores.

Garantir os requisitos de segurança, mesmo ocorrendo faltas relativas às comunicações.		Aplicação robusta no tratamento de casos excepcionais.
Garantir a durabilidade da informação associada à resolução do exame.		Fora do âmbito do protótipo. Foco da aplicação GC.
Garantir o acesso aos dados necessários à autenticação mesmo num ambiente desligado.		Este ponto é simulado no âmbito do protótipo

Tabela 6.3 – Requisitos Ergonomia e Desempenho, nível de satisfação.

Requisitos Ergonomia e Desempenho	**Satisfação**	**Comentários**
Usar suportes para os IDs que não sejam maiores do que o formato dos cartões de estudante actuais (cartão de crédito).		Uso dos cartões *Calypso* (Lisboa Viva).
A interface de resolução do exame deve ser fácil de usar.		Utilização de objectos gráficos e técnicas de interacção comuns a aplicações informáticas
Disponibilizar informação sobre exames facilmente acessível e compreensível.		Fora do âmbito do protótipo.
Permitir facilidade e rapidez nas inscrições em exames.		Fora do âmbito do protótipo, possível integração com sistema Fénix (caso IST).
Permitir a recolha automática de exames.		Garantido pela aplicação, realizado electronicamente.
Permitir a realização do exame em diversas salas sem limitações de distância.		Aplicação GS independente da localização.

Permitir a realização assíncrona do exame nas diversas salas.		Várias aplicações GS em execução são independentes entre si.
Permitir a alunos não inscritos realizar o exame.		Embora tido em conta nas especificações, não foi totalmente implementado no protótipo.
Permitir variedade de equipamentos.		Desde que suportem uma máquina virtual Java.
Permitir ao aluno ter acesso à sua resolução para efeitos de revisão de provas.		Não implementado, mas de simples execução.
O processo de autenticação, autorização, distribuição do exame e recolha do exame em cada sala devem ter um desempenho suficiente para que não haja atrasos significativos relativamente à hora estipulada para o início do exame, assim como para o instante em que o aluno pode abandonar a sala.		Não foi testado em ambiente multi-utilizador, pelo que não é possível garantir com certeza.

6.2 Avaliação técnica

O projecto fez uso de variadas tecnologias, o âmbito de uso de cada uma e um resumo da mesma, encontra-se na descrição da arquitectura tecnológica, ponto 4.5.

Como avaliação global da tecnologia usada faz-se um saldo bastante positivo, o seu uso foi efectuado sempre com uma abordagem que privilegiou a reutilização de tecnologia, e a futura extensão da solução desenvolvida.

6.2.1 Metodologia

O desenvolvimento da solução baseou-se em protótipos iniciais que permitiram validar o uso da tecnologia mais crítica, i.e., as que poderiam levantar problemas por inexistência de suporte das plataformas usadas, e as que poderiam não estar preparadas para correr no ambiente de desenvolvimento seleccionado.

Os protótipos serviram para validar o uso das principais tecnologias, nomeadamente JNI, API AML, Bluetooth e persistência, em ambos os ambientes de desenvolvimento. Como é natural o ambiente de desenvolvimento que poderia levantar e levantou mais problemas foi o *Compaq Ipaq 3870* com SO *Windows Mobile 2002*, pois não existe tanto suporte e soluções para este tipo de ambientes.

6.2.2 Tecnologia

Na reutilização de tecnologia existente destaca-se o uso da arquitectura *Calypso* e da API AML. A arquitectura de segurança *Calypso* foi reaproveitada de modo a garantir a segurança necessária no acesso ao serviço a todos os intervenientes. Os cartões Lisboa Viva e suas características foram adaptadas com sucesso para cartões utilizador do Aluno. A API AML, as bibliotecas da *Link Consulting*, foram usadas praticamente inalteradas, sofreram, apenas, pequenas alterações que corresponderam a correcções de incompatibilidades de versões com o *wrapper* Java e alguns erros que impediam o correcto funcionamento na plataforma móvel usada, o *Compaq Ipaq 3870* com o ambiente *Windows Mobile 2002*. Alguns destes erros, embora de simples resolução, não foram facilmente descobertos e consumiram bastantes horas de trabalho, quer da parte do grupo, quer da parte de elementos da *Link*. A metodologia baseada em protótipos iniciais foi extremamente vantajosa para estes problemas, pois com os testes efectuados obteve-se um mais fácil isolamento dos erros e respectiva resolução.

Um dos grandes objectivos da solução, garantir a segurança necessária ao serviço, é concretizado maioritariamente através da reutilização da tecnologia *Calypso* e complementada com o uso de criptografia simétrica, que não é directamente suportada pelos cartões *Calypso* e teve de ser implementada em software. Isto cria dependências de software entre as aplicações GS e EE, pois têm de usar implementações dos mesmos algoritmos criptográficos.

No uso de toda a tecnologia e desenvolvimento da solução, nunca se encarou o protótipo como uma aplicação “apenas” protótipo. Este foi desenvolvido usando as metodologias aprendidas durante todo o curso, por iniciativa própria e por estudo adicional realizado no âmbito do projecto. Estas metodologias permitiram desenvolver uma solução onde se identifica claramente a lógica da aplicação, i.e., a inteligência da aplicação, e todas as interfaces que esta manipula e/ou acedem à lógica da aplicação. São exemplos das metodologias descritas: as camadas de abstracção para o uso dos dispositivos Bluetooth (ver

ponto 4.5.7.3); o mediador usado entre a lógica da aplicação e a biblioteca Java AWT, que permite à lógica da aplicação não ter conhecimento de qual o tipo de interface usado; o uso do padrão de desenho DAO na aplicação GS para o acesso à persistência, permitindo deste modo manter a independência da aplicação face ao suporte persistente; e o uso de XML como formato para sincronização de dados entre aplicações.

Esta separação clara de responsabilidades, tem o objectivo de privilegiar muito a organização da aplicação, tornando fácil a sua extensão e/ou alteração, sendo este claramente um dos pontos fortes da aplicação.

Apesar do grande esforço dedicado ao projecto, como é natural em qualquer solução, existem pontos menos fortes. Neste caso é o desempenho da aplicação GS, embora seja quase totalmente justificável pelas capacidades limitadas do dispositivo usado e a complexidade da aplicação e tecnologias existentes. Existem pontos que poderiam ser melhorados, tais como:

- O modo de invocação de serviços no Bluetooth;
- A complexidade da interface gráfica.

Mas existem outros que não:

- O tempo que o programa demora a abrir a Base de Dados;
- O tempo que demora a conseguir comunicar com um cartão inteligente.

Estas operações complexas chegam a levar alguns segundos a completar, o que é razoável, mas poderia ser uma limitação num caso real de utilização com muitos alunos, por exemplo, para autenticar, operação que possuí escritas e leituras dum cartão do aluno.

7 Conclusões

Neste capítulo apresentam-se as principais contribuições do trabalho desenvolvido, o trabalho futuro e apreciações finais representativas da opinião pessoal dos elementos do grupo que realizaram este trabalho.

7.1 Principais contribuições

O caso de estudo escolhido, juntamente com os requisitos impostos para o trabalho, envolveu grandes desafios, que foram na sua maioria resolvidos, o que faz crer que é um saldo bastante positivo.

Os requisitos principais do trabalho são requisitos de segurança e mobilidade, a segurança é o factor nos dias de hoje que pode determinar o sucesso de adopção dum serviço, e a mobilidade está associada à proliferação dos dispositivos móveis, e consequente tendência de migração dos serviços para este ambiente.

O serviço associado ao caso de estudo escolhido já é fornecido hoje em dia, mas em moldes bastante diferentes, que serviram de base ao levantamento dos requisitos para o novo molde, tendo sempre como objectivo mínimo a garantia da segurança existente, e sempre que possível a sua extensão ou melhoria. Como é natural, o novo molde traz muitas vantagens a diversos níveis, mas este trabalho foca a segurança e a mobilidade.

A solução desenvolvida cumpre todos os requisitos de segurança necessários e propostos, este objectivo foi atingido com a dificuldade acrescida de reutilizar tecnologia, tal como descrito nos capítulos anteriores, a segurança foi obtida usando uma arquitectura inicialmente desenhada para o ambiente dos transportes, que foi adaptada com sucesso de modo a garantir a confiança necessária ao serviço.

Além dos requisitos referidos que são satisfeitos, salienta-se também o facto da solução ter sido desenhada dum modo robusto, suportando casos de falha, sem dependência de infra-estrutura dedicada, e dum modo totalmente escalável. O seu funcionamento desligado permite esta robustez acrescida, essencial para um sistema de negócio com esta importância. A robustez advém do facto de não haver dependências entre salas, i.e., o exame poder decorrer em diversas salas com desfasamentos no tempo e totalmente desligadas entre si. Falhas numa sala em nada afectam a realização do exame noutras salas.

As propriedades da solução descritas, tornam evidente o facto da solução ser totalmente escalável, em nada afecta a realização do exame, este decorrer numa sala ou em várias, apenas tem de existir um dispositivo por sala com a aplicação GS instalada, e respectivos recursos que sejam necessários.

Através de todos estes requisitos e propriedades atingiu-se uma solução que:

- Usa poucos recursos computacionais, pelo menos da parte dos examinadores (o IST, por exemplo), apenas um dispositivo (um PDA por exemplo) por sala;
- Não teria custos muito pesados de adopção da tecnologia, pois já existe e poderia funcionar em simultâneo com o passe de transportes Lisboa Viva (neste caso em instituições de ensino de Lisboa ou de outras cidades com cartões da família Calypso);
- Não necessita de reserva permanente de espaços físicos;
- Não impõe limitações ao número e espaçamento das salas utilizadas;
- Necessita que cada aluno possua um dispositivo electrónico portátil (PC portátil, PDA com algumas capacidades, outro semelhante);

Da análise dos pontos supra citados retira-se que o ponto mais limitativo à adopção para uma utilização real é o facto de cada aluno ter de possuir um dispositivo, embora cada vez se note uma divulgação maior deste tipo de dispositivos. É um objectivo ambicioso, mas não impossível. Como apoio à transição de tecnologia, a instituição poderia sempre disponibilizar alguns dispositivos aos alunos que necessitassem.

7.2 Trabalho futuro

Como qualquer solução informática, existem sempre pontos que podem ser melhorados, sendo esta uma aplicação protótipo ainda é mais natural que não seja excepção.

Ao longo do desenvolvimento do protótipo identificaram-se pontos que poderiam ser melhorados, ou realizados de outro modo, de entre os quais se destacam os seguintes:

- Tornar a aplicação EE mais “leve”, transferindo uma maior responsabilidade para a aplicação GS no que respeita à interface gráfica de resolução do exame. Se a aplicação EE receber um programa de resolução do exame, que se saiba apresentar graficamente

por si, a interface gráfica pode ser totalmente parametrizada pelo docente da disciplina;

- O tempo de duração da aplicação apenas pode ser controlado após a realização do exame, i.e., o aluno pode ultrapassar o tempo permitido para a resolução do exame, que esse factor apenas é validado *à posteriori*. Tentar melhorar esta situação de modo a obrigar o aluno a entregar a aplicação quando terminasse o tempo, envolve conseguir bloquear o dispositivo do aluno de modo a que este não consiga alterar o tempo desde que a aplicação EE recebe o exame até à respectiva entrega da resolução;
- Testar a troca de informação dos cartões inteligentes de modo remoto, i.e., autenticação e escrita da chave de sessão sem necessidade de o cartão do aluno ter de ser temporariamente entregue ao operador do exame. Envolve transportar as comunicações entre o cartão SAM do operador e cartão utilizador do aluno através da comunicação sem fios, o que levanta problemas de troca de dados confidenciais por uma ligação insegura. Uma hipótese estudada para a resolução deste problema é a introdução dum protocolo complexo fazendo uso de criptografia simétrica e assimétrica para cifrar e autenticar os intervenientes. Uma versão inicial deste protocolo encontra-se descrito no apêndice A.

Além das melhorias descritas acima, é de salientar que o protótipo não foi testado em ambiente multi-utilizador. Como trabalho futuro, na transição de protótipo para produto, é importante validar o desempenho da aplicação com vários utilizadores, i.e., com vários alunos a quererem aceder ao serviço e efectuar testes com tecnologias alternativas que possam ser mais rápidas, validando se o *Bluetooth* é a melhor opção como comunicação sem fios, se existem SGBDs melhores do que o escolhido, e avaliar a relação "custo – benefício" de migrar a aplicação para um PDA com melhor desempenho. É também importante efectuar testes de usabilidade da solução, realizando melhorias iterativas da interface utilizador.

Existem ainda alguns pontos que, embora devidamente identificados, não foram incluídos no desenvolvimento do protótipo. Isto deveu-se a limitações de tempo, mas sendo a solução desenvolvida um protótipo, considera-se natural que não se encontrem todos os requisitos implementados. Alguns destes pontos são:

- Suporte por parte dos níveis de abstracção *Bluetooth* de transmissão de volumes de dados superiores a 1024KB, esta é uma limitação da implementação da API OBEX usada. Esta limitação poderia ser torneada sem qualquer mudança de código "aplicacional", i.e., sem alterações ao código da lógica da aplicação. As alterações ocorreriam apenas nas camadas de abstracção sobre a comunicação *Bluetooth* (ver ponto 4.5.7.3).
- Na aplicação EE não está implementado o armazenamento persistente da assinatura que é recebida pelo aluno ao submeter a sua resolução de exame. É um ponto de fácil implementação, envolve escrita num ficheiro e um diálogo gráfico para decidir a localização do respectivo ficheiro.

7.3 Apreciação final

Uma das tecnologias usadas foi a plataforma Java. A plataforma Java está normalizada por um conjunto de especificações cujas implementações têm de cumprir. Estas especificações são importantes pois permitem que a plataforma abstraia aspectos como o *hardware*, ou o sistema operativo sobre os quais funciona. Desta forma, o cliente da plataforma tem a possibilidade de desenvolver soluções independentes desses aspectos, reduzindo, bastante, os custos de adaptação da solução a outras plataformas. No que respeita à obtenção de uma maior massificação no fornecimento e utilização de serviços, a utilização da plataforma Java é uma mais valia já que ao ser produzida uma solução, nomeadamente um serviço, este ficará acessível a um conjunto vasto de plataformas para as quais exista uma máquina virtual Java. Isto sem que os custos de desenvolvimento aumentem.

Outra tecnologia usada durante a realização do trabalho foi a tecnologia Calypso através da API AML. A utilização da API AML, através de um conjunto de bibliotecas, proporciona uma abstracção de aspectos como o hardware, sistema operativo, o tipo de cartões utilizador ou tipos de comunicação utilizados. A existência desta *API* permitiu, numa primeira fase,

desenvolver a solução GS, num PC e mais tarde adaptar essa solução à plataforma *PocketPC* com o mínimo de esforço[10].

A utilização do sistema de segurança Calypso, através da API AML é um exemplo de que uma tecnologia existente, utilizada no domínio dos transportes, pode ser adaptada a outro domínio, neste caso ao ensino. Esta adaptação evita os custos inerentes ao desenvolvimento de uma solução de raiz. Além disso permite que um universo já existente de utilizadores da tecnologia, através do sistema Lisboa Viva, seja potencial utilizador do novo serviço.

No entanto, a API AML, é uma tecnologia proprietária o que impede que outros fabricantes de soluções na área da segurança implementem a interface da *API*, não sendo por isso fácil a substituição da tecnologia Lisboa Viva por outra que, a nível da segurança proporcionada seja equivalente. A adopção da API AML como norma na área da segurança seria uma mais valia para o sucesso deste tipo de serviços.

Relativamente à comunicação, a plataforma J2ME dispõe do Generic Connection Framework (GCF) que disponibiliza um conjunto de interfaces genérico para envio e recepção de dados em dispositivos de recursos limitados [29]. O GCF impõe um conjunto de interfaces que os vários protocolos de comunicação têm que disponibilizar. A existência desta especificação facilita a substituição do mecanismo de comunicação, no entanto, não é suficiente, pois no nível definido pela GCF trabalha-se com bytes, sendo que o ideal seria trabalhar com objectos. Na realização do trabalho houve, portanto, a necessidade de criar uma abstracção para o transporte de dados que escondesse os detalhes do mecanismo de comunicação, neste caso Bluetooth, optando-se por uma interface semelhante ao RMI (capítulo 4.5.7.3). A existência de normas, nesta área, que proporcionassem ao cliente da plataforma Java a possibilidade de usar interfaces do tipo RMI, ou outra, abstraindo-se do tipo de tecnologia usada na comunicação, Bluetooth, Infravermelhos, Wifi, etc. seria uma mais valia. A existência deste tipo de normas permitiria reduzir os custos associados ao desenvolvimento de soluções para vários tipos de tecnologias de comunicação. Permitiria, ainda a interoperabilidade de tecnologias heterogéneas.

[10] Na realidade existiu algum esforço devido a problemas relacionados com a implementação de alguns componentes da API AML para a plataforma PocketPC.

Neste trabalho optou-se por usar um sistema distribuído em que os componentes funcionam desligados uns dos outros necessitando apenas de ligação, entre si, em momentos não críticos. Desta forma evitam-se eventuais problemas relacionados com a dependência de ligações entre componentes tais como falhas de rede ou ocorrência de falhas no sistema central que poderia ficar indisponível. Estas eventuais falhas poderiam colocar em risco a prestação do serviço diminuindo a sua fiabilidade. Existem pontos de falha, mas estes ou são locais e não comprometem o restante sistema, ou são relativos a falhas que podem ocorrer em momentos não críticos. Relativamente a pontos de falha locais tem-se como exemplo o caso de uma avaria no PDA de uma sala, situação que não afectaria as restantes salas. As operações de sincronização, dado que o sistema é desligado, apesar de sujeitas a falhas, como qualquer sistema, não ocorrem em momentos críticos. Se existir uma falha a operação pode sempre ser efectuada noutra altura sem comprometer o funcionamento do serviço, ao contrário do que se passaria num sistema sempre ligado.

Este tipo de sistema apresenta uma maior flexibilidade de desenvolvimento uma vez que existe maior facilidade em construir os componentes do sistema separadamente.

Os exemplos de utilização da plataforma Java, da tecnologia através da API AML permitem-nos concluir que a adopção de normas e de mecanismos de abstracção de aspectos particulares das plataformas tecnológicas são factores importantes na área dos dispositivos móveis. Estes factores permitem que os custos de disponibilização de serviços para outras tecnologias sejam bastante inferiores, alargando o mercado potencial desses serviços. Permitem também facilitar a reutilização de tecnologia existente, alargando o mercado potencial de clientes desses serviços.

A utilização de sistemas distribuídos em que os componentes funcionam desligados entre si tem vantagens intrínsecas permitindo alcançar maior flexibilidade de desenvolvimento e alcançar um sistema mais robusto, no entanto a utilização de um sistema deste género está sempre dependente dos requisitos do sistema.

Com o aumento progressivo das capacidades associadas aos dispositivos móveis, os serviços serão mais complexos, e também se assistirá ao aumento na procura e oferta de serviços.

8 Referências

[1] Jothy Rosenberg, David L. Remy, 2004, *Securing Web Services with WS-Security*, Sams Publishing.

[2] Kim Topley, 2002, *J2ME in a Nutshell*, O'Reilly.

[3] Paul Clements, et al., 2002, *Documenting Software Architectures: Views and Beyond*, Addison Wesley.

[4] ASK SA, 2000, *GTML 2 User Manual*, França.

[5] Java Community Process (JCP), 2004, *Security and Trust Services API (SATSA) for Java™ 2 Platform, Micro Edition*, Sun Microsystems, Inc., USA.

[6] Miguel Pardal, 2005, *Modelo de Serviços Distribuídos*, no âmbito da tese de mestrado "Segurança de Serviços Electrónicos" (Titulo provisório) a publicar em Novembro de 2005.

[7] Alexandre Domingues, Paulo Barreto, 2003, Relatório do trabalho final de curso *Pagamentos Electrónicos em Infra-estruturas de Televisão Digital Interactiva (T-Commerce)*.

[8] http://java.sun.com/j2me/, *Java 2 Platform, Micro Edition*, Sun Microsystems, Inc., USA.

[9] http://www-306.ibm.com/software/wireless/weme/features.html, *WebSphere Everyplace Micro Environment*, IBM Corporation.

[10] http://www.microsoft.com/windowsmobile/developers/default.mspx, *Windows Mobile for Developers*, Microsoft. Corporation.

[11] http://java.sun.com/products/jdk/awt/, *The AWT*, Sun Microsystems, Inc., USA.

[12] Todd Sundsted, http://www.javaworld.com/javaworld/jw-07-1996/jw-07-awt.html, *Introduction to the AWT, A description of Java's user interface toolkit*, JavaWorld.

[13] http://java.sun.com/docs/books/tutorial/uiswing/, *Swing*, Sun Microsystems, Inc., USA.

[14] http://www.jcp.org/en/jsr/detail?id=82, *JSR 82: Java APIs for Bluetooth*, Sun Microsystems, Inc., USA.

[15] http://java.sun.com/products/jdk/rmi/, *Java Remote Method Invocation*, Sun Microsystems, Inc., USA.

[16] http://www.pointbase.com/resourcecenter/pdfs/micro.pdf, *PointBase Micro Product Datasheet*, DataMirror Mobile Solutions, Inc., Santa Clara, USA.

[17] http://www.db4o.com/about/productinformation/db4o%20Product%20Information%20V4.5.pdf, *db4o-4.5- Tutorial*, db4objects Inc., San Mateo, USA.

[18] http://www-306.ibm.com/software/data/db2/everyplace/everyplacedb.html, *DB2 Everyplace*, *DB2 Everyplace Database*, International Business Machines Corporation, New York, USA.

[19] http://hsqldb.org/web/hsqlFeatures.html, *HSQLDB Features*.

[20] http://java.sun.com/products/jdbc/, *JDBC* , Sun Microsystems, Inc., USA.

[21] http://www.hibernate.org/, *Hibernate*, JBoss Inc., USA.

[22] http://db.apache.org/ojb/, *ObJectRelationalBridge-OJB*, The Apache Software Foundation, USA.

[23] http://java.sun.com/blueprints/corej2eepatterns/Patterns/DataAccessObject.html, *Core J2EE Patterns - Data Access Object*, Sun Microsystems, Inc., USA.

[24] http://kxml.objectweb.org/, *kXML Project*.

[25] https://ciist.ist.utl.pt/projectos/fenix.php, *Projecto Fénix*, CIIST – Centro de Informática do IST, Portugal.

[26] http://www.link.pt, *Projecto Lisboa Viva*, Link Consulting, Portugal.

[27] OVUM, 2005, http://store.ovum.com/detail.aspx?ID=1637, *Securing enterprise wireless channels*, OVUM.

[28] Tom Karygiannis and Les Owens, 2002, http://csrc.nist.gov/publications/nistpubs/800-48/NIST_SP_800-48.pdf, *Wireless Network Security 802.11, Bluetooth and Handheld Devices*, National Institute of Standards and Technology, USA.

[29] John Muchow, 2004, *The Generic Connection Framework - Network support in J2ME/MIDP*.

Apêndice A – Comunicação remota para estabelecimento de chave de sessão entre cartões inteligentes

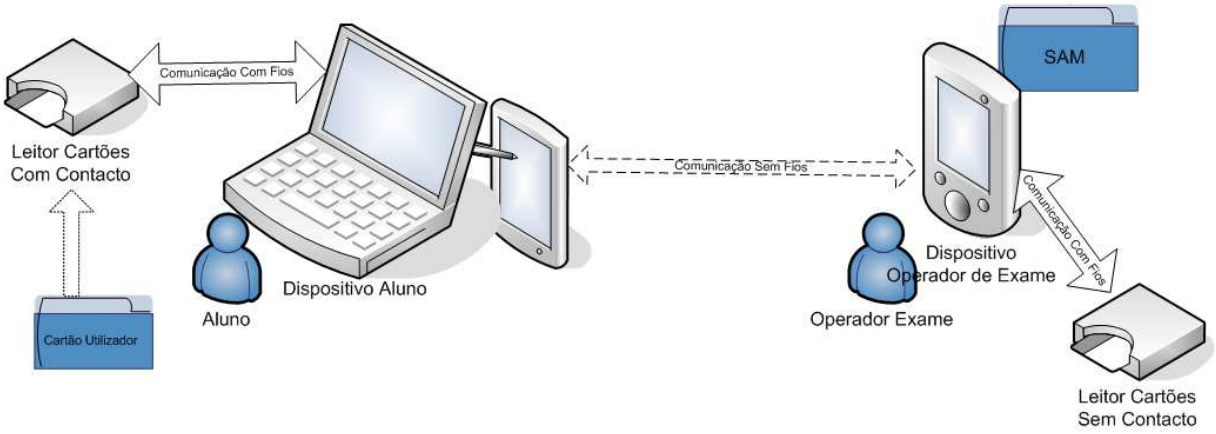


Figura 8.1 – Arquitectura alternativa de utilização da solução.

A descrição da solução apresenta como necessário a existência duma chave de sessão entre cada aluno e um operador de exame, este apêndice descreve um modo alternativo de estabelecer esta chave de sessão.

Estabelecimento de chave de sessão

- PDA gera par de chaves assimétricas, K+ e K-;
- PDA envia K- em aberto ao PC do Aluno.
- O Aluno verifica a autenticidade da chave (K-) graças à interacção segura Cartão – SAM, a chave K- ou o seu resumo é escrito no cartão do Aluno no âmbito de uma sessão segura (ver ponto 4.5.1.2);
- O PC do Aluno gera uma chave de sessão, Cs, e envia-a cifrada com a chave K- ao PDA;
- Só o PDA consegue decifrar a chave de sessão enviada pelo Aluno, pois só o PDA tem conhecimento de K+;
- O Professor envia, no âmbito de uma sessão segura, um resumo da chave de sessão Cs, Dig(Cs) a ser escrito no Cartão Utilizador do Aluno.

- O aluno sabe que o resumo enviado, Dig(Cs), é autentico graças à interacção segura Cartão – SAM, o resumo Dig(Cs) é escrito no cartão do Aluno no âmbito de uma sessão segura com participação do SAM;
- O aluno confirma se o resumo corresponde à chave Cs enviada ao Professor e permite a realização da sessão segura.
- O professor sabe que a chave de sessão que recebeu corresponde à enviada pelo aluno pois o aluno confirmou-a ao autorizar a sessão segura correspondente à escrita do Dig(Cs) no cartão do Aluno.

Nota: A api_aml não está preparada para a execução deste modo, pois não recebe como parâmetros qual o cartão em que vai escrever. Tem que se possuir à partida todos os números de série e respectivos detentores (alunos).

Diagrama do protocolo de estabelecimento de chave de sessão

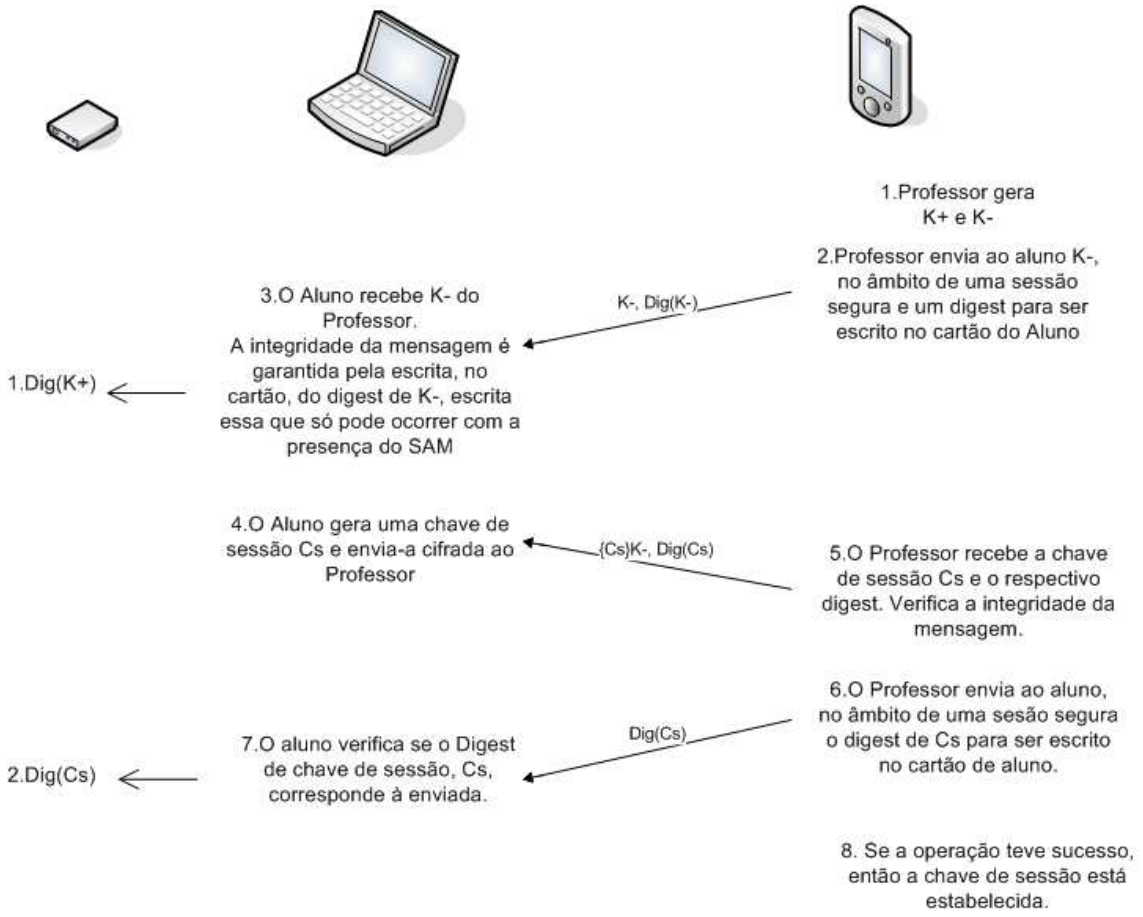


Figura 8.2 – Protocolo de estabelecimento de chave de sessão.