

Mestrado em Engenharia Informática
Estágio
Relatório Final

Integração NONIO-SAP

João Alexandre Amado
jamado@student.dei.uc.pt

Orientador:
Professor Doutor Marco Vieira
Data: 12 de Julho de 2012



FCTUC DEPARTAMENTO
DE ENGENHARIA INFORMÁTICA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Resumo

A capacidade de gerir o processo de obtenção de receitas de uma maneira eficaz e eficiente é um aspecto fundamental para qualquer instituição. A Universidade de Coimbra não foge à regra, e foi com base nessa premissa que nasceu o projecto NONIO-GP. Integrando o sistema de gestão académica NONIO (NOvo sistema académico uNiversitárIO) e o módulo de gestão financeira SAP-FI, o projecto irá permitir toda a gestão de uma das maiores fontes de rendimento da universidade, as propinas e outros emolumentos (onde se incluem candidaturas, requerimentos, certidões, etc.). Este projecto pretende responder a um cada vez maior número de exigências e especificidades de todo o processo de obtenção deste tipo de receitas, oferecendo uma ferramenta integrada que possibilita a estudantes e serviços ter acesso a informação de detalhada sobre a relação financeira entre um aluno e a Universidade de Coimbra.

Ao longo deste documento irá ser detalhada toda a participação do estagiário na evolução de um projecto que será certamente um dos mais importantes a nível informático para a Universidade de Coimbra. Sendo este um projecto que se inicia de raiz, irá ser descrito todo o trabalho desenvolvido pelo estagiário a nível de especificação de requisitos, implementação e um outro conjunto de actividades necessárias à evolução do projecto.

Palavras-Chave

NONIO, NONIO-GP, Gestão Financeira, SAP, Gestão de Propinas, XPaga, CxA SiGES, AMA, SIBS, UNICRE

Agradecimentos

Aos colegas do projecto NONIO gostaria de agradecer todo o apoio e excelente ambiente de trabalho que ofereceram. Em especial gostaria de agradecer ao Pedro Pinto por toda a confiança e responsabilidade que me depositou, ao Ricardo Bica por toda a disponibilidade para ajudar e ensinar e ao João Rafael todo o apoio dado ao longo do ano.

Ao Professor Doutor Marco Vieira gostaria de agradecer a oportunidade que me deu de trabalhar no projecto NONIO.

Por último, mas não menos importante, gostaria de agradecer a todos os amigos e familiares pelo apoio dado ao longo deste ano.

Índice

Capítulo 1 Introdução	7
Capítulo 2 Enquadramento	9
2.1. Projecto NONIO	9
2.1.1. Processo de Desenvolvimento	10
2.1.2. Arquitectura e Tecnologias	12
2.2. Projecto NONIO-GP	13
2.2.1. Aplicações de gestão de propinas predecessoras.....	14
2.3. Tecnologias de Integração de Sistemas	16
2.3.1. Metodologia Seleccionada	18
Capítulo 3 Actividades Desenvolvidas	20
3.1. Integração no Projecto NONIO e NONIO-GP	22
3.2. Módulo de Planos de Pagamento.....	22
3.2.1. Especificação.....	23
3.2.2 Construção.....	29
3.2.3 Lançamento	31
3.3. Módulo de Gestão de Conta Corrente.....	32
3.3.1 Especificação	33
3.3.2 Construção.....	40
3.3.3 Lançamento	40
3.4. Módulo de Caixa.....	43
3.5. Integração NONIO/SAP-FI.....	46
3.5.1 Arquitectura de Integração.....	50
3.5.2 Integração Automática de Pacotes.....	52
3.5.3 Integração Manual de Pacotes.....	54
3.5.3.1 Especificação.....	54
3.5.3.2 Construção.....	56
3.5.4 Lançamento	57
3.6. Módulo Propinas e Emolumentos	60
3.7. Módulo Alunos	61
3.8. Outras Actividades	63
Capítulo 4 Conclusões.....	69
Anexos.....	71

Referências.....72

Lista de Figuras

Figura 1 - Processo de Desenvolvimento de Software no projecto NONIO.....	10
Figura 2 - Ciclo de Vida de um Documento.....	11
Figura 3 - Estrutura Tecnológica das Aplicações do projecto NONIO.....	12
Figura 4 - Transferência de Ficheiros	16
Figura 5 - Base de Dados Partilhada.....	17
Figura 6 - Invocação de Procedimentos Remotos.....	17
Figura 7 - Mensagens Assíncronas	18
Figura 8 - Diagrama de <i>Gantt</i> representativo do trabalho desenvolvido no 1º Semestre.....	20
Figura 9 - Diagrama de <i>Gantt</i> representativo do trabalho planeado para o 2º Semestre.....	21
Figura 10 - Diagrama de <i>Gantt</i> representativo do trabalho realizado ao longo do estágio	22
Figura 11 - Ecrã prototipado do módulo de planos de pagamento	26
Figura 12 - Modelo de Dados Associado ao Módulo de Planos de Pagamento	28
Figura 13 - Definição de um serviço.....	29
Figura 14 - Definição de uma acção.....	29
Figura 15 - Definição do mapeamento de uma acção	29
Figura 16 - Definição de um <i>formbean</i>	30
Figura 17 - Número de linhas produzidas na implementação do módulo de planos de pagamento.....	31
Figura 18 - Ecrã do módulo de planos de pagamento já em produção.....	32
Figura 19 - Ecrã prototipado do módulo de gestão de conta corrente.....	38
Figura 20 - Ecrã prototipado do módulo de gestão de conta corrente.....	39
Figura 21 - Número de linhas produzidas na implementação do módulo de gestão de conta corrente.....	40
Figura 22 - Ecrã prototipado do módulo de gestão de conta corrente.....	42
Figura 23 - Ecrã do módulo de gestão de conta corrente já em produção	43
Figura 24 - Processo de integração de pacotes.....	50
Figura 25 - Definição do job de geração e envio de pacotes.....	53
Figura 26 - Definição do <i>webservice</i> de confirmação de processamento de pacotes.....	53
Figura 27 - Número de linhas produzidas na implementação da funcionalidade de envio automático de pacotes	54
Figura 28 - Ecrã prototipado da funcionalidade de integração de pacotes manual	56
Figura 29 - Número de linhas produzidas na implementação da funcionalidade de envio manual de pacotes.....	57
Figura 30 - Ecrã da funcionalidade de integração de pacotes manual já em produção.....	58
Figura 31 - Ecrã da funcionalidade de integração de pacotes manual já em produção.....	59
Figura 32 - Ecrã prototipado do módulo de propinas e emolumentos	60
Figura 33 - Ecrã prototipado do módulo de alunos	61
Figura 34 - Número de linhas produzidas na implementação do módulo de alunos	62
Figura 35 - Ecrã do módulo de alunos já em produção.....	62
Figura 36 – Arquitectura de Integração com plataforma AMA e PayPal.....	64
Figura 37 - Processo de actualização de dados.....	66

Lista de Tabelas

Tabela 1 - Lista de requisitos do módulo de planos de pagamento	24
Tabela 2 - Lista de requisitos da secção plano de pagamento do módulo de gestão de conta corrente.....	34
Tabela 3 - Lista de requisitos da secção de compensações do módulo de gestão de conta corrente.....	36
Tabela 4 - Lista de requisitos da secção de recebimentos do módulo de gestão de conta corrente.....	37
Tabela 5 - Lista de requisitos da secção de reposições do módulo de gestão de conta corrente.....	37
Tabela 6 - Lista de requisitos do módulo de caixa.....	44
Tabela 7 - Movimentos de dívidas	47
Tabela 8 - Pacotes de dívidas	47
Tabela 9 - Movimentos de recebimentos	49
Tabela 10 - Pacotes de recebimentos.....	49
Tabela 11 - Webservice de confirmação de processamento de pacotes.....	51
Tabela 12 - Retorno do webservice de confirmação de processamento de pacotes.....	51
Tabela 13 - Lista de requisitos da funcionalidade manual de integração de pacotes	55

Capítulo 1

Introdução

O presente documento tem como principal objectivo apresentar o trabalho realizado pelo estagiário durante o ano lectivo 2011/2012 no âmbito do projecto de estágio intitulado “Integração NONIO-SAP”. Orientado pelo Professor Doutor Marco Vieira e co-orientado pelo Engenheiro Pedro Pinto, gestor do projecto NONIO, o estágio foi realizado nos Serviços Centrais da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

Nascido em 2006, o projecto NONIO visou colmatar lacunas, a nível de processos de gestão académica, abertas pela adopção do Tratado de Bolonha pela Universidade de Coimbra. Com o objectivo inicial de ser uma aplicação de gestão académica para a Faculdade de Ciências e Tecnologia da Universidade de Coimbra, o projecto foi bem-sucedido e foi crescendo sendo agora não só responsável pela gestão académica mas também pela gestão pedagógica. O crescimento do projecto passou também pela assimilação das restantes Faculdades da Universidade de Coimbra, sendo agora o sistema NONIO a única aplicação responsável pelas áreas referidas em todas as Faculdades.

Sendo o sistema NONIO já responsável pela gestão académica e pedagógica ficou apenas a faltar a gestão de propinas para que se centralizassem no NONIO todos os processos mais importantes de gestão das relações entre a instituição e os alunos. Assim, e com base num projecto de constante evolução nasce o projecto NONIO-GP.

O projecto NONIO-GP foi criado com vista a melhorar todo o processo de obtenção de receitas de propinas e outros emolumentos (onde se incluem candidaturas, requerimentos, certidões, etc.). Todo este processo é cada vez mais complexo e tem um cada vez maior número de especificidades pelo que o sistema permitirá que estudantes e serviços possam ter acesso a informação detalhada e actualizada sobre todos os aspectos da relação financeira entre a Universidade de Coimbra e o aluno.

Com o objectivo de substituir as aplicações de gestão de propinas actuais, CxA SiGES e XPaga, o projecto NONIO-GP consiste na implementação das funcionalidades de consulta e gestão financeira relacionadas com propinas e emolumentos e na integração com o módulo de gestão financeira SAP-FI de modo a garantir a sincronização da informação entre os dois sistemas.

O módulo SAP-FI é a ferramenta financeira da Universidade de Coimbra. Neste sistema são registados todas as receitas (propinas, financiamento do estado, etc.) e também todas as despesas (salários, investimentos em equipamento, etc.). Ou seja, toda a gestão financeira é realizada neste sistema, daí a necessidade de integração dos dados relativos à gestão de propinas.

Sendo este um projecto iniciado de raiz em Agosto do ano 2011, ao longo deste documento iremos ver todo o trabalho e evolução pessoal e técnica do estagiário paralelamente à evolução do projecto NONIO-GP. Mais precisamente, será detalhado neste documento todo o trabalho do estagiário na especificação de requisitos, prototipagem, desenvolvimento e testes dos módulos de planos de pagamento, gestão de conta corrente, alunos e também na integração com SAP-FI. Adicionalmente serão detalhadas outras actividades realizadas pelo estagiário como a análise de riscos do projecto e o estudo da plataforma de pagamentos da Agência para a Modernização Administrativa (AMA). O trabalho realizado ao longo do ano foi uma experiência completa, permitindo ao estagiário passar por todas as fases de desenvolvimento de *software*.

O presente documento está dividido em quatro secções. Na primeira é feito um enquadramento geral do projecto NONIO e do seu processo de desenvolvimento, bem como uma descrição geral dos princípios base do projecto NONIO-GP. Nesta secção, é realizado ainda um estudo sobre as metodologias de integração de sistemas. Na segunda secção, são descritas todas as actividades do estagiário nos módulos de planos de pagamento, gestão de conta corrente, caixa, alunos, propinas emolumentos e também todo o processo de integração com SAP-FI. A terceira secção descreve outras actividades desenvolvidas pelo estagiário mas que não estão directamente relacionados com os módulos referidos anteriormente. A última secção apresenta as conclusões e experiência pessoal do estagiário da sua participação neste projecto.

Capítulo 2

Enquadramento

Neste capítulo pretende-se contextualizar o leitor, primeiro com o projecto NONIO, dando uma perspectiva geral sobre a história e sobre todo o processo de desenvolvimento que é aplicado por todos os que trabalham no projecto. De seguida pretende-se dar uma visão mais específica do projecto NONIO-GP. Para tal, será dado a conhecer ao leitor em traços gerais os objectivos do projecto e as aplicações de gestão de propinas predecessoras, sendo explicadas as razões que levaram à necessidade de mudança.

Por último, e tendo em conta a componente de integração deste projecto, é feito um levantamento de técnicas de integração de sistemas.

2.1. Projecto NONIO

O projecto NONIO surgiu no ano de 2006 como resposta às mudanças que decorreram da adopção do modelo de Bolonha pela Universidade de Coimbra. Esta mudança levou a que os sistemas existentes na altura deixassem de conseguir responder a todas as exigências do processo de gestão académica.

Com o funcionamento inicial apenas focado na Faculdade de Ciências e Tecnologia (FCT), o projecto tinha como objectivo a criação de um sistema robusto, eficaz e capaz de se adaptar à mudança que, para além de lidar com todas as alterações resultantes do processo de Bolonha, aumentasse toda a capacidade e qualidade da gestão académica da Faculdade.

Com o passar dos anos, a gestão académica das restantes faculdades foi integrada no NONIO. Assim, no ano de 2009 foram integradas as faculdades de Economia e Ciências do Desporto e Educação Física, em 2010 as faculdades de Medicina, Farmácia, Psicologia e Ciências da Educação e o Colégio das Artes. No ano lectivo corrente, foi integrada a última faculdade, a de Letras.

O sucesso do projecto levou a que este se expandisse para lá da área de gestão académica. Assim, no ano de 2010, o projecto sofreu uma evolução passando a ter como objectivo gerir não só a área académica mas também toda a área pedagógica que era, naquele momento, suportada pelo sistema *WoC* (*Web on Campus*). A integração da área de gestão pedagógica deu-se em Setembro de 2010, deixando o sistema de ser exclusivo da faculdade de Ciências e Tecnologia, passando a funcionar também nas faculdades de Ciências do Desporto e Educação Física e Economia.

No início do ano lectivo corrente, foram agregadas as restantes faculdades. Assim, toda a gestão académica e pedagógica está centralizada no NONIO, faltando apenas integrar a gestão de propinas para que todos os processos de gestão passem a ser tratados através do NONIO.

Tendo como gestor de projecto o Engenheiro Pedro Pinto, o grupo de trabalho que integra o projecto NONIO é constituído exclusivamente por elementos que frequentaram ou frequentam o Departamento de Engenharia Informática da Universidade de Coimbra.

Contando no momento com 10 elementos contratados a tempo integral e 2 estagiários, a equipa do NONIO é uma equipa bastante dinâmica e jovem onde a média de idades deverá rondar os 27 anos.

2.1.1. Processo de Desenvolvimento

O processo de desenvolvimento de *software* no projecto NONIO é baseado em regras pré-estabelecidas que procuram normalizar e sistematizar todo o processo de desenvolvimento, tendo como principal objectivo garantir a qualidade e cumprimento de requisitos do produto final.

Como se poderá ver pela figura 1 (retirada do documento geral do processo de desenvolvimento do NONIO – Anexo 1), todo o processo é sequencial (baseado na metodologia *waterfall*) contando no momento com quatro fases distintas mas complementares: especificação, *design*, construção e lançamento. Presentes ao longo de todo o processo encontram-se os documentos de processo e as actividades de gestão e manutenção.



Figura 1: Processo de Desenvolvimento de Software no projecto NONIO

A fase de **especificação** é o momento onde se definem os requisitos. Engloba a criação de um protótipo com base em páginas estáticas de *html*, que serve como auxílio a todo o processo, a escrita do documento de especificações adicionais onde se detalham

todas as especificidades que não são visíveis no protótipo e também a alteração ao modelo de dados com vista a criar as entidades e relacionamentos que suportem a criação ou alteração de funcionalidades ou módulos.

Na fase de **design** são construídos os *designs* detalhados das funcionalidades a desenvolver. Poderá ser necessário actualizar o *design* da base de dados e a arquitectura do *software*. Com a evolução do processo de desenvolvimento, esta fase passou a ser opcional.

A fase de **construção** consiste na implementação das funcionalidades e do código para os testes automáticos. Mais uma vez, com a evolução do processo, a fase relativa à codificação dos testes automáticos deixou de ser obrigatória.

Na fase de **lançamento**, a última do processo, são aplicados os testes funcionais e é produzido um relatório com o resultado destes. Caso os testes regressivos tenham sido codificados na fase anterior, segue-se o mesmo processo descrito para os testes funcionais.

Todos os documentos produzidos estão sujeitos a um controlo apertado de criação e alteração, cujo processo é representado na figura 2 (retirada do documento de plano de controlo de alterações – Anexo 1):

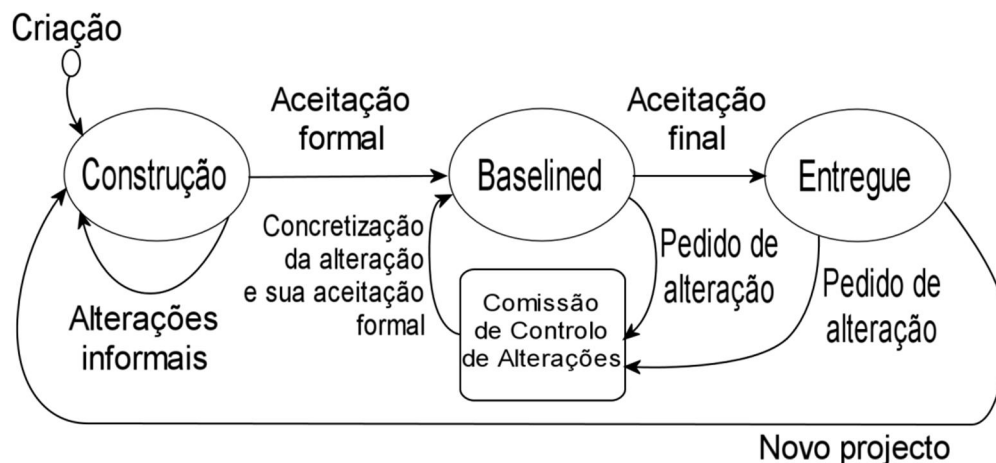


Figura 2: Ciclo de Vida de um Documento

Como podemos verificar pela figura acima, um documento depois de ser criado entra na fase de **construção**. Nesta fase poderá ser alterado livremente até que seja realizada uma aceitação formal onde o documento passa para o estado **baselined**. Nesta fase o documento sofrerá uma revisão por parte de uma comissão de controlo de alterações, cujos membros são definidos pelo gestor de projecto, e permanecerá nesta até que haja uma aceitação final onde o documento passa para o estado **entregue**. Caso haja a necessidade de alterar um documento no estado **entregue**, terá de ser realizado um pedido de alteração voltando o documento a passar por todos os passos da fase **baselined** descrita acima.

Para que se garanta a qualidade em todo o processo existem alguns procedimentos que deverão ser aplicados ao longo do processo de desenvolvimento.

Aplicados aos documentos existem os seguintes procedimentos:

- **Reuniões de análise** – Técnica manual de revisão com o principal objectivo de injectar qualidade nos documentos.
- **Inspecções** – Revisão formal de documentos com o objectivo de detectar erros.
- **Revisões** – Versão simplificada das inspecções que deve ser aplicada a documentos menos críticos.

Aplicados ao código fonte estão contemplados os seguintes procedimentos:

- **Testes Unitários** – Testes aplicados pelo programador a módulos produzidos pelo próprio.
- **Testes de Integração** – Testes aplicados pelo programador e por elementos externos à equipa de programação responsável pelo módulo quando estes estão associados a outros módulos.
- **Testes Funcionais** – Testes aplicados pelo programador e por elementos externos à equipa de programação responsável pelo módulo com o objectivo de verificar se as funcionalidades implementadas têm o funcionamento correcto e esperado.
- **Teste Regressivos** – Testes aplicados através de uma ferramenta de automatização a todos os módulos ou funcionalidades quando estas são associadas à aplicação.

2.1.2. Arquitectura e Tecnologias

O projecto NONIO disponibiliza actualmente três aplicação distintas cada uma direccionada a um tipo de utilizador. São elas: *Inforestud@nte*, *inforegestão* e *infordocente*. A primeira direcciona-se aos alunos e candidatos, a segunda a todos os elementos responsáveis pela gestão académica e a última ao corpo docente e funcionários das unidades orgânicas da Universidade de Coimbra.

A figura seguinte (retirada do documento de arquitectura e design genérico do NONIO – Anexo 2) ilustra as camadas aplicacionais do projecto e a sua estrutura tecnológica:

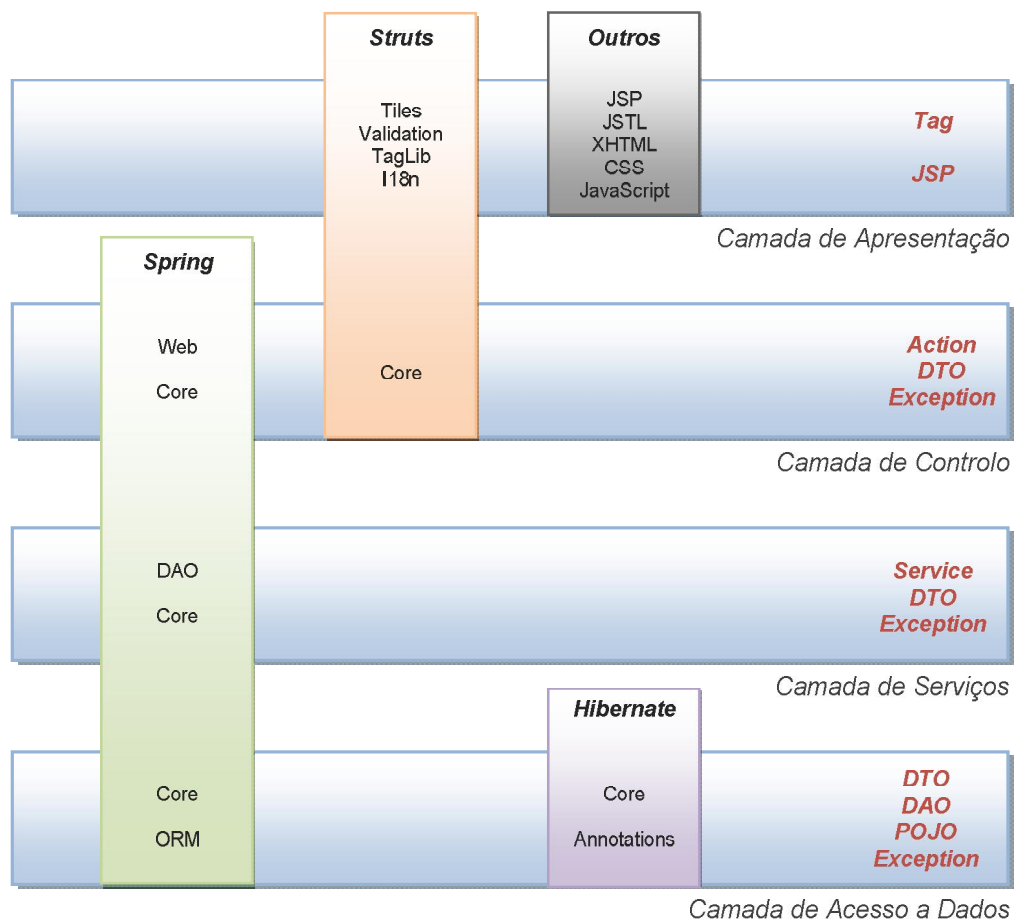


Figura 3: Estrutura Tecnológica das Aplicações do projecto NONIO

Como podemos verificar pela figura acima, as aplicações do projecto NONIO seguem o padrão de *design Model-View-Controller*[1], estando estruturadas em quatro camadas distintas. São elas: camada de apresentação, camada de controlo, camada de serviços e camada de acesso a dados. A primeira, representa o que o utilizador visualiza ou seja, representa a interface da aplicação. A segunda, tem a dupla função de por um lado lidar com pedidos provenientes da camada de apresentação e, por outro lado, de lidar com a informação que é devolvida da camada de serviços. A terceira, é responsável pela manipulação e consulta de toda a informação presente na base de dados e, na última camada, concentram-se todas as classes que permitem abstrair o acesso aos objectos definidos na base de dados.

Na figura também podemos observar que o projecto NONIO utiliza três *frameworks* baseadas em Java:

- **Struts** [2] - *Framework* que disponibiliza os mecanismos necessários à implementação de um modelo MVC. Esta *framework* disponibiliza os componentes Controller e View e interage com outras tecnologias para a implementação do componente *Model*. O *Struts* disponibiliza ainda mecanismos de validação de campos, estruturação de *layouts*, suporte para vários idiomas e um conjunto de *taglibs* que permitem simplificar a construção da componente *View*.
- **Spring** [3] – Framework cujo papel passa pela interligação das várias camadas através da implementação do padrão IoC (*Inversion of control*). Esta *framework* facilita também a utilização da *framework* *Hibernate*.
- **Hibernate** [4] – Esta *framework* é responsável por representar a estrutura relacional de uma base de dados numa hierarquia de objectos. O *hibernate* oferece a sua própria linguagem, baseada em SQL mas direccionada a objectos, o *HQL*.

2.2. Projecto NONIO-GP

Iniciado no mês de agosto de 2011, o projecto conta com elementos de três áreas distintas da Universidade de Coimbra. São elas: a área financeira, a área académica e a área informática, sendo a última representada por elementos do projecto NONIO e do GSIIC (Gestão de Sistemas e Infra-Estruturas de Informação e Comunicação).

Este projecto traduz-se no último passo para a integração no NONIO de todos os serviços de gestão das relações entre a instituição e os alunos da Universidade de Coimbra. Aquando da finalização do projecto, a gestão de propinas, a gestão académica e a gestão pedagógica estará toda centralizada no NONIO.

O projecto NONIO-GP foi criado com vista a melhorar todo o processo de obtenção de receitas de propinas e outros emolumentos (onde se incluem candidaturas, requerimentos, certidões, etc.). Todo este processo é cada vez mais complexo e tem um cada vez maior número de especificidades pelo que o sistema permitirá que estudantes e serviços possam ter acesso a informação detalhada e actualizada sobre todos os aspectos da relação financeira entre a Universidade de Coimbra e o aluno. Outro objectivo é cortar de vez todas as ligações com os sistemas predecessores de gestão de propinas (CXA SiGES [5] e XPaga) passando toda esta a estar concentrada numa única aplicação. Para tal, irão ser criadas, no sistema NONIO, funcionalidades de gestão, irão ser migrados todos os dados das aplicações de gestão de propinas predecessoras e feitas integrações com o sistema financeiro da universidade (SAP-FI [6]), com a plataforma de pagamentos da Agência para a

Modernização Administrativa (AMA), para suporte de pagamentos através de multibanco e cartão de crédito, e com o *PayPal*, para que este meio de pagamento seja também suportado.

Apostando em centralizar toda a lógica de negócio (onde se inclui a criação de prestações, cálculo de juros, criação de referências de pagamento, notificação de pagamentos etc.) no sistema NONIO, pretende-se que toda a gestão de propinas do ano lectivo 2012/2013 já seja realizada a partir deste.

De referir que os desenvolvimentos necessários em SAP-FI foram subcontratados a uma empresa externa.

Nas próximas subsecções serão discutidas as aplicações de gestão de propinas predecessoras, bem como as razões da necessidade de mudança entre aplicações.

2.2.1. Aplicações de gestão de propinas predecessoras

Existem duas grandes aplicações responsáveis por toda a gestão de propinas na Universidade de Coimbra: o CXA SiGES e o XPaga. Nos parágrafos seguintes é detalhado o papel de cada uma das aplicações ao longo dos últimos anos na gestão das propinas da Universidade de Coimbra e é também descrita uma tentativa, prévia ao projecto NONIO-GP, de substituição dos dois sistemas.

CXA SiGES

O módulo de controlo e gestão de tesouraria CXA SiGES é um módulo integrante da aplicação SiGES. Com início de desenvolvimento em 1994 pela empresa *Digitalis*, a aplicação SiGES foi durante largos anos responsável por toda a gestão académica das faculdades da Universidade de Coimbra, exceptuando-se a Faculdade de Ciências e Tecnologia. Criado utilizando as tecnologias Borland Delphi e Oracle, a aplicação integra 18 módulos que permitem gerir as áreas académicas e de recursos humanos. No que toca à gestão de propinas, o SiGES disponibiliza o módulo CXA SiGES, sendo este, no passado, responsável pela gestão de propinas dos alunos matriculados em cursos não pertencentes à Faculdade de Ciências e Tecnologia.

O crescimento do projecto NONIO, como referido anteriormente, levou a que a gestão académica de todas as faculdades passasse a ser feita directamente no sistema NONIO, o que resultou num abandono da aplicação SiGES como meio de gestão académica. Com o abandono da utilização da aplicação SiGES e estando o projecto NONIO desde a sua génese integrado com a aplicação XPaga, foi também abandonado o uso da aplicação CXA SiGES como aplicação de gestão de propinas.

Na realidade, o abandono do módulo CXA SiGES foi apenas parcial, já que nos dias que correm ainda se encontra em utilização mas apenas para a gestão de propinas de alunos com dívidas em cursos associadas a anos lectivos anteriores à integração da Faculdade, a que o curso pertence, no NONIO. Com o projecto NONIO-GP espera-se cortar todos os elos de ligação à aplicação CXA SiGES.

XPaga

Como foi referido na secção anterior, a gestão académica da Faculdade de Ciências e Tecnologia não era realizada através da aplicação SiGES, sendo realizada através da aplicação AGA (Aplicação de Gestão Académica).

Associada à aplicação AGA, encontramos o sistema de gestão de propinas XPaga. Criado pela mão do Engenheiro Jorge China, chefe de divisão do Serviço de Gestão de Sistemas e Infra-estruturas de Informação e Comunicação da Universidade de Coimbra, e desenvolvido utilizando a tecnologia Visual Basic, o XPaga é, neste momento, responsável por toda a gestão de propinas de alunos das Faculdades da Universidade de Coimbra, exceptuando-se o caso referido na secção anterior onde a gestão é feita pela aplicação CXA SiGES.

Desenhado inicialmente apenas para suportar a gestão de propinas de alunos da Faculdade de Ciências e Tecnologia de Universidade de Coimbra, foi sendo alterado de modo a acondicionar as restantes Faculdades e assim acompanhar o desenvolvimento do projecto NONIO.

No entanto, a evolução do processo de gestão de propinas levou a que o número de especificidades aumentasse bem como a complexidade de todo o processo, algo que não estava previsto aquando da criação da aplicação. Assim, foi decidida a criação de um projecto que substituísse o XPaga.

Integração SAP-NONIO

Este projecto, tal como o NONIO-GP, consistia numa integração entre NONIO e SAP-FI, importando todos os dados do XPaga e SiGES, mas com a diferença de que toda a lógica de negócio estaria centralizada em SAP-FI.

A falta de *know-how* em termos de criação de aplicações em SAP levou a que se subcontratasse uma empresa para fazer todos os desenvolvimentos necessários para o processo de gestão de propinas pudesse ser realizado a partir do sistema SAP-FI. Nestas circunstâncias, o sistema NONIO apenas invocaria os *Web Services* criados consumindo a informação e apresentando-a ao utilizador final.

No momento de integrar os dois sistemas percebeu-se que o resultado final não era o esperado e que existiam diversos problemas nas soluções implementadas, sendo tomada a decisão de abortar o projecto. Para essa decisão contribuiu também o facto de que a solução apresentada implicava a dependência de um sistema externo o que significava uma perda de controlo sobre a aplicação além do facto de que qualquer mudança que se pretendesse efectuar teria de ser subcontratada à empresa responsável, o que representaria mais encargos financeiros para a Universidade.

Com o abortar do projecto de substituição do XPaga, ficou também por resolver a necessidade de ter um sistema capaz de responder a todas as exigências que o processo de gestão de propinas apresenta. Assim, nasceu o projecto NONIO-GP.

2.3. Tecnologias de Integração de Sistemas

Um dos aspectos essenciais para o sucesso do projecto NONIO-GP será uma integração bem-sucedida com o módulo de gestão financeira SAP-FI.

Como iremos ver em detalhe mais à frente neste documento, terá de ser realizada uma sincronização diária onde irão ser enviados para SAP-FI pacotes de informação agregada de todas as dívidas e recebimentos registados em NONIO. Por sua vez, o módulo SAP-FI terá de notificar a aplicação NONIO do correcto processamento dos pacotes enviados. Assim, a metodologia através da qual se fará a troca de informação é uma decisão importante para que esta se realize com sucesso, pelo que foram estudadas soluções de integração.

Nos parágrafos seguintes serão descritas quatro metodologias de integração de sistemas, as suas vantagens e as suas limitações.

Transferência de Ficheiros

A ideia base desta metodologia é que cada uma das aplicações produza ficheiros que serão lidos pelas outras. Esta é talvez a solução mais fácil e mais básica para realizar uma integração entre dois sistemas.

Uma das principais decisões que se têm de tomar é qual o formato dos ficheiros. Na maioria das vezes, o formato que uma aplicação produz não é o formato que a aplicação consumidora precisa, sendo necessário transformar os dados para o formato correcto.

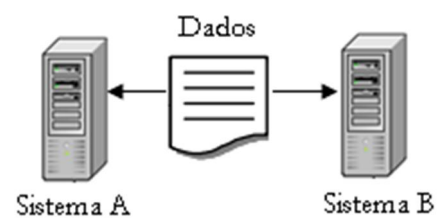


Figura 4: Transferência de Ficheiros

Outra das principais preocupações é decidir com que frequência é que se faz a sincronização de ficheiros. Normalmente, a geração e processamento de um ficheiro é uma tarefa pesada, sendo a decisão da periodicidade com que se executam estas acções um aspecto importante para que este método não comprometa o bom funcionamento de uma aplicação.

É neste aspecto que existem as maiores desvantagens neste método. Se a periodicidade for grande, o risco de inconsistências é muito elevado o que poderá levar a que os utilizadores da aplicação estejam a realizar operações sobre dados que podem já não ser válidos. Isto irá resultar em erros que posteriormente serão difíceis de resolver.

Reduzindo a periodicidade da sincronização, o problema de inconsistências nos dados seria reduzido mas o custo, em termos de performance da aplicação, seria elevado. Além disso, teria de se garantir que todos os ficheiros eram lidos e nenhum era perdido.

Como principais vantagens, esta metodologia garante que as aplicações não ficam dependentes uma da outra. Se uma aplicação não está pronta para receber o ficheiro, este poderá ser guardado e enviado depois. Esta metodologia é também independente da linguagem e plataforma desde que utilize um formato de mensagens comum (por exemplo, XML).

Base de Dados Partilhada

Esta metodologia tem por base o conceito de integrar duas aplicações através da partilha de uma base de dados cujo modelo seja criado de modo a lidar com todas as necessidades das diversas aplicações.

Este método, para além de eliminar a necessidade de se duplicar a informação, garante que todas as aplicações irão ter acesso aos mesmos dados prevenindo assim problemas de inconsistência.

As vantagens aparentes causam, no entanto, diversos problemas.

O primeiro passa pela criação do modelo de dados. Criar um modelo que seja aplicável a todas as aplicações e que garanta que todas as necessidades de todas as aplicações são tidas em conta não é uma tarefa fácil. A incapacidade de criar um modelo deste tipo faz com que muitas vezes seja logo posta de parte esta opção.

O segundo passa pelo facto de ser necessário controlar todo o acesso às tabelas da base de dados quando se pretendem realizar operações de actualização de dados. A obrigatoriedade de fazer *locks* nas tabelas, torna a base de dados num ponto de *bottleneck* que põe em causa todo o bom funcionamento das aplicações.

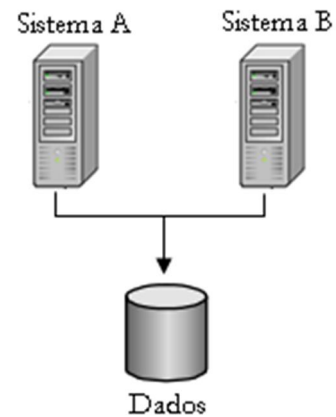


Figura 5: Base de Dados Partilhada

Remote Procedure Call

Esta metodologia aplica o princípio do encapsulamento de dados para integrar aplicações, disponibilizando, cada aplicação, uma interface com métodos disponíveis a serem invocados por outras.

Se uma aplicação necessita de informação que está guardada noutra aplicação, irá ser realizado um pedido directamente a essa aplicação. Se uma aplicação necessita de modificar os dados de outra, essa operação será realizada através da chamada de um método disponibilizado pela aplicação no qual os dados irão ser alterados. Isto permite que cada aplicação mantenha a integridade de dados, e possibilita que internamente, cada aplicação possa alterar o formato dos seus dados sem que isso afecte a outra aplicação.

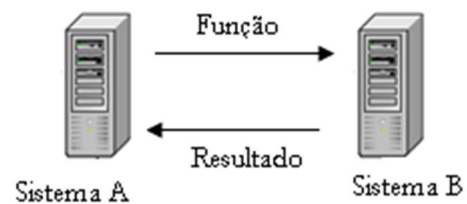


Figura 6: Remote Procedure Call

As tecnologias *CORBA*, *.NET Remoting*, *Java RMI* implementam esta metodologia. Nos dias que correm, uma das tecnologias mais utilizadas na integração de aplicações são os *Web Services*, nomeadamente utilizando *SOAP* e *XML*. Os *Web Services* têm a vantagem de integrarem facilmente com o protocolo *HTTP* facilitando a passagem nas *firewalls*.

Esta solução tem, no entanto, alguns inconvenientes. Nesta metodologia, a comunicação é feita de forma síncrona, ou seja, para que a comunicação decorra todas as aplicações têm de estar activas. Caso a aplicação que recebe um pedido não esteja activa, a aplicação que originou o pedido irá ficar à espera de um *timeout* já que a troca de informação não poderá ocorrer.

Mesmo quando as duas aplicações estão activas, a invocação de um método remoto irá fazer com que essa aplicação fique bloqueada à espera de uma resposta. Este é um dos principais problemas deste método.

Mensagens Assíncronas

Os sistemas de mensagens assíncronas procuram resolver os problemas referidos nos métodos detalhados nas secções anteriores.

Este método garante uma maior independência entre as aplicações, na medida que ao utilizar mensagens assíncronas não necessita que as aplicações estejam todas activas ao mesmo tempo. Deverá existir um serviço de mensagens que irá gerir todo o envio de mensagens, passando as aplicações apenas a preocuparem-se com a informação que devem passar mas não com a maneira como ele é

passada. Assim, a aplicação emissora poderá escolher vários recipientes para a recepção de mensagens.

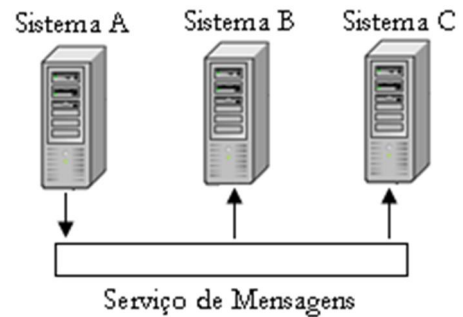


Figura 7: Mensagens Assíncronas

Com o envio constante de dados, muitos dos problemas de consistência de dados que podíamos ver, por exemplo, no método de transferência de ficheiros são resolvidos. No entanto, isso não significa que, por vezes, não existem problemas já que os dados podem não ser actualizados em todas as aplicações ao mesmo tempo.

Este método é visto por muitos como a maneira mais eficaz de realizar integrações entre sistemas.

2.3.1. Metodologia Seleccionada

A escolha da metodologia foi uma decisão conjunta entre os membros do projecto NONIO e os membros da empresa externa responsável pelos desenvolvimentos em SAP-FI.

Da perspectiva da aplicação NONIO a solução ideal passava pela utilização de *webservices SOAP*. Este facto é explicado pelas diversas integrações de sistemas que já foram realizadas utilizando esta metodologia. Nestas inclui-se, inclusive, uma integração com SAP-FI para a sincronização em NONIO da lista de recursos humanos existente em SAP-FI. Outros exemplos, incluem a integração com a aplicação *WOC (Web on Campus)* e a integração diária com o Serviço de Estrangeiros e Fronteiras (SEF).

Do ponto de vista do módulo SAP-FI, foi referido que a melhor solução também passaria pela utilização de *webservices SOAP* já que a criação e gestão de *webservices* é bastante facilitada pelo módulo *Sap Web Application Server* que disponibiliza ferramentas para esse efeito. Em termos práticos, já é realizada uma integração em que SAP-FI invoca um *webservice* de NONIO com o intuito de obter a lista de edições de disciplinas que irão funcionar num determinado ano lectivo.

Ponderadas as metodologias descritas, as suas vantagens e desvantagens, e também os aspectos referidos nos parágrafos anteriores, chegou-se à conclusão que a integração

utilizando *webservices SOAP* seria a solução que melhor se adaptaria ao problema e também às duas aplicações (NONIO e SAP-FI). Foram factores críticos na escolha a simplicidade de implementação, o facto de já se integrarem os dois sistemas com sucesso desta forma e também não existir nenhuma vantagem significativa (para o tipo de integração que se pretende fazer) noutras metodologias que motivasse uma troca.

Capítulo 3

Actividades Desenvolvidas

Neste capítulo serão detalhadas todas as actividades desenvolvidas pelo estagiário ao longo dos dois semestres de duração do estágio.

Sendo o NONIO-GP um projecto começado de raiz, muito do trabalho realizado durante o primeiro semestre passou pela criação de uma base sólida para o desenvolvimento da aplicação. O segundo semestre centrou-se na definição e implementação da integração com SAP-FI e no refinamento e implementação do trabalho realizado no primeiro semestre.

Ao longo das secções seguintes iremos acompanhar o trabalho desenvolvido pelo estagiário paralelamente à evolução de todo o projecto. Mais especificamente serão detalhados os módulos que contaram com a participação do estagiário sendo explicado o processo de desenvolvimento utilizado. Na última secção serão detalhadas outras actividades desenvolvidas pelo estagiário.

No anexo 6 estão discriminadas todas as reuniões de especificação de requisitos e inspecções dos módulos que contaram com o contributo do estagiário. Neste anexo poderão ser visualizadas as datas e durações de outras actividades que contaram com a presença do mesmo.

O diagrama seguinte procura resumir temporalmente o trabalho efectuado no primeiro semestre:

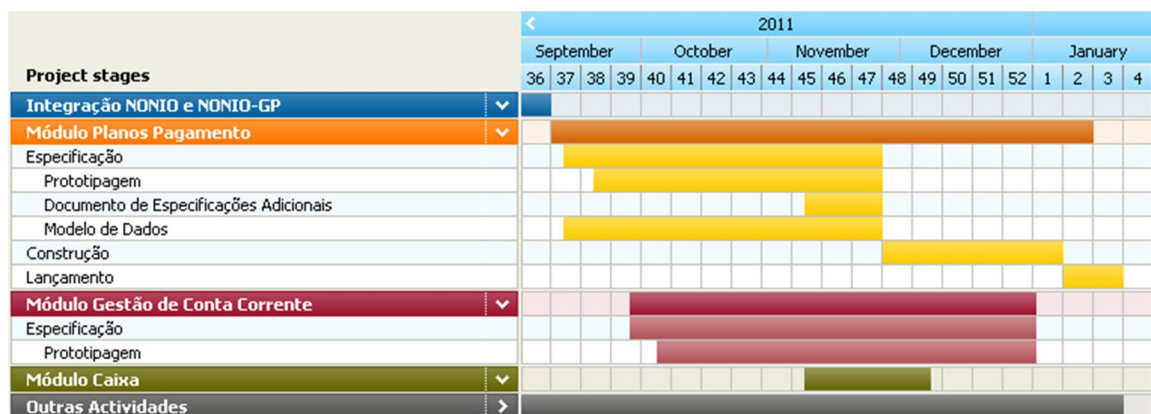


Figura 8: Diagrama de Gantt representativo do trabalho desenvolvido no 1º Semestre

Como podemos verificar existiram várias tarefas que foram realizadas em paralelo. Esta situação deve-se ao elevado grau de interligação entre as diferentes funcionalidades o que obrigava a que existisse um visão global de todos os módulos para que no processo de especificação pudesse ter em conta todos os pontos de contacto entre estes. Isto significa que os módulos de planos de pagamento, gestão de conta corrente e caixa são complementares entre si, tendo de ser coerentes e funcionar como peças de puzzle que encaixam entre si.

Para o segundo semestre, a planificação realizada aquando da entrega do relatório intermédio de estágio foi a seguinte:

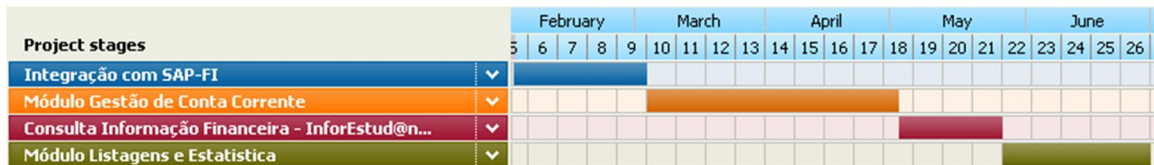


Figura 9: Diagrama de Gantt representativo do trabalho planeado para o 2º Semestre

O planeamento definido passava, numa primeira fase, pela realização dos procedimentos automáticos de integração com SAP-FI. Estava previsto que esta tarefa se alongasse pelo mês de Fevereiro. Durante o mês de Março e Abril estava definido que o estagiário se iria encarregar de refinar o protótipo, escrever o documento de especificações adicionais, implementar e testar a primeira secção do módulo de gestão de conta corrente. Nos últimos dois meses o estagiário estaria encarregue da implementação da consulta de informação financeira na vista *InforEstud@nte* e por último da implementação do módulo de listagens e estatísticas.

No entanto, devido a mudanças no planeamento do próprio projecto NONIO a planificação dos trabalhos do estagiário teve de ser alterada de modo a que se adaptasse às necessidades do projecto.

Assim, numa primeira fase, o estagiário foi responsável pela integração entre NONIO e SAP-FI. Esta tarefa alongou-se mais do que o esperado já que, para além da integração automática entre NONIO e SAP-FI, houve a necessidade de desenvolver uma solução de integração manual da informação presente em NONIO. Este facto adicionado a alguma indefinição por parte da área de gestão financeira sobre alguns aspectos da integração levou a sucessivas mudanças nos requisitos, resultou em que esta tarefa se prolongasse até meio de Março.

No que toca ao módulo de gestão de conta corrente, como foi referido apenas estava previsto o refinamento do protótipo, escrita do documento de especificações adicionais, implementação e testes para o primeiro separador, que corresponde ao plano de pagamento do aluno. No entanto, as já referidas mudanças no planeamento do projecto NONIO levaram a que fosse o estagiário a implementar todo o módulo de gestão de conta corrente.

Em relação à consulta de informação na vista *InforEstud@nte* (módulo de propinas e emolumentos) o estagiário foi responsável pela criação de uma primeira versão do protótipo.

Adicionalmente, o estagiário foi responsável por realizar as alterações ao módulo de Alunos, nomeadamente aos processos de deslacragem de inscrições em frequência, alteração do estado de matrículas e anulação de matrículas. Estas alterações foram necessárias já que as operações listadas são realizadas com bastante frequência e têm influência directa nos dados financeiros do aluno.

A figura seguinte representa o trabalho realizado ao longo de todo o ano lectivo:

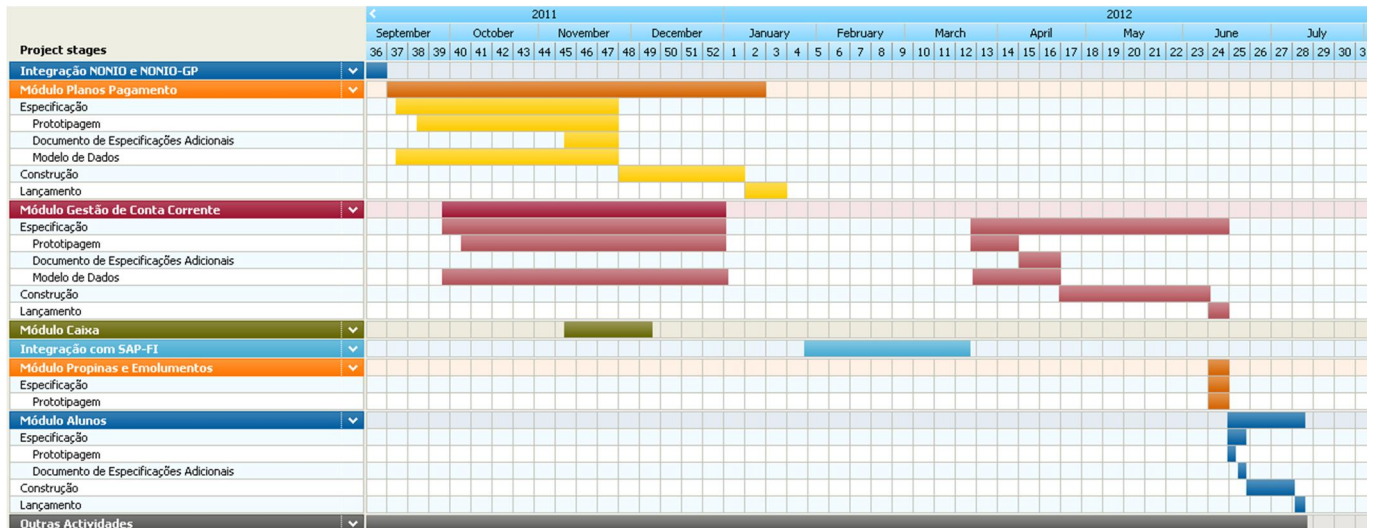


Figura 10: Diagrama de Gantt representativo do trabalho realizado ao longo do estágio.

3.1. Integração no Projecto NONIO e NONIO-GP

Um aspecto fundamental de um projecto da dimensão que o NONIO possui, é que existam regras e processos para a elaboração de tarefas. Estes processos garantem que o trabalho desenvolvido será realizado utilizando linhas directrizes com méritos comprovados e que procuram garantir que o resultado de uma determinada tarefa apresenta um padrão elevado de qualidade.

Tendo isso em conta, a primeira tarefa do estagiário passou pelo estudo do processo de desenvolvimento, do plano de garantia de qualidade e do plano de controlo de alterações, bem como de todas as tecnologias e arquitectura do projecto NONIO. As secções 2.1.1 e 2.1.2 representam um resumo da leitura e estudo realizado pelo estagiário.

Paralelamente à integração no projecto NONIO o estagiário teve também de se integrar no projecto NONIO-GP.

A integração no grupo constituinte do projecto NONIO-GP ocorreu sem problemas. No entanto, o conhecimento do estagiário sobre todo o processo de gestão de propinas era praticamente inexistente pelo que este procurou sempre absorver a maior quantidade de informação possível dos elementos com maior conhecimento sobre a área de negócio em questão. Complementarmente, e para que a integração se processasse de um modo mais rápido, o estagiário estudou o regulamento de gestão de propinas da Universidade de Coimbra [7], bem como todas as situações especiais que podem ser atribuídas a um aluno e que tenham impacto nos dados financeiros deste. No entanto, sendo este um processo tão complexo e cheio de especificidades a aprendizagem de novas situações e pormenores foi uma constante ao longo do estágio.

3.2. Módulo de Planos de Pagamento

O módulo de planos de pagamento foi o primeiro módulo no qual o estagiário centrou as suas atenções. Este módulo foi integralmente realizado pelo estagiário e passou pelas fases de especificação, onde se incluem a prototipagem, escrita do documento de

especificações adicionais e criação do modelo de dados, pela fase de construção e por último, pela fase lançamento.

3.2.1. Especificação

As reuniões especificação de requisitos são um dos aspectos que maior preponderância tem no processo de desenvolvimento de *software*. Uma especificação de requisitos com pouca qualidade poderá comprometer o sucesso do projecto, já que todas as fases seguintes irão centrar-se na procura de soluções para problemas que na realidade não existem ou que não são os problemas em que a solução se deveria focar.

No caso do projecto NONIO-GP, poderemos considerar que os *stakeholders* são representados pelos membros das áreas académica e financeira. Foi com base em reuniões de análise que se agrupou toda a informação necessária para a criação de um protótipo que representa a visão da área informática da informação e requisitos recolhidos.

Numa primeira fase, o objectivo passou por reunir informação e requisitos suficientes sobre o que é que o módulo teria de suportar para que se pudesse dar início à construção do protótipo. Na segunda fase, já com um protótipo inicial, foram discutidas alterações e melhoramentos tanto ao protótipo (algo que será discutido com mais detalhe na secção seguinte) como dos requisitos. Na realidade, a segunda fase acaba por ser uma evolução da primeira onde se inseriu o protótipo como ferramenta de ajuda.

Na tabela seguinte podemos ver uma compilação de todos os requisitos que foram identificados para este módulo:

#	Requisito	Descrição
1	Pesquisar modelos de planos de pagamento.	Deverá ser possível pesquisar por nome, ano lectivo, tipo de plano, tipo de dívida, unidade orgânica e curso.
2	Consultar modelos de plano pagamento pesquisados.	Deverá ser possível consultar os detalhes dos modelos de planos de pagamento pesquisados.
3	Adicionar modelos de planos de pagamento.	Deverá ser possível adicionar modelos de planos de pagamento para um determinado ano lectivo do tipo normal e plurianual. Para o tipo plurianual deverá ser possível definir a sua duração. Adicionalmente, deve ser possível definir se o modelo tem associada uma taxa de inscrição e uma dívida de pré-inscrição.
4	Editar os valores das prestações para os modelos anuais do plano de pagamento.	Deverá ser possível definir o conjunto de prestações para um determinado ano lectivo do modelo de plano de pagamento.
5	Editar detalhes do modelo de plano de pagamento	Deverá ser possível editar os detalhes de um determinado modelo de plano de pagamento. Poderá ser editada o nome, a duração (para modelos plurianuais), e necessidade ou não do pagamento de taxa de inscrição ou dívida de pré-

		inscrição.
6	Obter listagem de alunos afectados por mudanças no modelo de planos de pagamento.	Cada vez que exista uma mudança a um modelo de plano de pagamento e existam planos de pagamento de alunos criados a partir do modelo a alterar, deverá ser fornecida uma listagem desses alunos.
7	Adicionar cursos ao modelo de plano de pagamento.	Deverá ser possível associar um curso ao modelo de plano de pagamento.
8	Remover cursos do modelo de plano de pagamento.	Deverá ser possível desassociar um curso do modelo de plano de pagamento.
9	Remover o modelo de plano de pagamento.	Deverá ser possível eliminar um modelo de plano de pagamento.

Tabela 1: *Lista de requisitos do módulo de planos de pagamento*

Para um melhor entendimento por parte do leitor, nos próximos parágrafos irão ser definidos alguns dos conceitos referidos acima.

Um modelo de plano de pagamento, representa o conjunto de prestações (onde se incluem taxas) que um aluno terá de pagar num determinado ano lectivo caso o seu curso esteja associado a esse modelo. Assim, para um determinado ano lectivo, um curso apenas poderá estar associado a um plano de pagamento.

Um modelo de plano de pagamento poderá ser de dois tipos distintos: Normal e Plurianual. No tipo Normal, o modelo de plano de pagamento é válido apenas para um ano lectivo. No tipo Plurianual, tal como o nome indica, o modelo de plano pagamento será válido durante uma duração definida aquando da criação do mesmo. Cada ano lectivo em que o plano de pagamento é válido representa um modelo anual do plano de pagamento.

O modelo de plano de pagamento terá um tipo de dívida associada que poderá ser do tipo Propina ou Preço. Associado ao plano de pagamento pode existir a necessidade ou não do pagamento de uma taxa de inscrição. Terá ainda de se definir se o plano de pagamento requer ou não o pagamento de uma dívida de pré-inscrição.

Como se pode ver pela tabela acima, o módulo de planos de pagamento irá permitir pesquisar, adicionar modelos de planos de pagamento, editar os detalhes destes, editar os valores das prestações no ou nos modelos anuais em que o plano de pagamento é válido, associar e remover cursos e por fim remover um determinado plano de pagamento.

No processo de associação de um curso há que garantir que este apenas poderá estar associado a um modelo de plano de pagamento para um determinado ano lectivo. O processo de desassociação de um curso apenas poderá ser realizado caso ainda não existam alunos com planos de pagamento gerados a partir do modelo de plano de pagamento. A remoção de um modelo de plano de pagamento segue o mesmo conceito e apenas poderá ser realizada caso não exista nenhum aluno dos cursos associados a este que tenham um plano de pagamento gerado a partir do modelo de plano de pagamento.

Prototipagem

Com base na informação recolhida nas primeiras reuniões de especificação de requisitos foi pedido ao estagiário que iniciasse a construção do protótipo.

Foi neste momento que o estagiário sentiu as primeiras dificuldades. A falta de experiência levou a que a primeira solução apresentada tivesse um *design* algo confuso e não respondesse a todos os requisitos já recolhidos. Esta situação levou o estagiário a repensar o seu método e a analisar toda a estrutura e práticas comuns de diversos protótipos de ecrãs já em funcionamento na aplicação. Este aspecto foi bastante importante pois permitiu que as restantes soluções apresentadas tivessem um grau de qualidade incomparavelmente superior à primeira solução apresentada.

O processo de prototipagem, como já foi referido na secção anterior, corria paralelamente às reuniões de especificação de requisitos. À medida que eram acrescentados ou alterados requisitos, o estagiário era responsável por pensar numa solução para que estes pudessem ser incorporados no protótipo. Durante todo este processo existiram várias indefinições e várias mudanças o que tornou este processo moroso e complexo.

A solução apresentada pelo estagiário passava depois por duas fases. A primeira, consistia numa reunião de apresentação ao gestor de projecto e ao colega João Rafael. Nestas reuniões era discutida a solução apresentada e formas de a melhorar. No final de cada uma das reuniões o estagiário era responsável por fazer as modificações acordadas caso fosse necessário. A segunda fase passava pela apresentação do protótipo aos elementos das áreas financeira e académica, para que estes, como utilizadores finais da aplicação e como maiores conhecedores da área de negócio dessem a sua opinião e sugestões de melhoramento.

Este processo foi repetido até se chegar a uma versão estável do protótipo que, na opinião de todas as áreas representadas, conseguia responder a todas as exigências e requisitos.

Já com uma versão estável do protótipo foi da opinião do estagiário que todo o processo de adicionar e remover prestações continuava confuso e poderia ser melhorado. Naquele momento, a adição e remoção de prestações era feita em ecrãs separados, o que levou a que o estagiário propusesse, à imagem do que já é feito noutra local da aplicação, um único ecrã para gerir este processo. A modificação foi prontamente aceite tanto pelo gestor de projecto como pelos membros das áreas financeira e académica.

No final, o protótipo conta com um total de 11 ecrãs cujo processo de escrita do documento de especificações adicionais irá ser discutido na secção seguinte. O protótipo pode ser encontrado no anexo 8 (*academicos\PlanosPagamento*). Na imagem seguinte apresenta-se um exemplo de um ecrã prototipado:

Início : Planos de Pagamento : Detalhes

Alunos

Dados Pessoais
Dados Curriculares
Propinas e Emolumentos

Inscrições

Matrícula
Frequência
Exame
Estágios Pedagógicos
Situções Especiais
Extensões de Prazos

Atendimento

Documentos
Reclamações
Pagamentos
Infelizmente
Requerimentos

Gestão

Candidaturas
Creditações
Pautas
Estágios e Teses
Notificações
Normas e Procedimentos
Calendários

Administração

Cursos
Disciplinas
Menores
Transições Bolonha
Transições Planos Estudo
Auditoria
Anos Lectivos
Utilizadores
Configurações
Listagens e Estatísticas
Qualidade
Inquéritos
Gestão de Espaços
Lembretes

Financeira

Planos Pagamento
Gestão Conta Corrente
Caixa
Listagens e Estatísticas
Outras Configurações

Planos de Pagamento
Mapa de Definição de Planos
Excepções

[<Remover Plano>](#)
[<Lacrar Plano>](#)
[<Editar Detalhes>](#)

Detalhes

Designação: Plano 1	Ano lectivo: 2011/2012
Tipo de plano: Normal	Tipo de dívida: Propina
Tem Dívida de Pré-Inscrição: Sim, apenas para inscrições 1º ano/1ª vez	Data de lacragem:

Prestações - 2011/2012 <Editar>

Valor total: 1000€	Dívida de pré-inscrição: 250€
--------------------	-------------------------------

Descrição	Valor	Data Limite
Taxa de Inscrição	20€	-
1ª Prestação	250€	31-10-2011
2ª Prestação	250€	31-01-2012
3ª Prestação	250€	31-03-2012
4ª Prestação	250€	31-05-2012
Total:	1020€	

Cursos Aplicáveis - 4 Cursos <Associar Novo>

Ciclo	Designação	U.O.	
1º	Licenciatura em Design e Multimédia	DEI	<Desassociar>
1º	Licenciatura em Engenharia Informática	DEI	<Desassociar>
2º	Mestrado em Design e Multimédia	DEI	<Desassociar>
2º	Mestrado em Engenharia Informática	DEI	<Desassociar>

[Voltar](#)

Figura 11: Ecrã prototipado do módulo de planos de pagamento

Documento de Especificações Adicionais

Quando o protótipo atingiu uma fase estável iniciou-se a escrita do documento de especificações adicionais do módulo de plano de pagamentos. Neste é detalhado todo o modo de funcionamento do módulo, sendo particularmente alvo de análise as situações que não são explícitas no protótipo, tal como validações de campos e todas as consequências da realização de uma determinada acção.

De modo a facilitar a tarefa, o estagiário decidiu ler e estudar a especificação de um módulo realizada por um elemento do NONIO com mais experiência. A especificação escolhida foi a do módulo de Calendários. Esta opção revelou-se acertada pois o estudo do documento tornou a tarefa bastante mais simples já que a leitura do mesmo permitiu ao estagiário entender o processo utilizado e perceber as práticas comuns utilizadas na escrita deste tipo de artefacto.

Com o término da escrita do documento, foi realizada uma reunião de inspecção do documento produzido onde participaram os membros das áreas académica e financeira pertencentes ao grupo do projecto NONIO-GP. Este género de reunião não é uma prática comum no processo de desenvolvimento do NONIO já, que na maioria das funcionalidades implementadas, os membros possuem conhecimento suficiente da área de negócio para que a reunião de inspecção possa decorrer sem a presença de membros de outras áreas. No entanto, dado o carácter crítico da aplicação e a falta de experiência dos elementos do NONIO na área de negócio das propinas foi considerado fundamental a realização desta reunião para que se garanta a qualidade do documento de especificações adicionais produzido.

Neste género de reuniões, o documento é lido em voz alta por um dos participantes, tendo os outros a missão de detectar erros, incorrecções e ambiguidades que possam existir.

Após a reunião de inspecção do documento de especificações adicionais em que participaram os membros das áreas financeira e académica, foi ainda realizada uma inspecção formal em que apenas participaram elementos pertencentes ao projecto NONIO. Esta reunião, para além de ser uma última revisão ao documento, é necessária para que, segundo o plano de controlo de alterações vigente no NONIO, o documento possa passar ao estado entregue.

O documento produzido pode ser encontrado no anexo 3.

Modelo de Dados

A figura 5 representa o modelo de dados que suportará todo o módulo de planos de pagamento, tendo sido definido paralelamente a todas as fases descritas anteriormente.

Na figura poderemos ver que um modelo de plano de pagamento tem associado um ou mais modelos anuais de planos de pagamento que representam os anos lectivos em que o plano é válido. Por sua vez um modelo anual tem associado um conjunto de prestações. Ao modelo de plano de pagamento são associados um ou mais cursos.

Na tabela “CONFIG_FINANCEIRAS_AL” é definido o valor da taxa de inscrição para um determinado ano lectivo.

A tabela “PLANOS_PAGAMENTO_ANUAL_ALU” faz a ligação entre a matriculas de alunos e o modelo de plano de pagamento associado a essa matricula.

Quando existem modificações que resultam na alteração do estado da matricula do aluno para financeiramente inconsistente, os motivos dessa alteração são guardados na tabela “OBSERVACOES_FINANCEIRAS” sendo depois mostrados no contexto do plano de pagamento de um determinado aluno no módulo gestão de conta corrente.

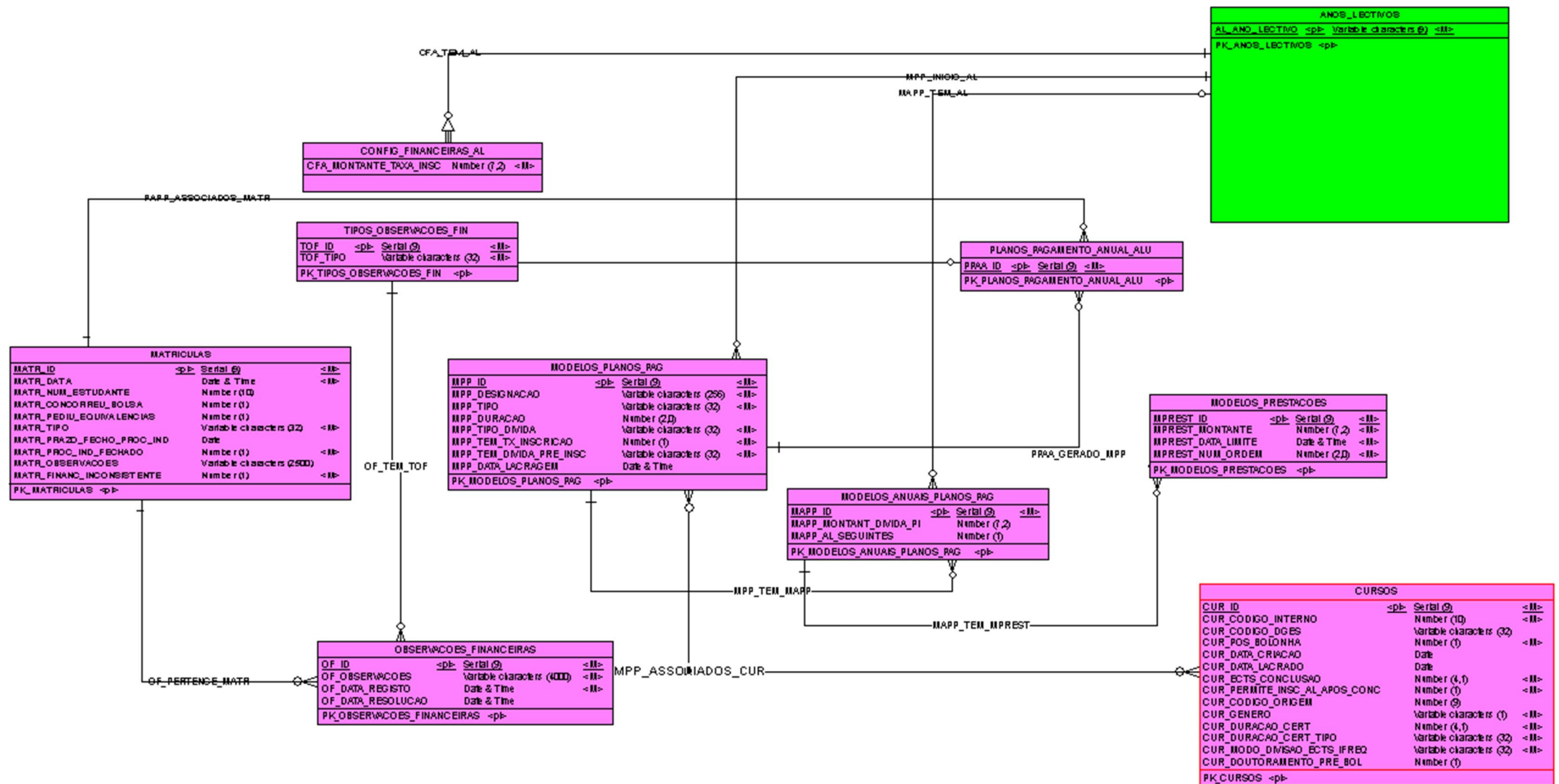


Figura 12: Modelo de Dados Associado ao Módulo de Planos de Pagamento

3.2.2 Construção

Com a finalização da fase de especificação, estavam reunidas todas as condições para que o estagiário passasse à implementação do módulo.

Iniciada na última semana de Novembro e finalizada na primeira semana de Janeiro, a implementação deste módulo ofereceu um desafio interessante e motivador ao estagiário.

Sem nunca ter trabalhado com duas das tecnologias utilizadas no trabalho, *Struts* e *Hibernate*, a primeira etapa da tarefa passou pelo estudo e assimilação dos conceitos inerentes às duas tecnologias. A integração com a tecnologia *Struts* foi sem a dúvida a mais complicada. Todo o processo de configuração que é necessário realizar cada vez que se necessita de criar algum serviço, *action*, *formbean* e os seus respectivos *mappings* requer alguma prática e experiência.

Os serviços (responsáveis pelo acesso à camada de dados) têm de definir quais os *DAO's* (*data access objects*) e eventualmente outros serviços que irão ser utilizados. Na figura seguinte encontra-se um exemplo da definição de um serviço:

```
<bean id="planospagamentoPlanosPagamentoTarget" class="nonio.service.planospagamento.PlanosPagamentoService" >
  <property name="modeloPlanoPagDAO" ref="extendedModeloPlanoPagDAO" />
  <property name="procPesquisaPlanosPag" ref="procPesquisaPlanosPag" />
  <property name="configFinanceiraALDAO" ref="extendedConfigFinanceiraALDAO" />
  <property name="modeloAnualPlanoPagDAO" ref="modeloAnualPlanoPagDAO" />
  <property name="modeloPrestacaoDAO" ref="modeloPrestacaoDAO" />
  <property name="uoResponsavelCursoDAO" ref="extendedUoResponsavelCursoDAO" />
  <property name="cursoDAO" ref="cursoDAO" />
  <property name="planoPagamentoAnualAluDAO" ref="extendedPlanoPagamentoAnualAluDAO" />
  <property name="matriculaDAO" ref="matriculaDAO" />
  <property name="observacaoFinanceiraDAO" ref="observacaoFinanceiraDAO" />
  <property name="tipoObservacaoFinDAO" ref="tipoObservacaoFinDAO" />
  <property name="inscrFrequenciaDAO" ref="extendedInscrFrequenciaDAO" />
  <property name="logOperacoesService" ref="securityLogOperacoesServiceTarget" />
</bean>
```

Figura 13: Definição de um serviço

Por seu lado, as acções (que funcionam como *proxy* entre os serviços e a camada de apresentação) têm de definir explicitamente quais os serviços que utilizam. Um exemplo pode ser encontrado na seguinte figura:

```
<!-- PESQUISA PLANOS PAGAMENTO -->
<bean name="/planospagamento/pesquisaPlanosPagamento" class="nonio.action.planospagamento.planos.PesquisaPlanosPagamentoAction">
  <property name="planosPagamentoService" ref="planospagamentoPlanosPagamentoTxManager"/>
  <property name="calendariosCandidaturasService" ref="calendariosCandidaturasServiceTxManager" />
</bean>
```

Figura 14: Definição de uma acção

Tem ainda de ser definido o mapeamento entre a acção que se acabou de criar e os *jsp's* que são invocados bem como com outras acções que podem ser invocadas a partir da nova acção. Na figura seguinte é apresentado um exemplo do mapeamento:

```
<!-- PESQUISA PLANOS PAGAMENTO -->
<action path="/pesquisaPlanosPagamento"
  type="org.springframework.web.struts.DelegatingActionProxy"
  parameter="method"
  name="pesquisaPlanosPagamentoFormBean"
  validate="false"
  scope="session"
>
  <forward name="pesquisaPlanosPagamento" path="/WEB-INF/planospagamento/planos/pesquisaPlanosPagamento.jsp"/>
  <forward name="accaoDetalhesPlanosPagamento" path="/planospagamento/detalhesPlanosPagamento.do" redirect="true"/>
  <forward name="accaoAdicionarPlanosPagamento" path="/planospagamento/adicionarPlanosPagamento.do" redirect="true"/>
  <forward name="accaoPesquisaPlanosPagamento" path="/planospagamento/pesquisaPlanosPagamento.do" redirect="true"/>
</action>
```

Figura 15: Definição do mapeamento de uma acção

Por último, é necessário definir o *formbean* que possibilita a passagem de dados entre a camada de apresentação e a acção. Na figura seguinte é apresentado um exemplo da definição de um *formbean*:

```
<form-bean name="pesquisaPlanosPagamentoFormBean" type="org.apache.struts.validator.DynaValidatorForm">
  <form-property name="designacao" type="java.lang.String" />
  <form-property name="anoLectivo" type="java.lang.String"/>
  <form-property name="anosLectivos" type="java.util.ArrayList"/>
  <form-property name="tipoPlano" type="java.lang.String"/>
  <form-property name="tipoDivida" type="java.lang.String"/>
  <form-property name="unidadeOrganica" type="java.lang.String"/>
  <form-property name="unidadesOrganicas" type="java.util.ArrayList"/>
  <form-property name="cursos" type="java.util.ArrayList"/>
  <form-property name="curso" type="java.lang.String"/>
</form-bean>
```

Figura 16: Definição de um *formbean*

Assim, nos primeiros dias de implementação foi perdido algum tempo com a resolução de erros resultantes de incorrecções nas configurações. Como seria de esperar, com a repetição deste processo, a configuração deixou de ser um problema. No que toca à integração com a tecnologia *Hibernate*, a assimilação de conceitos foi bastante mais simples já que esta já tem como principal objectivo facilitar todo o acesso à camada de dados, sendo apenas necessário estudar as diferenças mais relevantes entre a linguagem SQL e a linguagem HQL (*Hibernate Query Language*).

Como segunda etapa do processo de implementação, o estagiário decidiu seguir o método utilizado nas fases de prototipagem e especificação. Assim, foi analisada a implementação de módulos já em produção e foi estudada toda a estrutura e práticas comuns utilizadas pelos membros mais experientes do projecto NONIO. Esta decisão permitiu que a implementação se tornasse mais fácil e ficasse de acordo com os padrões normalmente utilizados no projecto NONIO.

Após a sistematização do processo de criação de *actions*, *formbeans* e os seus respectivos *mappings* as principais dificuldades encontradas pelo estagiário centraram-se no ecrã que permite gerir o número de prestações de um determinado modelo anual. O cariz dinâmico desse ecrã criou problemas a nível da passagem de valores da camada de apresentação para a camada de controlo. Num formulário normal, utilizando *Struts*, o

programador não se tem de preocupar com a criação explícita de um *formbean* (tendo apenas de definir o seu formato), estando esse trabalho ao cargo da *framework*. Sendo o número de prestações um valor dinâmico e tendo cada prestação associado um valor e uma data limite, o *Struts* não oferece nenhuma maneira nativa para a realização desta tarefa. Assim, o estagiário teve de estudar uma solução que permitisse lidar com este caso, solução essa que foi encontrada com sucesso, tendo o estagiário criado um *formbean* de raiz para permitir a passagem de dados entre camadas com sucesso. De referir que a solução encontrada pelo estagiário irá ser utilizada por outros elementos do NONIO em situações semelhantes.

Outra dificuldade encontrada foi o facto de durante a fase de implementação a atenção do estagiário não estar apenas centrada na codificação, estando, na realidade, dividida pela especificação de requisitos e prototipagem dos módulos caixa e gestão de conta corrente. No entanto, esta foi uma experiência que certamente irá resultar numa melhoria da capacidade do estagiário para lidar com várias exigências ao mesmo tempo.

A fase de construção deu também origem à escrita de um pedido de alteração ao documento de especificações adicionais e ao protótipo já que, durante o processo de codificação, foram encontradas algumas situações que não tinham sido contempladas aquando da escrita inicial do documento. O pedido de alteração poderá ser encontrado no anexo 4.

Na figura seguinte são apresentadas o número de linhas de código produzidas pelo estagiário neste módulo. Apesar de este ser um indicador que pode ser considerado subjectivo (um elevado número de linhas não significa necessariamente uma complexidade elevada), permite ter uma ideia geral do volume de trabalho que o módulo exigiu em termos de implementação. Estes valores têm também como objectivo possibilitar uma comparação entre as diversas funcionalidades implementadas pelo estagiário.

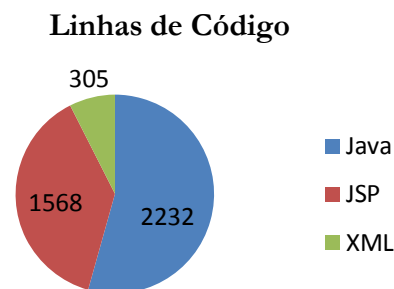


Figura 17: Número de linhas produzidas na implementação do módulo de planos de pagamento

3.2.3 Lançamento

Nesta fase, a implementação realizada pelo estagiário foi sujeita a testes funcionais por parte de outro elemento do NONIO, com o objectivo de se detectarem erros na implementação. Esta foi uma experiência pela qual o estagiário nunca tinha passado e apesar de não terem sido encontrados erros críticos, foram encontrados alguns erros relacionados com pormenores aos quais não foi dada a devida atenção pelo estagiário. Entre os erros assinalados encontram-se falta de validações em alguns campos e perda de contexto entre ecrãs.

Nesta fase, o estagiário apercebeu-se das diferenças entre a realização de um trabalho universitário e a realização de um trabalho num contexto real. Num trabalho universitário é, muitas vezes, dada pouca atenção aos pormenores, estando a preocupação mais centrada numa visão global do trabalho. Num ambiente de trabalho real, tal não pode acontecer.

Todo o pormenor conta e não pode ser descurado pelo que alguns dos erros encontrados revelaram alguma falta de experiência do estagiário. No entanto, esta situação resultou numa evolução do estagiário como programador e os erros detectados nesta fase não voltaram a ser cometidos.

Após a correcção dos erros por parte do estagiário, foi disponibilizado o módulo para que fossem definidos todos os modelos de planos de pagamento do ano lectivo 2012/2013. Estes são peça essencial para que se possam criar os planos de pagamento dos alunos que se inscrevam no ano lectivo referido.

No momento da escrita do relatório, já foram criados 26 modelos de planos de pagamento que cobrem todos os cursos existentes na Universidade de Coimbra.

Na imagem seguinte é apresentado um ecrã do módulo já em produção:

Planos de Pagamento

Detalhes

Designação: Plano Teste Ano Lectivo: 2012/2013
 Tipo de Planos: Plurianual (2 Anos) Tipo de Dívida: Propina
 Tem Dívida de Pré-Inscrição: Sim, apenas para inscrições 1º ano/1ª vez Data de Lacragem:

Prestações - 2012/2013 Editar

Valor Total: 1000.00 € Dívida de Pré-Inscrição: 0.00 €

Descrição	Montante	Data Limite
Taxa de Inscrição		
1ª Prestação	500.00 €	15-02-2012
2ª Prestação	500.00 €	15-02-2012
Total:	1000.00 €	

Prestações - 2013/2014 Editar

Valor Total: 1000.00 €

Descrição	Montante	Data Limite
Taxa de Inscrição		
1ª Prestação	1000.00 €	12-12-2014
Total:	1000.00 €	

Prestações - Anos Lectivos Seguintes Editar

Não existem prestações definidas.

Cursos Aplicáveis - 3 Cursos Associar Novo

Ciclo	Designação	U.O.	
2º Ciclo	Mestrado em Crítica de Arte e Arquitectura	COLARTES	Desassociar
2º Ciclo	Mestrado em Estudos Curatoriais	COLARTES	Desassociar
3º Ciclo	Doutoramento em Arte Contemporânea	COLARTES	Desassociar

Voltar

Figura 18: Ecrã do módulo de planos de pagamento já em produção

3.3. Módulo de Gestão de Conta Corrente

Na secção anterior foi discutido o módulo de planos de pagamento, onde se criam os modelos que darão origem aos planos de pagamento específicos de cada aluno.

Nesta secção irá ser detalhado o módulo que permite consultar e gerir o plano de pagamento criado para cada aluno. É seguro afirmar que este módulo, integralmente desenvolvido pelo estagiário, é o que mais complexidade tem associada sendo, por isso, todo o processo de especificação de importância crítica.

3.3.1 Especificação

O processo de especificação deste módulo decorreu de maneira semelhante à descrita para o módulo de planos de pagamento.

Para um melhor entendimento do módulo é de todo o interesse a definição prévia de alguns conceitos:

- **Prestação** – Representa um item que pode ser pago. Poderá ter diversos tipos onde se incluem propinas, certidões, certificados, etc.
- **Fracção** – Uma prestação poderá ter uma ou mais fracções. Associado a fracções poderão existir juros. Uma fracção terá uma referência associada.
- **Referência** - Representa o que pode ser compensado para pagar uma ou várias fracções.
- **Entidade Pagadora** – Entidade responsável por uma prestação ou por um recebimento. Na maioria dos casos será o próprio aluno mas poderão existir terceiras entidades pagadoras onde se incluem empresas, fundações etc.
- **Recebimento** – É o pagamento de um determinado montante por parte de um aluno ou terceira entidade pagadora. O recebimento terá um meio de pagamento associado e uma data de pagamento.
- **Compensação** – É o pagamento de uma referência. É constituída por um ou mais recebimentos.
- **Operação de Compensação** - Agrega uma ou várias compensações. Uma operação de compensação pode ter um recibo associado.
- **Reposição** – Representa o montante de um recebimento que é reembolsado ao aluno.
- **Crédito de um Recebimento** – É calculado a partir da fórmula:
 $Crédito = MontanteTotal - (\sum CompensaçõesValidas + MontantesRepostos)$
- **Matrícula Financeiramente Inconsistente** – Matrículas cujos dados financeiros poderão não estar correctos.
- **Ocorrência Financeira** – Mensagem a detalhar o motivo pelo qual a matrícula foi colocada no estado financeiramente inconsistente.

A partir deste módulo será possível visualizar seis grandes conjuntos de informação, divididas em seis secções. São eles: o plano de pagamento do aluno, os itens a pagamento, a conta corrente do aluno, as compensações, os recebimentos e as reposições.

Nos parágrafos seguintes serão detalhados quais as ideias base que sustentam as secções referidas.

Plano Pagamento

A tabela seguinte representa o conjunto de requisitos levantados para esta secção:

#	Requisito	Descrição
1	Consultar plano de pagamento.	Deverá ser apresentada informação sobre a matrícula, inscrição em frequência, lista de prestações, plano de pagamento modelo e situações especiais.

2	Consultar ocorrências financeiras.	Deverá ser possível visualizar todas as ocorrências financeiras associadas à matrícula.
3	Consultar detalhes de uma prestação.	Consultar detalhes de uma prestação onde se incluem as fracções constituintes da prestação, referências anuladas e detalhes sobre a importação (se aplicável).
4	Consultar prestações anuladas.	Deverá ser possível consultar uma listagem com todas as prestações anuladas.
5	Gerir ocorrências financeiras.	Deverão existir mecanismos de gestão das ocorrências financeiras onde se inclui a adição, resolução e anulação de resolução de ocorrências financeiras.
6	Adicionar Prestações	Deverá ser possível adicionar prestações do tipo propinas, taxa de inscrição, multas e juros importados, definindo as suas fracções e data limite.
7	Editar Prestações	Deverá ser possível editar as fracções e data limite de uma determinada prestação.
8	Anular Prestações	Deverão existir funcionalidades de anulação de prestações em bloco e individualmente.
9	Regularizar Prestações	Deverá ser possível compensar prestações utilizando o crédito que o aluno tem disponível.

Tabela 2: *Lista de requisitos da secção plano de pagamento do módulo de gestão de conta corrente*

Como podemos verificar pela tabela acima, nesta secção será possível consultar e gerir o plano de pagamento do aluno numa determinada matrícula.

Em termos de consulta são apresentados ao utilizador dados sobre a matrícula do aluno (incluindo o estado financeiro da matrícula), inscrição em frequência do aluno, prestações e fracções do plano de pagamento, o modelo de plano de pagamento utilizado na criação do plano de pagamento específico do aluno bem como todas as situações especiais com impacto financeiro.

Em termos de gestão do plano de pagamento será possível gerir o estado financeiro de uma matrícula. Para isso são disponibilizadas funcionalidades de adicionar, resolver e anular a resolução de ocorrências financeiras. De referir que uma matrícula é considerada no estado “Financeiramente Inconsistente” caso exista pelo menos uma ocorrência não resolvida. O utilizador terá a possibilidade de resolver ocorrências individualmente ou, caso a resolução seja semelhante para várias ocorrências, em bloco.

A gestão do estado financeiro da matrícula do aluno é um aspecto bastante importante já que para matrículas no estado “Financeiramente Inconsistente” existem determinadas operações sobre os dados financeiros do aluno que não poderão ser realizadas. Por exemplo, quando uma matrícula está no estado referido não será possível compensar prestações nas tesourarias. Esta situação é facilmente explicada já que quando uma matrícula se encontra no estado “Financeiramente Inconsistente” tal poderá significar que os montantes das prestações poderão estar desactualizados pelo que permitir a compensação de prestações nestas condições iria potenciar erros.

Os utilizadores poderão também anular prestações individualmente ou em bloco. Quando se anulam prestações parcialmente ou totalmente compensadas o valor pago pelo aluno ser-lhe-á creditado.

A adição de prestações é também umas das funcionalidades presentes nesta secção. Será possível adicionar quatro tipos de prestações: Propinas, Taxa de Inscrição, Juros Importados e Multas.

Dependendo do seu tipo as prestações podem ser consideradas como fraccionáveis ou não. Propinas e Juros Importados são consideradas prestações fraccionáveis pelo que poderão ter mais do que uma fracção. Cada fracção poderá estar associada a uma entidade pagadora diferente o que significa que a responsabilidade do pagamento da prestação pode estar dividida por várias entidades. Por outro lado, Taxa de Inscrição e Multas são consideradas não fraccionáveis, pelo que apenas terão uma fracção.

As prestações criadas poderão também ser editadas. Apenas poderão ser alteradas fracções totalmente por compensar ou seja, fracções às quais não esteja associada nenhuma compensação. O utilizador poderá também alterar a data limite de uma prestação o que irá forçar o recálculo dos juros.

Por último, irá ser possível regularizar dívidas do plano de pagamento. Existem várias situações onde o aluno poderá ficar com crédito, como por exemplo na anulação de uma prestação já compensada ou na anulação de uma operação de compensação. Esta funcionalidade permite a utilização desse crédito para compensar prestações do plano de pagamento do aluno.

Itens a Pagamento

Nesta secção é possível ver todas as prestações por compensar associadas ao aluno seleccionado. Para cada prestação é possível consultar o que falta pagar sendo dada a indicação de se a prestação já venceu ou não.

Será apresentada uma secção por cada matrícula associada ao aluno que tenha prestações por compensar. Serão ainda mostradas secções de disciplinas isoladas e outros emolumentos caso o aluno tenha prestações por compensar que se incluam nestes dois grupos.

Conta Corrente

Nesta secção serão listadas todas as operações realizadas sobre os dados financeiros de aluno. Funcionará como um extracto de multibanco oferecendo uma maneira rápida e eficaz de perceber todas as alterações realizadas aos dados financeiros do aluno.

Compensações

Na secção de compensações poderão ser consultadas todas as operações de compensação associadas a uma determinada pessoa.

Na tabela seguinte estão compilados todos os requisitos recolhidos para esta secção:

#	Requisito	Descrição
1	Pesquisar compensações.	Deverá ser possível pesquisar operações de compensações por período temporal, por modo de pagamento, por posto de recebimento e por

		utilizador responsável pelo registo da operação de compensação.
2	Consultar detalhes de uma compensação.	Deverá ser possível consultar os detalhes de uma compensação. Na informação a apresentar deverão incluir as prestações compensadas bem como os recebimentos utilizados na operação de compensação.
3	Emitir Recibos.	A aplicação deverá possibilitar a emissão de recibos originais e segundas vias.
4	Anular Operação de Compensação.	O utilizador poderá anular uma operação de compensação sendo o valor total pago creditado ao aluno.

Tabela 3: *Lista de requisitos da secção de compensações do módulo de gestão de conta corrente*

Como podemos verificar, o utilizador poderá pesquisar por operações de compensação e consultar os seus detalhes. A informação detalhada da operação de compensação incluirá quais as prestações compensadas bem como quais os recebimentos utilizados nas compensações. Para cada operação de compensação será dada ao utilizador a possibilidade de emitir um recibo comprovativo da operação.

Em termos de processos de gestão, o utilizador poderá anular uma operação de compensação. Isso significa que todas as compensações associadas à operação irão ser anuladas. No entanto, os recebimentos utilizados na operação de compensação continuarão válidos permanecendo como crédito associado ao aluno.

Recebimentos

Nesta secção o utilizador poderá consultar todos os recebimentos associados a uma determinada pessoa.

Na tabela seguinte estão resumidos todos os requisitos recolhidos para esta secção:

#	Requisito	Descrição
1	Pesquisar recebimentos.	Deverá ser possível pesquisar recebimentos por período temporal, tipo de meio de pagamento, número do meio de pagamento e entidade pagadora à qual o recebimento está associado.
2	Consultar detalhes de um recebimento.	Deverá ser possível consultar os detalhes de um recebimento. A informação a apresentar deverá incluir as operações de compensação onde o recebimento é utilizado. Além disso, e caso existam, devem ser apresentadas as reposições associadas ao recebimento.
3	Anular recebimentos.	A aplicação deverá possibilitar a anulação de um recebimento sendo as operações de compensação onde o recebimento é utilizado também anuladas.

4	Alterar entidade pagadora responsável pelo recebimento.	O utilizador poderá alterar a entidade pagadora responsável por um determinado recebimento.
5	Transferir recebimentos entre alunos.	Deverá ser possível transferir um recebimento associado a um aluno para outro aluno diferente.

Tabela 4: *Lista de requisitos da secção de recebimentos do módulo de gestão de conta corrente*

O utilizador poderá pesquisar por recebimentos e consultar os seus detalhes. A informação detalhada do recebimento incluirá as compensações onde o recebimento ou parte do recebimento é utilizado bem como todas as reposições associadas a este.

Em termos de processo de gestão será possível anular um recebimento, sendo todas as operações de compensação associadas a este também anuladas.

Nesta secção será também possível transferir recebimentos entre alunos. O utilizador poderá seleccionar quais os recebimentos que pretende transferir bem como qual o aluno para o qual pretende transferir os recebimentos seleccionados. O utilizador decidirá depois se pretende compensar alguma prestação do aluno para o qual os recebimentos irão ser transferidos ou se, por outro lado, pretende que os recebimentos apenas sejam transferidos não sendo compensada qualquer prestação. Caso decida não compensar nenhuma prestação, o utilizador terá de seleccionar qual a entidade pagadora a que o recebimento ficará associado (se ao aluno para quem o recebimento irá ser transferido, se para uma terceira entidade pagadora). No final, as compensações às quais o recebimento estava associado (compensações do aluno de quem os recebimentos serão transferidos) serão anuladas sendo realizadas as novas compensações seleccionadas pelo utilizador ou sendo apenas transferido o recebimento para o novo aluno.

No seguimento da funcionalidade de transferir recebimentos foi detectada a necessidade de existir a possibilidade de alterar a entidade pagadora responsável associada a um determinado recebimento. Esta funcionalidade será bastante importante para a resolução de erros na transferência de recebimentos entre alunos.

Reposições

Nesta secção será possível consultar todas as reposições associadas aos recebimentos de uma determinada pessoa.

Na tabela seguinte estão representados os requisitos recolhidos:

#	Requisito	Descrição
1	Consultar reposições.	Consultar todas as reposições associados a recebimentos de um determinado aluno.
2	Adicionar reposições.	O utilizador deverá poder repor um determinado valor de um recebimento com crédito.
3	Anular reposições.	Deverá ser possível anular uma reposição.
4	Editar reposições.	Deverá ser possível editar os detalhes de uma reposição.

Tabela 5: *Lista de requisitos da secção de reposições do módulo de gestão de conta corrente*

Como podemos verificar o utilizador poderá ver todas as reposições associadas a uma determinada pessoa bem como os detalhes dessa reposição.

O utilizador poderá também adicionar reposições se a pessoa em contexto estiver associada a recebimentos com crédito. A anulação de reposições deverá ser também possível, sendo o valor da reposição anulada creditado ao aluno. Por fim, o utilizador poderá editar os detalhes de uma determinada reposição.

Prototipagem

A prototipagem deste módulo foi, sem qualquer dúvida, um dos aspectos mais desafiantes das tarefas desenvolvidas pelo estagiário.

Aprendendo com os erros cometidos no início da prototipagem do módulo anterior, o estagiário procurou sempre oferecer uma solução cujo *design* fosse claro e que correspondesse a todos os requisitos. A missão, no entanto, não foi fácil. A constante mudança, adição e melhoramento dos requisitos analisados levaram a que o protótipo tivesse de sofrer constantes actualizações para acomodar a nova informação.

No final, o protótipo, que pode ser encontrado no anexo 8 (*academicos\ContaCorrente*), conta com 38 ecrãs dos quais poderemos ver alguns exemplos nas imagens seguintes:

Início : Pesquisa Alunos : Escolher Matrícula : Plano de Pagamento : Adicionar Prestação

Alunos
 Dados Pessoais
 Dados Curriculares
 Propinas e Emolumentos

Inscrições
 Matrícula
 Frequência
 Exame
 Estágios Pedagógicos
 Situações Especiais
 Extensões de Prazos

Atendimento
 Documentos
 Reclamações
 Pagamentos
 Infoestudante
 Requerimentos

Gestão
 Candidaturas
 Creditações
 Pautas
 Estágios e Teses
 Notificações
 Normas e Procedimentos
 Calendários

Administração
 Cursos
 Disciplinas
 Menores
 Transições Bolonha
 Transições Planos Estudo
 Auditoria
 Anos Lectivos
 Utilizadores
 Configurações
 Listagens e Estatísticas
 Qualidade
 Inquéritos
 Gestão de Espaços
 Lembretes

Financeira
 Planos Pagamento
 Gestão Conta Corrente
 Caixa
 Listagens e Estatísticas
 Outras Configurações

José Manuel Silva (2006125104)
 Licenciatura em Engenharia Informática
 Mudar Curso | Mudar Aluno

Plano de Pagamento Itens a Pagamento Conta Corrente Compensações Recebimentos Reposições

Atenção!
 Ao seleccionar o botão gravar irá alterar o estado financeiro da matrícula para financeiramente inconsistente.

Adicionar Prestação

Tipo de Prestação: Propina/Preço
 Taxa de Inscrição
 Juros Importados
 Multa

Descrição: 1ª Prestação

Data Limite: * dd-mm-aaaa

Observações:

* Campos de preenchimento obrigatório

Fracções Remover Última Entrada

Entidade Pagadora	Montante
Aluno	<input type="text"/> €
FCT	<input type="text"/> €

Adicionar Nova Fração

Total da Prestação: 0 €

Figura 19: Ecrã prototipado do módulo de gestão de conta corrente

Início : Pesquisa Alunos : Reposições : Adicionar Reposição

José Manuel Silva (2006125104)
Mudar Aluno

Planos de Pagamento Itens a Pagamento Conta Corrente Compensações Recebimentos **Reposições**

Adicionar Reposição

Observações: *

Despachos Aluno:

Nº do Despacho: *

Data do Despacho: * dd-mm-aaaa

Comprovativo da Reposição: Nenhum ficheiro selecionado

* Campos de preenchimento obrigatório

Seleccionar Recebimentos a Repor

Recebimentos Associados a:

Meio Pagamento	Data Registo	Montante		Crédito	Montante a Repor
		Recebido	Utilizado		
PayPal	27-09-2011	500.00 €	350.00 €	150.00 €	<input type="text"/> € <input type="checkbox"/>
TPA	27-09-2011	350.00 € ^{a)}	134.00 €	216.00 €	<input type="text"/> € <input type="checkbox"/>
Total Reposto:					0 €

^{a)} O valor apresentado contempla reposições já efectuadas

Figura 20: Ecrã prototipado do módulo de gestão de conta corrente

Documento de Especificações Adicionais

O estagiário foi também responsável pela escrita do documento de especificações adicionais onde são detalhados todos os pormenores que não são visíveis no protótipo tal como validações e restrições.

Aquando do término da escrita do documento foi seguido o mesmo processo de revisão e aceitação descrito para o módulo de planos de pagamento. Este processo incluiu reuniões com os membros das áreas de gestão financeira e académica numa primeira fase e posteriormente, para a aceitação formal do documento, reuniões de inspecção com membros do projecto NONIO.

O documento produzido pode ser encontrado no anexo 3.

Modelo de Dados

Paralelamente a todas as fases descritas acima foi definido o modelo de dados necessário para suportar todos os requisitos especificados. Como pudémos verificar este módulo é bastante extenso e complexo o faz com que o próprio modelo de dados também partilhe essas características.

O modelo de dados pode ser encontrado no anexo 8 (neste anexo é apresentado todo o modelo de dados do NONIO, estando as entidades relevantes localizadas do lado esquerdo a rosa).

3.3.2 Construção

Nesta fase o estagiário já estava perfeitamente à vontade com as especificidades das tecnologias utilizadas no projecto. Este facto levou a que as dificuldades encontradas aquando da construção do módulo de planos de pagamento não tivessem sido um factor.

A criação de serviços, *actions*, *formbeans* seguiu as considerações descritas no módulo de planos de pagamento. No entanto, para a construção deste módulo, o estagiário teve de realizar o mapeamento entre as classes *java* utilizadas pelo *hibernate* e as tabelas presentes na base de dados. É a partir deste mapeamento que se abstrai o acesso à base de dados, sendo a estrutura relacional desta transformada numa hierarquia de objectos. Esta foi uma situação nova já que para o módulo de planos de pagamento esse mapeamento já estava realizado.

A implementação de todas as funcionalidades presentes neste módulo foi uma tarefa bastante interessante. O grande número e diversidade de ecrãs a implementar, que obrigaram, por exemplo, a utilizar a tecnologia *Ajax* (tecnologia na qual o estagiário não tinha prática), levaram a que o estagiário tivesse de implementar com maior rapidez mas mantendo a qualidade. Essa foi, aliás, uma das evoluções mais notórias que o estagiário sentiu, isto é, a capacidade de fazer mais em menos tempo.

Assim, a implementação das funcionalidades ocorreu sem problemas de maior, sendo o único aspecto menos positivo de relevo uma má interpretação do estagiário do significado de um campo na base de dados. O campo em questão representa o total compensado de uma prestação (incluindo juros), tendo o estagiário assumido que apenas se tratava do valor compensado sem juros. Este facto obrigou a alterações nas funções de cálculo de montantes associados a compensações.

Na figura seguinte são apresentadas o número de linhas de código produzidas pelo estagiário neste módulo. Podemos verificar que o número de linhas produzidas neste módulo é substancialmente superior ao número de linhas produzidas na implementação do módulo de planos de pagamento.

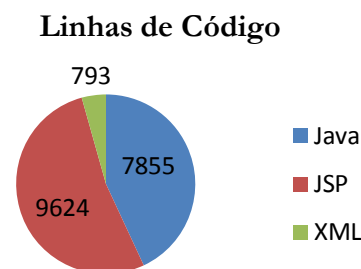


Figura 21: Número de linhas produzidas na implementação do módulo de gestão de conta corrente

3.3.3 Lançamento

Tal como vimos para o módulo de planos de pagamento, nesta fase a implementação realizada pelo estagiário foi testada por outro elemento do NONIO.

Os erros encontrados, onde se incluíram problemas nas ordenações de algumas listagens e problemas nas validações de alguns campos, foram facilmente corrigidos pelo estagiário.

Após o término da implementação do módulo de gestão de conta corrente foi pedido ao estagiário que preparasse uma formação sobre as funcionalidades de gestão presentes no módulo.

Na formação estiveram presentes os membros de gestão académica e financeira que irão ser responsáveis pela gestão de propinas dos alunos. No total estiveram presentes 10 pessoas. Esta, decorreu sem qualquer problema, sendo a audiência bastante participativa, procurando sempre perceber quais as ideias base que deram origem à solução implementada, não se limitando a apenas tentar sistematizar o modo de funcionamento do módulo.

O processo de gestão de propinas é algo bastante complexo pelo que as experiências das pessoas envolvidas neste processo são algo diferentes. Cada pessoa já passou por situações diferentes da pessoa ao lado, o que motivou algumas discussões (saudáveis) bastante interessantes onde se identificaram alguns aspectos que poderiam ser melhorados. Por exemplo, no processo de adição ou edição de uma prestação haver a possibilidade de associar um documento à prestação. Este pedido é motivado pelo facto de existirem situações onde a operação de adicionar ou editar uma prestação é justificada por um determinado despacho. Neste momento, associado à operação de adicionar ou editar uma prestação, o utilizador poderá escrever observações mas, no caso de existir um despacho não seria muito prático ter que copiar o seu conteúdo para as observações pelo que esta será uma alteração que certamente se irá disponibilizar numa próxima fase.

No final foram vários os elogios já que as pessoas presentes se aperceberam do salto qualitativo que a aplicação irá trazer a todo o processo de gestão de propinas.

No momento da escrita deste relatório já foram criados 92 planos de pagamento baseados nos 26 modelos que foram criados no módulo de planos de pagamento. Dos 92, 10 deles já foram alterados através do módulo de gestão de conta corrente. As alterações passaram pela anulação de prestações e pela criação de novas prestações com valores diferentes. A baixa utilização deste módulo é explicada pelo facto de ainda ter apenas decorrido a 1ª fase de inscrições em cursos na Universidade de Coimbra. No momento em que forem realizadas as candidaturas e respectivas inscrições de alunos provenientes do secundário, espera-se que a utilização deste módulo aumente exponencialmente.

As figuras seguintes exemplificam alguns dos ecrãs implementados pelo estagiário:

Plano Pagamento

Itens a Pagamento Conta Corrente Compensações Recebimentos Reposições

Detalhes Matrícula Gestão de Consistência Financeira

Aluno: Valdenilde dos Santos Araujo (2012100486)

Ano Matrícula: 2012/2013 **Estado Matrícula:** Ativo

Curso: Mestrado em Literatura da Língua Portuguesa: Investigação e Ensino

Estado Financeiro da Matrícula: **Financeiramente Inconsistente** (Ocultar Informação)
 (03-07-2012) Foi adicionada uma prestação ao plano de pagamento do aluno.
 (03-07-2012) Foram anuladas prestações do plano de pagamento do aluno.

Ano Lectivo: 2012/2013

Detalhes Inscrição Frequência - 2012/2013

Modo Inscrição: Integral **Estado:** Lacrado (18-05-2012)

Data de Inscrição: 18-05-2012 **Data de Anulação:**

[Regularizar Dívidas](#)
[Adicionar Prestação](#)
[Anular Prestações em Bloco](#)
[Histórico Prestações Anuladas](#)

Plano Pagamento - 2012/2013

Descrição	Total		Total por Pagar		Data Limite	Pago
	Montante	Juros	Montante	Juros		
Taxa de Inscrição	20.00 €	0 €	0 €	0 €	18-05-2012	✓
1ª Prestação	259.30 €	0 €	0 €	0 €	31-10-2012	✓
2ª Prestação	259.30 €	0 €	259.30 €	0 €	31-01-2013	✗
3ª Prestação	259.30 €	0 €	59.30 €	0 €	31-03-2013	✗
4ª Prestação	150.00 €	0 €	99.00 €	0 €	31-05-2013	✗

Entidade Pagadora	Montante	Juros	Total por Pagar	Pago
Aluno	50.00 €	0 €	50.00 €	✗
FCT-Fundação p/a Ciência e Tecnologia	50.00 €	0 €	49.00 €	✗
Núcleo estudantes Medicina - AAC	50.00 €	0 €	0 €	✓
Total:	150.00 €	0 €	99.00 €	

Plano Pagamento Modelo - 2012/2013

Descrição	Montante	Data Limite
Taxa de Inscrição	20.00 €	-
1ª Prestação	259.30 €	31-10-2012
2ª Prestação	259.30 €	31-01-2013
3ª Prestação	259.30 €	31-03-2013
4ª Prestação	259.30 €	31-05-2013
Total:	1057.20 €	

Situações Especiais (1)

MLLPiE 2012/2013 [Matrícula 2012/2013]	Bolseiros da Universidade (Anual) Pedido (03-05-2012) Aceite (20-06-2012)
---	---

[Voltar](#)

Figura 22: Ecrã do módulo de gestão de conta corrente já em produção

Início Pesquisa Alunos Recebimentos

Valdenilde dos Santos Araujo (2012100486)
Mudar Aluno

InforGestão

Plano Pagamento Itens a Pagamento Conta Corrente Compensações Recebimentos Reposições

Pesquisa [Transferir Recebimentos](#)

Registado em: Hoje
 Última Semana
 Último Mês
 Outro Período:
 Data Início: dd-mm-aaaa Data Fim: dd-mm-aaaa

Incluir Recebimentos Anulados: Sim
 Não

Meio de Pagamento: Número:

Referência MB:

Entidade Pagadora:

Recebimentos

Meio Pagamento	Data Registo	Data Efeito	Montante Recebido	Crédito
Cheque n.º: 123456 BANIF-BI 21-12-2011	03-07-2012	03-07-2012	400.00 €	140.70 €
Terminal de Pagamento Automático Núcleo estudantes Medicina - AAC	03-07-2012	03-07-2012	50.00 €	-
Terminal de Pagamento Automático	03-07-2012	03-07-2012	200.00 €	-
Terminal de Pagamento Automático FCT-Fundação p/a Ciência e Tecnologia	03-07-2012	03-07-2012	1.00 €	-
Paypal	18-05-2012	18-05-2012	259.30 €	-
Paypal	18-05-2012	18-05-2012	20.00 €	-
Paypal	08-03-2012	08-03-2012	50.00 €	-

Figura 23: Ecrã do módulo de gestão de conta corrente já em produção

3.4. Módulo de Caixa

O módulo de Caixa será o módulo utilizado pelos elementos da secção de tesouraria dos diversos pontos de atendimento da Universidade de Coimbra.

O trabalho do estagiário neste módulo esteve apenas centrado na especificação, que ocorreu de forma semelhante ao descrito para os módulos anteriores. Das diversas reuniões de especificação de requisitos foi compilada a seguinte lista:

#	Requisito	Descrição
1	Consultar todas as prestações a pagamento.	Deverá ser possível consultar todas as prestações por pagar de uma determinada pessoa.
2	Compensar prestações.	Deverá ser possível compensar prestações por pagar utilizando um ou mais recebimentos.
3	Pesquisar compensações.	Deverá ser possível pesquisar operações de compensações por período temporal, por modo de pagamento, por posto de recebimento e por utilizador responsável pelo registo da operação de compensação.

4	Consultar detalhes de uma compensação.	Deverá ser possível consultar os detalhes de uma compensação. Na informação a apresentar deverão incluir as prestações compensadas bem como os recebimentos utilizados na operação de compensação.
5	Emitir Recibos.	A aplicação deverá possibilitar a emissão de recibos originais e segundas vias.
6	Anular Operação de Compensação.	O utilizador poderá anular uma operação de compensação. No entanto, os recebimentos também serão anulados não sendo por isso creditado qualquer valor ao aluno.

Tabela 6: *Lista de requisitos do módulo de caixa*

Será no módulo Caixa que serão processados todos os recebimentos de alunos que sejam efectuados presencialmente.

Neste módulo irá ser possível compensar prestações e/ou outros emolumentos (certidões, diplomas etc.). A compensação poderá ser realizada para apenas um item ou para um conjunto itens. O módulo caixa suportará os meios de pagamento numérico, multibanco, cheque e transferência bancária e poderão ser utilizados um ou mais meios de pagamento para compensar um ou mais itens.

Com a entrada em funcionamento do projecto NONIO-GP passará a ser dada a possibilidade ao aluno de pagar apenas parte de uma determinada prestação, ou uma parte dos juros devidos. Essa possibilidade não será dada para a compensação de outros emolumentos já que estas são consideradas dívidas não fraccionáveis. Assim, o aluno mesmo que não tenha dinheiro suficiente para pagar a totalidade de uma prestação poderá pagar apenas parte e aplicação tratará de fraccionar a dívida e actualizar o plano de pagamento do aluno.

No módulo caixa deverá ser também possível ver todo o histórico de compensações realizadas e anuladas. Os utilizadores da aplicação terão também a possibilidade de emitir a 2ª via de um recibo, anular uma compensação e alterar o meio de pagamento de uma compensação que já tenha registada. As duas últimas operações apenas poderão ser realizadas enquanto os dados das operações não tenham sido integrados em SAP-FI.

Para além das reuniões de especificação de requisitos com os elementos das áreas académica e financeiras, foi incumbida ao estagiário a tarefa de realizar uma análise de requisitos contextual.

A análise de requisitos contextual consiste na recolha de requisitos directamente da fonte, ou seja, directamente do local onde a aplicação irá ser utilizada e tendo em conta todo o contexto envolvente. Esta análise foi bastante importante pois apesar de a área académica estar representada nas reuniões de especificação de requisitos, a função normalmente desempenhada pelos elementos representantes não engloba a parte de tesouraria.

Através deste método pretendeu-se descobrir potenciais requisitos que apenas quem trabalha regularmente na tesouraria poderia identificar.

Assim, o estagiário deslocou-se à tesouraria do posto de atendimento do Palácio dos Grilos e acompanhou o trabalho realizado pelos elementos responsáveis pela tesouraria. Os

objectivos passavam não só pela recolha de requisitos mas também por analisar todas as situações com que os elementos da tesouraria eram confrontados ou já tinham sido confrontados. Ou seja, interessava identificar as situações limite que já tinham acontecido para que estas pudessem ser contempladas no módulo de caixa.

Como é fácil perceber alguns dos requisitos recolhidos neste processo já tinham sido definidos nas reuniões de análise pelo que a lista seguinte representa aqueles que ainda não tinham sido contemplados e que se podem encontrar no protótipo:

- **Mesmo recibo poder contemplar diversas formas de pagamento** - Utilizando a aplicação *XPaga*, se um aluno decidir pagar um determinado item com dois meios de pagamento diferentes terão de ser passados dois recibos diferentes. A partir do NONIO-GP isso não irá acontecer, tendo o modelo dos recibos sido alterado para acomodar esta situação.
- **Poder alterar o modo de pagamento de um recibo que já foi emitido** – É frequente existirem erros na definição do meio de pagamento com que o aluno pagou determinado item. Assim, segundo os elementos da tesouraria este é um requisito bastante importante na medida que poderá evitar erros nos mapas pagamentos que se têm de gerar no final de cada dia.
- **Imprimir 2ªs vias de recibos** – Segundo os elementos da tesouraria este é um pedido bastante frequente por parte dos alunos e que na aplicação corrente não pode ser executado.

Esta foi sem dúvida uma experiência bastante enriquecedora pois permitiu ao estagiário um contacto directo com o utilizador final e uma análise das exigências a que os estes irão ser sujeitos aquando da utilização da aplicação. O resultado desta tarefa foi bastante satisfatório pois permitiu identificar requisitos importantes que poderiam não ter sido contemplados caso o estagiário não tivesse realizado esta tarefa.

A prototipagem e a escrita do documento de especificações adicionais do módulo de Caixa esteve ao cargo de outro elemento do NONIO. No entanto, o estagiário juntamente com o autor do protótipo e o gestor de projecto realizaram reuniões regulares onde, tal como para os outros módulos, era discutida e analisada a solução proposta e eram dadas sugestões de melhoramento.

Em termos do documento de especificações adicionais o estagiário esteve encarregue de o rever para detectar erros que foram posteriormente corrigidos pelo autor. O estagiário esteve também envolvido em todo o processo de análise e aprovação formal do documento de especificações adicionais.

3.5. Integração NONIO/SAP-FI

Um dos aspectos fundamentais do projecto NONIO-GP passa pela integração em SAP-FI da informação registada em NONIO.

A integração entre NONIO e SAP-FI foi um processo colaborativo entre elementos do NONIO, elementos da área de gestão financeira e também elementos da empresa externa responsável pelos desenvolvimentos em SAP-FI. Este documento irá centrar-se maioritariamente nos aspectos da implementação na aplicação NONIO.

Nas secções anteriores foram descritos os mecanismos de registo e gestão de propinas e emolumentos. Em NONIO são registados dados relativos a dívidas, recebimentos, reposições, crédito e anulações. Cada uma destas acções irá gerar o que se convencionou chamar um movimento de integração em SAP-FI. Existem dois tipos de movimentos de integração: dívidas e recebimentos.

No entanto, para SAP-FI não são enviados todos os movimentos registados em NONIO. Previamente ao envio é realizado um processo de agregação em pacotes dos movimentos referidos acima, segundo critérios específicos para dívidas e recebimentos.

Um aspecto importante a salientar é que nos dados enviados a SAP-FI não haverá discriminação da informação por aluno. Todos os movimentos de dívidas ou recebimentos cuja responsabilidade seja do próprio aluno estarão associados a uma figura genérica “ALUNO”. Apenas haverá discriminação para movimentos cuja responsabilidade esteja associada a uma entidade terceira pagadora (Empresas, fundações, serviços sociais, etc.). Este aspecto significa que em SAP-FI não haverá a possibilidade de consultar as dívidas ou recebimentos de um determinado aluno, sendo esse género de consulta apenas possível em NONIO.

A discriminação de movimentos por entidade terceira pagadora é uma necessidade já que é apenas em SAP-FI que são emitidas facturas a clientes. Este aspecto levou a que se identificasse a necessidade de existir não apenas uma funcionalidade de envio automático de pacotes mas também a possibilidade de enviar pacotes manualmente.

A integração automática de pacotes será realizada através de um *job* diário agendado para correr de madrugada. No entanto, são frequentes as situações onde existe a necessidade de emitir facturas relativas a dívidas criadas no momento. Isso significa que esperar pela integração automática de pacotes não é uma solução, tendo de ser o utilizador a integrar os pacotes manualmente, definindo-os como prioritários.

Como foi referido existem dois tipos de movimentos de integração em SAP-FI que irão dar origem aos pacotes a enviar. Nos próximos parágrafos irão ser descritos os dois tipos de movimentos e os pacotes gerados a partir destes.

Dívidas

Os movimentos de dívidas irão ser originados pela criação ou anulação de prestações. Apenas são consideradas prestações do tipo Taxa de Inscrição ou Propinas já que só para estas é criada a dívida em SAP-FI antes da compensação da prestação. Para os restantes tipos de prestação a dívida sem SAP-FI apenas é criada quando existe a indicação que a prestação foi compensada. Assim, não é necessário enviar qualquer tipo de informação sobre o registo ou anulação deste tipo de prestações.

Este tipo de movimento terá um determinado montante e estará associado a um ano lectivo, curso, unidade orgânica e entidade pagadora que representa a informação da prestação que acabou de ser criada ou anulada. A distinção entre anulação e criação é feita através do sinal do montante do movimento.

A agregação dos movimentos em pacotes é realizada tendo em conta as características dos movimentos. Assim, movimentos com características semelhantes serão integrados no mesmo pacote. Mais precisamente, movimentos com um montante de sinal igual, da mesma entidade pagadora, com o mesmo tipo de prestação e da mesma unidade orgânica, curso e ano lectivo serão parte constituinte do mesmo pacote.

Através deste meio de integração, em SAP-FI existirão totais de dívidas por tipo prestação, unidade orgânica, curso, ano lectivo e entidade pagadora. Isto significa que existe uma visão global sobre tudo o que é devido à Universidade de Coimbra.

De modo a exemplificar, será simulado todo o processo de registo e agregação de movimentos bem como os pacotes criados do seguinte cenário:

“Criação de três prestações de 250 € para três alunos diferentes, sendo dois do curso de engenharia informática e um do curso de engenharia civil. Posteriormente, a prestação criada para um dos alunos de engenharia informática é anulada.”

Movimentos Criados

Tipo Prestação	Ano Lectivo	Curso	Unidade Orgânica	Entidade Pagadora	Montante
Propinas	2012/2013	LEI	DEI	ALUNO	250 €
Propinas	2012/2013	LEI	DEI	ALUNO	250 €
Propinas	2012/2013	LEC	DEC	ALUNO	250 €
Propinas	2012/2013	LEI	DEI	ALUNO	-250 €

Tabela 7: *Movimentos de dívidas*

A criação de prestações irá originar três movimentos com um sinal positivo. A anulação irá originar um movimento com sinal negativo. Todos os movimentos estarão associados à entidade pagadora genérica “ALUNO” já que a responsabilidade da prestação está associado ao próprio aluno e não a uma entidade terceira pagadora.

Pacotes Gerados

Tipo Prestação	Ano Lectivo	Curso	Unidade Orgânica	Entidade Pagadora	Montante
Propinas	2012/2013	LEI	DEI	ALUNO	500 €
Propinas	2012/2013	LEC	DEC	ALUNO	250 €
Propinas	2012/2013	LEI	DEI	ALUNO	-250 €

Tabela 8: *Pacotes de dívidas*

Na tabela representativa dos movimentos criados, os dois primeiros seguem todos os critérios de agregação pelo que serão parte integrante do mesmo pacote. O pacote originado, de cor azul, terá um valor total de 500€ correspondente à soma dos dois movimentos representados na mesma cor. O movimento representado a vermelho irá dar origem a um pacote, representado pela mesma cor, já que não existem outros movimentos que sigam todos os critérios de agregação. O mesmo raciocínio é aplicado ao movimento de cor verde, pelo que irá originar o pacote representado pela mesma cor.

Recebimentos

Os movimentos de recebimentos irão ser originados pelo registo e anulação de recebimentos, compensações e reposições. Nestes movimentos também estará reflectida a gestão de todo o crédito existente.

Como vimos anteriormente, é com base em recebimentos que irão ser realizadas compensações de prestações. O registo ou anulação de compensações irá dar origem a um movimento de integração em SAP-FI. Este tipo de movimentos terão um montante, tipo de prestação, ano lectivo, curso, unidade orgânica e entidade pagadora tal e qual como um movimento do tipo de dívidas. Adicionalmente, estes movimentos estarão associados a um meio de pagamento do recebimento utilizado na compensação, ao posto de recebimento (se aplicável) onde o recebimento foi registado e a uma data de registo do recebimento. Tal como nos movimentos de dívidas, a distinção entre anulação e registo é feita através do sinal do montante.

Como foi referido, é também através destes movimentos que é realizada toda a gestão do crédito existente em NONIO. Assim, cada vez que exista uma operação que origine crédito (ex. Anulação de uma compensação) é criado um movimento adicional para reflectir esse facto. Em SAP-FI, o crédito será totalmente indiscriminado, sendo apenas um total, pelo que o movimento a enviar será apenas constituído por um montante e a indicação que se trata de crédito. Como seria de esperar, o contrário também acontece já que em operações que utilizem crédito (ex. Regularizações) será criado um movimento que reflecta o montante utilizado. A distinção entre registo e utilização de crédito é realizada através do sinal do montante.

As anulações de recebimentos poderão originar um conjunto de movimentos descritos acima. Ao anular um recebimento, todas as compensações que estejam associadas a este terão de ser anuladas. Adicionalmente, e caso operações utilizando o recebimento a anular tenham originado crédito, terá de ser criado um movimento que reflecta a anulação desse crédito em SAP-FI.

Em termos de reposições, como vimos anteriormente, estas apenas poderão ser realizadas para recebimentos com crédito. Isto significa que os movimentos gerados também irão incidir sobre o valor total de crédito registado em SAP-FI. Uma reposição irá dar origem a um movimento constituído por um montante e a indicação que se trata de uma reposição. Em termos do crédito registado em SAP-FI, a reposição irá reflectir-se numa redução desse crédito. Por outro lado, a anulação de uma reposição irá reflectir-se num aumento do crédito registado em SAP-FI.

No que toca à agregação dos movimentos descritos em pacotes existirão critérios distintos. Por um lado, todos os movimentos assinalados como crédito serão agregados pelo sinal do movimento (que distingue entre o registo ou anulação de crédito). O mesmo acontece para todos os movimentos assinalados como reposições. Por outro lado, os movimentos correspondentes a compensações serão agregados pelo sinal do montante, tipo de prestação, ano lectivo, curso, entidade pagadora, tipo de meio de pagamento, posto de

recebimento e data de registo. Isto significa que movimentos que partilhem as características listadas acima serão parte constituinte do mesmo pacote.

De modo a exemplificar, será simulado todo o processo de registo e agregação de movimentos bem como os pacotes criados do seguinte cenário:

“Compensação de uma prestação de propinas de 250 € de um aluno do curso de Engenharia Informática. Foi utilizado um recebimento no valor de 250 €, pago em cheque no posto de recebimento do Pólo II em 23-07-2012.

No mesmo dia e posto de recebimento é compensada outra prestação de propinas de 250 € de um aluno de engenharia informática utilizando um recebimento no valor de 500 €, pago em cheque. Posteriormente, o segundo recebimento é anulado.”

Movimentos Criados

Tipo Prestação	Ano Lectivo	Curso	U.O	Entidade Pagadora	Posto Receb.	Meio Pag.	Data Registo	Crédito	Reposição	Mont.
Propinas	2012/2013	LEI	DEI	ALUNO	Polo II	Cheque	23-07-12	Não	Não	250 €
Propinas	2012/2013	LEI	DEI	ALUNO	Polo II	Cheque	23-07-12	Não	Não	250 €
-	-	-	-	-	-	-	-	Sim	-	250 €
Propinas	2012/2013	LEI	DEI	ALUNO	Polo II	Cheque	23-07-12	Não	Não	-250 €
-	-	-	-	-	-	-	-	Sim	-	-250 €

Tabela 9: *Movimentos de recebimentos*

As duas compensações de prestações irão originar os dois primeiros movimentos. No caso da segunda compensação, o valor do recebimento é superior ao valor total da prestação pelo que existe a necessidade de comunicar a SAP-FI a existência de crédito. Esta situação é reflectida no terceiro movimento.

Posteriormente, e devido à anulação do segundo recebimento, são criados dois movimentos distintos. O primeiro tem como objectivo anular a compensação realizada. O segundo, anular o crédito passado a SAP-FI.

Pacotes Gerados

Tipo Prestação	Ano Lectivo	Curso	U.O	Entidade Pagadora	Posto Receb.	Meio Pag.	Data Registo	Crédito	Reposição	Mont.
Propinas	2012/2013	LEI	DEI	ALUNO	Polo II	Cheque	23-07-12	Não	Não	500 €
-	-	-	-	-	-	-	-	Sim	-	250 €
Propinas	2012/2013	LEI	DEI	ALUNO	Polo II	Cheque	23-07-12	Não	Não	-250 €
-	-	-	-	-	-	-	-	Sim	-	-250 €

Tabela 10: *Pacotes de recebimentos*

Na tabela representativa dos movimentos criados, os dois primeiros seguem todos os critérios de agregação pelo que serão parte integrante do mesmo pacote. O pacote originado, de cor azul, terá um valor total de 500€ correspondente à soma dos dois movimentos representados na mesma cor. O movimento representado a vermelho irá dar origem a um pacote, representado pela mesma cor, já que não existem outros movimentos que sigam todos os critérios de agregação. O mesmo raciocínio é aplicado aos movimentos de cor verde e roxa pelo que irão originar pacotes representados por cores semelhantes.

3.5.1 Arquitectura de Integração

Na secção anterior vimos, em detalhe, os critérios de geração de pacotes de dívidas e recebimentos. Nesta secção será detalhada toda a arquitectura de integração dos referidos pacotes.

Como foi referido acima o processo de integração pode ser manual ou automático. Na figura seguinte é representado um diagrama sequencial do processo de integração de pacotes em SAP-FI.

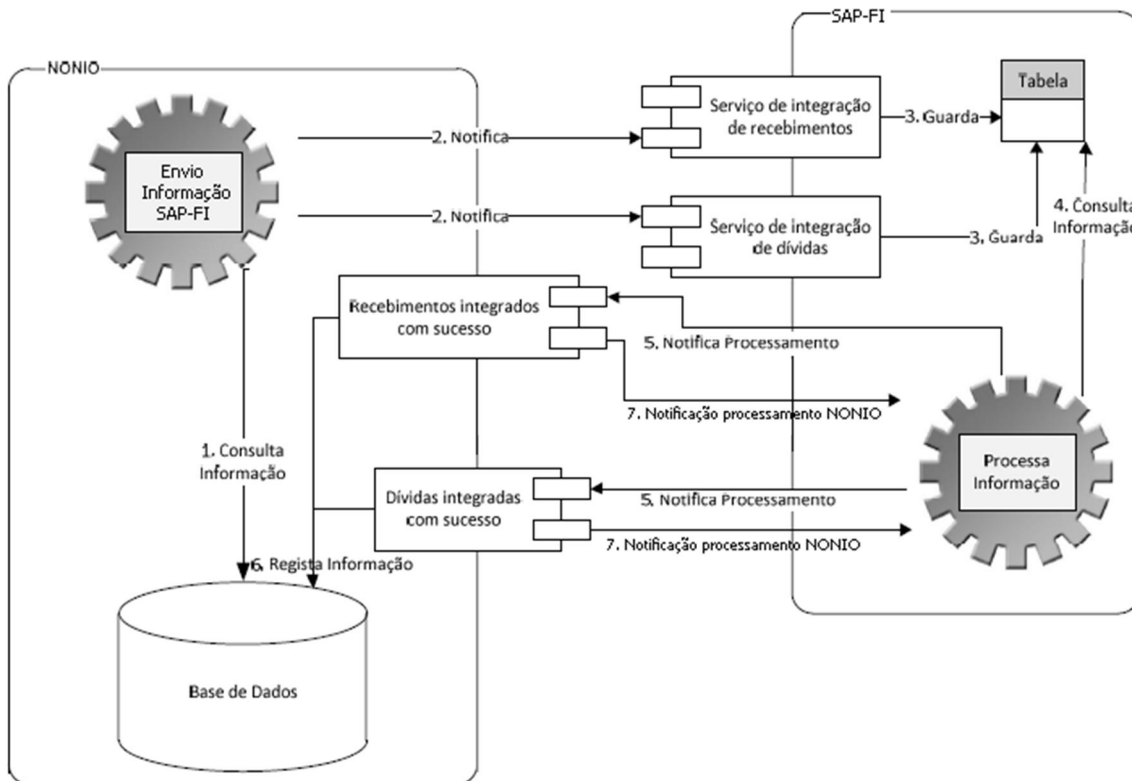


Figura 24: Processo de integração de pacotes

Como podemos verificar numa primeira fase será realizada uma consulta à base de dados para a realização do processo de agregação de todos os movimentos de dívidas e recebimentos, sendo posteriormente criados os respectivos pacotes. Ultrapassada esta fase são invocados os *webservices SOAP* disponibilizados por SAP-FI que irão receber as listas de pacotes geradas.

Os pacotes de dívidas e recebimentos entram depois numa fila à espera de serem processados. Em SAP-FI existe um *job* responsável pelo processamento de pacotes definido para correr às 6:00 da madrugada. No entanto, como foi referido na secção anterior os

pacotes poderão ser marcados como prioritários (apenas no envio manual) o que fará com que quando um pacote deste género chegue a SAP-FI seja imediatamente processado não tendo de ficar na fila de espera nem de aguardar pela hora agendada para o processamento de pacotes.

Após o término do procedimento de processamento de pacotes, SAP-FI é responsável por invocar um *webservice SOAP* disponibilizado por NONIO. Este *webservice* tem como objectivo receber a confirmação de quais os pacotes que foram processados com sucesso. Assim, o *webservice* irá receber os seguintes parâmetros:

Parâmetros	Tipo	Descrição
listaIDsPacotes	Integer[]	Lista com os identificadores dos pacotes processados com sucesso em SAP-FI
TipoPacote	String	“DIVIDAS” ou “RECEBIMENTOS

Tabela 11: *Webservice de confirmação de processamento de pacotes*

Existirão duas invocações a este *webservice*. Uma para enviar os identificadores dos pacotes processados de dívidas e outra para os de recebimentos.

Quando a confirmação é recebida em NONIO, os pacotes são assinalados como integrados. Na resposta à invocação do *webservice* por parte de SAP-FI irá a confirmação de quais os pacotes que foram assinalados como integrados com sucesso em NONIO sob a forma de uma lista de identificadores. A resposta irá conter os seguintes parâmetros:

Parâmetros	Tipo	Descrição
ListaPacotesProcessados	Integer []	Lista de pacotes que foram processados com sucesso no NONIO.
Sucesso	boolean	Indica o sucesso/insucesso da operação.
Detalhes	String	Detalhes do motivo de insucesso da operação.

Tabela 12: *Retorno do webservice de confirmação de processamento de pacotes*

Quando a resposta é recebida em SAP-FI o pacote é marcado como integrado finalizando assim o processo de integração.

Tratamento de Falhas

Nos parágrafos anteriores vimos como é que o processo de integração decorre perante condições ideais, ou seja sem que haja problemas de comunicação entre NONIO e SAP-FI. Assim, é de todo o interesse perceber como é que possíveis falhas de comunicação irão ser tratadas.

A arquitectura pensada garante que nenhum pacote ficará por processar ou é processado mais do que uma vez. O tratamento de falhas assenta sobre duas premissas base. A primeira, do lado de NONIO é o facto de se enviarem sempre todos os pacotes que não

estejam assinalados como integrados e não apenas os novos pacotes gerados pelo processo de agregação. Deste modo garante-se que um pacote nunca ficará por enviar já que irá ser constantemente enviado até que seja confirmado o seu processamento. A segunda, do lado de SAP-FI, é que se irão descartar todos os pacotes cujo identificador já tenha sido recebido. Assim, garante-se que um pacote nunca irá ser processado mais do que uma vez.

Caso seja recebido um pacote com um identificador já existente isso significará um de três cenários:

- O pacote foi processado com sucesso mas existiu um erro na invocação do *webservice* de confirmação de processamento de pacotes.
- O pacote foi processado com sucesso mas a invocação do *webservice* de confirmação de processamento de pacotes ainda não foi realizada.
- O processamento do pacote ainda não foi realizado com sucesso.

Nestes três cenários é seguro descartar o pacote enviado já que a informação já está presente em SAP-FI. Nos dois primeiros cenários, irá ser realizada ou repetida a invocação ao *webservice* de confirmação de processamento de pacotes. No último cenário terá de se esperar que o processamento seja realizado com sucesso para depois se invocar o referido *webservice*.

A acrescentar ao descrito anteriormente também é garantido que SAP-FI irá enviar todos os identificadores de pacotes processados com sucesso para os quais ainda não tenha sido recebida a confirmação NONIO de que o pacote foi assinalado como integrado.

Deste modo, garante-se que situações onde a invocação ao *webservice* de confirmação de processamento de pacotes tenha sido realizada com sucesso (tendo o pacote sido marcado como integrado em NONIO, não sendo por isso voltado a enviar nos procedimentos de integração) mas onde a resposta contendo a confirmação de NONIO de que os pacotes associados aos identificadores enviados por SAP-FI tinham sido assinalados como integrados se tenha extraviado estão contempladas. Isto acontece, já que em NONIO, mesmo que seja recebida a confirmação de processamento de um pacote que já esteja assinalado como integrado, é enviada uma nova confirmação para SAP-FI.

A conjugação dos mecanismos descritos acima garante que o processamento de um pacote só é finalizado quando existir a confirmação desse facto tanto do lado de NONIO como de SAP-FI, garantindo assim que a arquitectura desenhada é tolerante a falhas.

3.5.2 Integração Automática de Pacotes

Em termos de modelo de dados foram criadas quatro novas entidades. Duas delas, INTEGRACAO_RECEBIMENTOS_SAP e INTEGRACAO_DIVIDAS_SAP, serão responsáveis por armazenar todos os movimentos de integração em SAP-FI gerados por operações em NONIO. Nas entidades PACOTES_INT_DIVIDAS_SAP e PACOTES_INT_RECEB_SAP serão armazenados todos os pacotes gerados pelo procedimento de agregação de movimentos. O modelo de dados está disponível no anexo X.

Definidos os requisitos de agregação de pacotes, arquitectura da integração e modelo de dados, o estagiário passou à construção da funcionalidade automática de integração de pacotes.

Na primeira fase o estagiário criou as funcionalidades de agregação dos movimentos de integração em SAP-FI em pacotes. Quando a tarefa foi concluída o estagiário implementou as funcionalidades de invocação dos *webservices* disponibilizados por SAP-FI. Estas funcionalidades foram implementadas num serviço semelhante ao descrito na secção “Construção” do módulo de Planos de Pagamento.

Foi também necessária a definição do *job* de geração e envio automático dos pacotes. Esta é realizada através da utilização da *Framework Spring* em conjunção com a *Framework Quartz* que oferece um conjunto de funcionalidades de agendamento de processos. Na imagem seguinte podemos visualizar a definição do *job*:

```
<bean name="geraEnviaPacotesSAPJob" class="org.springframework.scheduling.quartz.JobDetailBean">
  <property name="jobClass" value="edu.fctuc.noniows.services.financeira.jobs.GeraEnviaPacotesSAPJob" />
  <property name="jobDataAsMap">
    <map>
      <entry key="geracaoEnvioPacotesSAPService" value-ref="utilGeracaoEnvioPacotesSAPServiceTxManager" />
      <entry key="jobGeraEnviaPacotesSAPActivo" value="\${arranque.jobGeraEnviaPacotesSAPActivo}" />
    </map>
  </property>
</bean>
```

Figura 25: Definição do *job* de geração e envio de pacotes

O *job* é executado todos os dias às 3h da madrugada.

Finalizadas com sucesso as duas tarefas anteriores passou-se à implementação do *webservice* de confirmação de processamento de pacotes.

O primeiro passo foi a implementação da funcionalidade que irá receber uma lista de identificadores de pacotes processados e o seu tipo marcando-os como integrados em NONIO. Os identificadores dos pacotes marcados como integrados são depois enviados na resposta à invocação do *webservice* finalizando o processo de integração. Essas funcionalidades foram implementadas num serviço definido de forma idêntica ao referido para o módulo de planos de pagamento.

O segundo passo consistiu na definição do *webservice* em si. Na imagem seguinte é ilustrada o modo se disponibiliza o acesso a funcionalidades presentes no projecto NONIO:

```
<service name="GestaoFinanceiraSAP" scope="transportsession" targetNamespace="https://noniows.uc.pt/">
  <description>Gestao de pacotes SAP</description>
  <parameter name="ServiceObjectSupplier">org.apache.axis2.extensions.spring.receivers.SpringServletContextObjectSupplier</parameter>
  <parameter name="ServiceClass">edu.fctuc.noniows.services.financeira.GestaoFinanceiraSAP</parameter>
  <parameter name="SpringBeanName">wsGestaoFinanceiraSAPProxy</parameter>

  <messageReceivers>
    <messageReceiver mep="http://www.w3.org/2004/08/wsdl/in-only" class="org.apache.axis2.rpc.receivers.RPCInOnlyMessageReceiver" />
    <messageReceiver mep="http://www.w3.org/2004/08/wsdl/in-out" class="org.apache.axis2.rpc.receivers.RPCMessageReceiver" />
  </messageReceivers>

  <excludeOperations>
    <operation>setWsSecurity</operation>
    <operation>setGestaoPacotesProcessadosSAPService</operation>
    <operation>setNotificacaoMensagemDAO</operation>
    <operation>setNotificacaoPessoaDAO</operation>
    <operation>setTipoNotificacaoDAO</operation>
    <operation>geraNotificacaoErroProcessamentoPacotes</operation>
  </excludeOperations>
</service>
```

Figura 26: Definição do *webservice* de confirmação de processamento de pacotes

Terá de ser configurado qual o serviço que a chamada ao *webservice* irá invocar bem como quais os métodos desse serviço que não estarão acessíveis ao exterior.

A implementação das funcionalidades foi realizada sem problemas de maior. No entanto, houve a necessidade de alterar o modo como erros no processo de marcação de pacotes como integrados em NONIO eram enviados para SAP-FI.

Esta alteração deveu-se ao facto de, nos testes de integração, ter sido identificado um problema na recepção de *soap-faults* (mecanismo genérico para tratamento de erros) por parte de SAP-FI. Quando existia um erro em NONIO a marcar como integrado algum pacote, o *soap-fault* enviado originava a excepção “System Error” em SAP-FI. Estudado o problema chegou-se à conclusão que em SAP-FI não era suportado o retorno de *soap-faults* pelo que se teve de criar um objecto genérico para o efeito.

Na figura seguinte é apresentado o número de linhas de código produzido pelo estagiário na funcionalidade de integração automática de pacotes em SAP-FI.

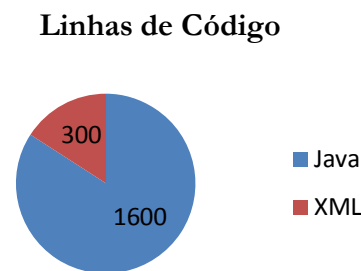


Figura 27: Número de linhas produzidas na implementação da funcionalidade de envio automático de pacotes

3.5.3 Integração Manual de Pacotes

Como foi referido chegou-se à conclusão que teria de existir uma forma de integrar pacotes de uma forma manual. Utilizando muito do trabalho realizado para a integração automática de pacotes, foram desenvolvidas funcionalidades para dar a possibilidade ao utilizador de seleccionar qual ou quais os pacotes que pretende integrar num determinado momento.

3.5.3.1 Especificação

Este processo decorreu de modo semelhante ao descrito para os restantes módulos abordados neste documento.

Das reuniões de especificação foi compilada a seguinte lista de requisitos:

#	Requisito	Descrição
1	Pesquisar por pacotes de dívidas e recebimentos gerados.	Deverá ser possível pesquisar por número de pacote, tipo, e período temporal.
2	Consultar detalhes do pacote.	Deverá ser possível consultar as características do pacote, incluindo os movimentos que foram agregados neste.

3	Simular processo de agregação e geração de pacotes de dívidas e recebimentos.	Deverá ser possível consultar todos os pacotes que seriam gerados caso a integração entre NONIO e SAP-FI fosse realizada naquele instante.
4	Integrar pacotes seleccionados pelo utilizador.	Os pacotes seleccionados deverão ser enviados para SAP-FI.
5	Marcar pacote como prioritário.	Deverá ser possível marcar um pacote como de processamento prioritário.
6	Marcar pacote como integrado ou não integrado.	Deverá ser possível marcar um pacote como integrado ou não integrado.

Tabela 13: *Lista de requisitos da funcionalidade manual de integração de pacotes*

Como podemos verificar pela tabela acima deverá ser possível pesquisar pelos pacotes que já foram gerados. Essa pesquisa deverá ser realizada por tipo de pacote (recebimentos ou dívidas), número de pacote, e período temporal.

O utilizador poderá também simular o processo de geração e agregação de pacotes de recebimentos e dívidas. A simulação apresenta a agregação de todos os movimentos de integração em SAP que ainda não estejam associados a um pacote. À lista de pacotes gerada pela simulação são adicionados todos os pacotes que ainda não tenham sido marcados como integrados em NONIO. Assim, o utilizador poderá não só enviar pacotes gerados pela simulação mas também todos aqueles que ainda não tenham sido marcados como integrados.

Os pacotes resultantes da simulação apenas serão persistidos na base de dados se forem seleccionados para serem enviados pelo utilizador. O utilizador terá ainda a possibilidade de marcar um pacote como sendo prioritário o que fará com que este seja automaticamente processado quando é recebido em SAP-FI não tendo de esperar pela hora agendada para o processamento de pacotes.

Prototipagem

A prototipagem da solução ocorreu de forma paralela às reuniões de especificação de requisitos funcionando como ferramenta de ajuda para a definição dos mesmos. Com base nos requisitos recolhidos, o estagiário foi responsável por apresentar uma proposta que atendesse a todos os aspectos identificados nas reuniões. O protótipo, que pode ser encontrado no anexo 8 (*academicos\operacoesRemotas*), conta com 3 ecrãs. Segue-se um exemplo da solução final:

Início : Integração Financeira

Operações Remotas | **Integração Financeira**

Gerir Pacotes Dívidas
Gerir Pacotes Recebimentos

Filtro de Pesquisa

Pacote N.º:

Tipo Pacote: Dívidas
 Recebimentos
 Todos

Relativo a: Hoje
 Última Semana
 Último Mês
 Outro Período:

Data Início: dd-mm-aaaa Data Fim: dd-mm-aaaa

Pacotes (20-01-2011 a 11-11-2011)

Pacote	Descrição	U.O	Curso	Meio Pagamento	Montante	Integrado SAP
123456 Recebimentos (20-12-2012)	Taxa de Candidatura Aluno 2012/2013	DEI	LEI 2º Cido	Dinheiro 20-12-2012 Polo I	1000.00 €	20-12-2012 ✓
654321 Recebimentos (21-12-2012)	Taxa de Candidatura Anulação Aluno 2012/2013	DEI	LEI 2º Cido	Dinheiro 20-12-2012 Polo I	-200.00 €	21-12-2012 ✓
87906 Dívidas (22-12-2012)	Prestação de Propinas Governo de Cabo Verde 2012/2013	DEI	LEI 2º Cido	PayPal 22-12-2012 Polo II	500.00 €	✗
258963 Recebimentos (21-12-2012)	Aluno Genérico	-	-	-	200.00 €	21-12-2012 ✓

Figura 28: Ecrã prototipado da funcionalidade de integração de pacotes manual

Modelo de Dados

O modelo de dados é partilhado entre as funcionalidades de integração automática e as funcionalidades de integração manual.

Documento de Especificações Adicionais

O estagiário foi também responsável pela escrita do documento de especificações adicionais da funcionalidade de integração manual de pacotes.

O documento foi depois analisado numa reunião de inspeção onde estiveram presentes elementos do NONIO.

3.5.3.2 Construção

A construção das funcionalidades de integração manual de pacotes decorreu sem qualquer problema. Nesta fase o estagiário já se encontrava à vontade com todas as especificidades da utilização de *struts* (vimo-las na descrição dos módulos descritos anteriormente) e tendo em conta que houve um aproveitamento de todo o trabalho realizado na implementação das funcionalidades de integração automática, todo o processo de desenvolvimento decorreu sem qualquer sobressalto.

Na figura seguinte são apresentadas o número de linhas de código produzidas pelo estagiário na funcionalidade integração manual de pacotes.

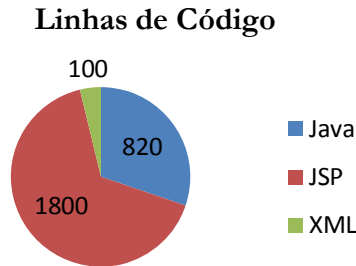


Figura 29: Número de linhas produzidas na implementação da funcionalidade de envio manual de pacotes

3.5.4 Lançamento

O processo de testes da integração entre NONIO e SAP-FI foi algo que esteve presente ao longo de todo o 2º semestre. Para além dos testes realizados por membros do NONIO ao processo de agregação e criação de pacotes, foram realizados diversos testes de integração com SAP-FI.

A colaboração entre o estagiário e a empresa externa responsável pelos desenvolvimentos em SAP-FI foi uma constante ao longo de todo este processo. Os primeiros testes de integração procuraram testar toda a comunicação entre NONIO e SAP-FI. Num primeiro momento, foi testada a invocação do *webservice* de envio de pacotes de dívidas e recebimentos disponibilizado por SAP-FI. Posteriormente, foi testada a invocação do *webservice* desenvolvido pelo estagiário para a comunicação do processamento de pacotes por parte de SAP-FI.

Como já foi referido, foi encontrado um problema na recepção de *soap-faults* por parte de SAP-FI o que motivou alterações à implementação para a resolução do problema.

No que toca aos testes para a validação do processamento de pacotes por parte de SAP-FI, estiveram envolvidos o estagiário, membros da área financeira e membros da empresa externa. O estagiário criava os pacotes de modo a que estes refletissem as situações a testar (dívidas, crédito, reposições, compensações etc.), sendo depois os membros da área financeira a validar se todos os passos realizados em SAP-FI eram os previstos e se o resultado final era o esperado.

Nas várias sessões de testes realizadas foram encontradas diversas situações em que o resultado final não era o esperado. Estas situações motivaram várias alterações de requisitos por parte da área de gestão financeira o que teve algumas consequências tanto na implementação realizada pelo estagiário como nos desenvolvimentos necessários em SAP-FI. Mais precisamente, para a resolução de alguns dos problemas encontrados teve de se alterar o conteúdo dos pacotes, adicionando informação mais detalhada sobre o curso e unidade orgânica. Teve de ser também alterado o processo de agregação dos pacotes de recebimentos já que foi necessário modificar alguns dos critérios pelos quais os movimentos eram agregados.

Todas as alterações levaram a que os desenvolvimentos em SAP-FI se atrasassem pelo que foi decidido dividir a entrada em produção da integração em duas fases distintas.

A primeira, a decorrer no momento da escrita do relatório, consiste na integração diária de apenas pacotes de recebimentos para o tipo de prestação taxa de candidatura. As taxas de candidatura foram o primeiro tipo de emolumento a ser criado e cobrado através de

NONIO pelo que se decidiu entrar em produção apenas a integrar pacotes deste tipo de prestação.

Previamente à entrada em produção desta fase foram realizados todos os testes necessários para que se garantisse que os pacotes do tipo de prestação taxa de candidatura seriam processados correctamente por SAP-FI. Neste momento, já foram enviados e processados com sucesso 776 pacotes.

Para a segunda fase, espera-se que as funcionalidades de processamento de pacotes em SAP-FI já estejam preparados para receber todos os tipos de pacotes gerados em NONIO. Espera-se que em finais do mês de Julho se possa avançar com a entrada em produção da segunda fase, estando todas as funcionalidades em NONIO já implementadas.

Nas imagens seguintes poderão ser visíveis dois ecrãs da funcionalidade de integração manual de pacotes:

Operações Remotas
Integração Financeira

[Gerir Pacotes Dívidas](#)
[Gerir Pacotes Recebimentos](#)

Filtro de Pesquisa

Pacote Nº:

Tipo Pacote: Dívidas
 Recebimentos
 Todos

Relativo a: Hoje
 Última Semana
 Último Mês
 Outro Período:

Data Início: dd-mm-aaaa Data Fim: dd-mm-aaaa

Pacotes

Pacote	Descrição	U.D.	Curso	Meio Pagamento	Montante	Integrado SAP
687 Recebimentos (03-07-2012)	Taxa de Candidatura Aluno 2012/2013	DEI	MEI 2º Cido Continuidade	Multibanco 02-07-2012	50.00 €	✘
686 Recebimentos (03-07-2012)	Taxa de Candidatura Aluno 2012/2013	FDUC	MD 2º Cido Especialização Avançada	Paypal 03-07-2012	50.00 €	✔ 03-07-2012
685 Recebimentos (03-07-2012)	Taxa de Candidatura Aluno 2012/2013	FMUC	MSO 2º Cido Especialização Avançada	Multibanco 02-07-2012	50.00 €	✔ 03-07-2012
684 Recebimentos (03-07-2012)	Taxa de Candidatura Aluno 2012/2013	FDUC	MD 2º Cido Especialização Avançada	Paypal 02-07-2012	150.00 €	✔ 03-07-2012

Figura 30: Ecrã da funcionalidade de integração de pacotes manual já em produção

Operações Remotas | Integração Financeira

Envio para SAP bem sucedido.

[Enviar Pacote](#)
[Marcar Como Integrado](#)

Detalhes

Tipo Pacote: Recebimentos	Data Criação: 03-07-2012
Tipo: Taxa de Candidatura	Ano Lectivo: 2012/2013
Posto Recebimento: -	Montante: 150.00 €
Meio Pagamento: Paypal Código SAP: Y	Data Registo: 02-07-2012
Entidade Pagadora: Aluno	
Unidade Orgânica: Faculdade de Direito Id: 142 Sigla: FDUC	
Curso: Mestrado em Direito (Código 20102006) Id: 3266 Sigla: MD Ciclo: 2º Ciclo Categoria: Especialização Avançada	
Registo Imediato: Não	Anulação Recebimento: Não
Id Pacote: 684	Id Pacote Dívidas: -
Data 1º Envio: 03-07-2012	Data Último Envio: 03-07-2012
Data Integração:	

Movimentos Integrados

Ano Lectivo	Descrição	Meio Pagamento	U.O.	Curso	Posto	Montante
2012/2013	Taxa de Candidatura Aluno	Paypal 02-07-2012	FDUC	MD	-	50.00 €
2012/2013	Taxa de Candidatura Aluno	Paypal 02-07-2012	FDUC	MD	-	50.00 €
2012/2013	Taxa de Candidatura Aluno	Paypal 02-07-2012	FDUC	MD	-	50.00 €
Total:						150.00 €

[Voltar](#)

Figura 31: Ecrã da funcionalidade de integração de pacotes manual já em produção

3.6. Módulo Propinas e Emolumentos

Tendo em conta a alteração na planificação para o trabalho do segundo semestre, apenas houve disponibilidade para a criação de uma primeira versão do protótipo para este módulo. O protótipo pode ser encontrado no anexo 8 (*alunos/temp_PropinasEmolumentos*).

Inserido na vista *InforEstud@nte*, este módulo será uma versão reduzida do módulo de gestão de conta corrente. Obviamente, o aluno não terá acesso às funcionalidades de gestão podendo apenas consultar os seus dados financeiros.

A proposta irá ser apresentada no decorrer das próximas aos membros da gestão académica e financeira, não o tendo já sido feito por falta de disponibilidade dos membros das referidas áreas.

Na figura seguinte poderemos ver um exemplo do protótipo criado:

Plano de Pagamento
Itens a Pagamento
Histórico Compensações

Matrícula Financeiramente Inconsistente!
 Se a situação persistir por favor informe os serviços académicos para que esta seja analisada.

Detalhes Matrícula

Aluno:	José Manuel Silva (2006125104)		
Ano Matrícula:	2007/2008	Estado da Matrícula:	Activo
Curso:	Licenciatura em Engenharia Informática		
Estado Financeiro da Matrícula:	Financeiramente Inconsistente		

Ano Lectivo: 2011/2012

Detalhes Inscrição Frequência - 2011/2012

Modo Inscrição:	Integral	Estado:	Deslacrado
Data Inscrição:	21-12-2011	Data Anulação:	

Plano Pagamento - 2011/2012

Descrição	Total		Total por Pagar		Data Limite	Pago
	Montante	Juros	Montante	Juros		
Taxa de Inscrição Governo de Cabo Verde	5.00 €	0 €	0 €	0 €	31-10-2011	✓
1ª Prestação (Importada)	250.00 €	0 €	0 €	0 €	31-10-2011	✓
2ª Prestação (Importada)	250.00 €	0 €	125.00 €	20.00 €	31-01-2012	✗
3ª Prestação	250.00 €	0 €	250.00 €	0 €	31-03-2012	✗
Total:	755.00 €	0 €	375.00 €	20.00 €		

▼ **Plano Pagamento Modelo - 2011/2012**

[Voltar](#)

Figura 32: Ecrã prototipado do módulo de propinas e emolumentos

3.7. Módulo Alunos

O estagiário foi também responsável pelas alterações necessárias às funcionalidades de anulação de uma inscrição em frequência, anulação de uma matrícula e alteração do estado de uma matrícula. Estas alterações foram necessárias já que estas operações têm influência directa nos dados financeiros do aluno.

Esta foi a única ocasião, durante todo o estágio, em que o estagiário não criou uma solução de raiz tendo apenas de alterar a solução existente. Assim, houve a necessidade da escrita de um pedido de alteração (que pode ser encontrado no anexo 4) que inclui modificações ao protótipo e ao documento de especificações adicionais do módulo de alunos.

As alterações necessárias foram semelhantes nas diversas funcionalidades já que passaram por dar ao utilizador a possibilidade de anular prestações do plano de pagamento do aluno. Para a anulação de uma inscrição em frequência é dada a possibilidade ao utilizador de anular prestações do plano de pagamento do ano lectivo da inscrição em frequência que se pretende anular. Na anulação de uma matrícula ou alteração do estado de uma matrícula é dada a possibilidade de anular prestações do plano de pagamento do ano lectivo da última inscrição em frequência da matrícula.

No que toca ao modelo de dados não houve a necessidade de realizar alterações já que as modificações às funcionalidades já eram suportadas pelo modelo de dados existente.

Aprovado o pedido de alteração, o estagiário foi responsável por realizar as alterações necessárias ao documento de especificações adicionais e também ao protótipo, que pode ser encontrado no anexo 8 (*académicos/alunosCurricular*).

A imagem que se segue representa o protótipo do ecrã de anulação de uma matrícula já com a possibilidade de anulação de prestações:

: Início : Alunos : Dados Curriculares : Estado Matrícula : Definir Observações

Alunos
 Dados Pessoais
 Dados Curriculares
 Propinas e Emolumentos

Inscrições
 Matrícula
 Frequência
 Exame
 Estágios Pedagógicos
 Situações Especiais
 Extensões de Prazos

Atendimento
 Documentos
 Reclamações
 Pagamentos
 Infrestudante
 Requerimentos

Gestão
 Candidaturas
 Creditações
 Pautas
 Estágios e Teses
 Notificações
 Normas e Procedimentos
 Calendários

Administração
 Cursos
 Disciplinas
 Menores
 Transições Bolonha
 Transições Planos Estudo
 Auditoria
 Anos Lectivos
 Utilizadores
 Configurações
 Listagens e Estatísticas
 Qualidade
 Inquéritos
 Gestão de Espaços
 Lembretes

Financeira
 Planos Pagamento
 Gestão Conta Corrente
 Caixa
 Listagens e Estatísticas
 Outras Configurações

César Francisco Gonçalves Rodrigues - 970509006
 Comunicações e Multimédia
 <Mudar Curso | Mudar Aluno>

Resumo Inscrições Avaliações Creditações Disciplinas Estatuto e Regalias Médias e Conclusão Estado Matrícula

Definir Observações

Novo Estado de Matrícula: Anulada

Observações:

Nº do Despacho:

Data do Despacho:

Data de Operação: * dd-mm-aaaa

* Campos de preenchimento obrigatório

Plano Pagamento - 2011/2012

Descrição	Total		Total por Pagar		Data Limite	Pago	<input type="checkbox"/>
	Montante	Juros	Montante	Juros			
Taxa de Inscrição	5.00 €	0 €	0 €	0 €	31-10-2011	<input checked="" type="checkbox"/>	<input type="checkbox"/>
1ª Prestação	250.00 €	0 €	0 €	0 €	31-10-2011	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2ª Prestação	250.00 €	0 €	125.00 €	0 €	31-01-2012	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3ª Prestação	250.00 €	0 €	250.00 €	0 €	31-03-2012	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Cancelar Continuar

Figura 33: Ecrã prototipado do módulo de alunos

A fase de construção passou por um reaproveitamento do trabalho realizado no módulo de gestão de conta corrente. Neste módulo, a funcionalidade de anulação de prestações em bloco é bastante semelhante às alterações que se pretendiam realizar no módulo de alunos o que facilitou bastante a tarefa. No entanto, o facto de estar, pela primeira vez, a alterar código já existente levou a que o estagiário tivesse de analisar toda a implementação já existente para depois inserir as alterações necessárias sem que esse facto trouxesse nenhum problema de funcionamento ao código já existente.

Na figura seguinte são apresentadas o número de linhas de código produzidas pelo estagiário na funcionalidade integração manual de pacotes.

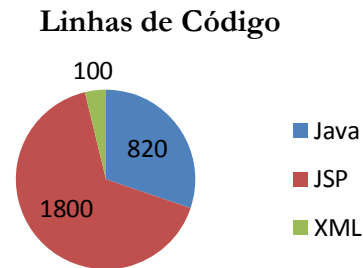


Figura 34: Número de linhas produzidas na implementação do módulo de alunos

Na fase de lançamento as alterações realizadas pelo estagiário foram testadas, não sendo encontrado qualquer problema. As alterações foram, assim, colocadas em produção.

A imagem seguinte representa o ecrã de anulação de uma matrícula já com as alterações efectuadas:

Início Alunos Dados Curriculares Estado Matrícula Definir Observações

Valdenilde dos Santos Araujo - 2012100486
Mestrado em Literatura da Língua Portuguesa: Investigação e Ensino
Mudar Curso | Mudar Aluno

InforGestão

Alunos

- Dados Pessoais
- Dados Curriculares**
- Propinas e emolumentos

Inscrições

- Matrícula
- Frequência
- Exames
- Extensões de Prazos

Atendimento

- Pedidos
- Documentos
- Requerimentos

Gestão

- Candidaturas
- Pautas
- Calendários
- Notificações
- Normas e Procedimentos

Administração

- Cursos
- Disciplinas
- Menores
- Transições
- Transições Planos Estudo
- Auditoria
- Anos Letivos
- Lembretes
- Utilizadores
- Operações Remotas
- Listagens e Estatísticas

Resumo Inscrições Avaliações Creditações Médias e Conclusão Estatuto e Regalias Estado Matrícula

Definir Observações

Novo Estado de Matrícula: Anulado

Observações:

Nº Despacho:

Data do Despacho: dd-mm-aaaa

Data de Operação: * dd-mm-aaaa

Plano Pagamento - 2012/2013

Descrição	Total		Total por Pagar		Data Limite	Pago	<input type="checkbox"/>
	Montante	Juros	Montante	Juros			
Taxa de Inscrição	20.00 €	0 €	0 €	0 €	18-05-2012	✓	<input checked="" type="checkbox"/>
1ª Prestação	259.30 €	0 €	0 €	0 €	31-10-2012	✓	<input type="checkbox"/>
2ª Prestação	259.30 €	0 €	259.30 €	0 €	31-01-2013	✗	<input checked="" type="checkbox"/>
3ª Prestação	259.30 €	0 €	259.30 €	0 €	31-03-2013	✗	<input checked="" type="checkbox"/>
4ª Prestação	259.30 €	0 €	259.30 €	0 €	31-05-2013	✗	<input type="checkbox"/>

Cancelar Continuar

Figura 35: Ecrã do módulo de alunos já em produção

3.8. Outras Actividades

O conteúdo desta secção detalha todas as actividades realizadas pelo estagiário que se encontram fora do âmbito dos três módulos descritos nas secções anteriores.

Estudo da Plataforma de Pagamentos AMA

Com vista à integração com o NONIO, foi pedido ao estagiário que estudasse a plataforma de pagamentos da Agência para a Modernização Administrativa (AMA)[8].

A plataforma de pagamentos da AMA é um sistema, criado pela PT-SI [9], que visa o suporte de diversos meios de pagamentos electrónicos nos quais se incluem o Visa, MasterCard e Multibanco.

Toda a comunicação entre a aplicação cliente e a plataforma da AMA é realizada através de *Web Services*. A aplicação cliente apenas terá de realizar o pedido, sendo o processo de tratamento desse pedido completamente transparente para a aplicação.

A plataforma oferece uma integração com a entidade de pagamentos SIBS [10], permitindo assim que se criem referências para as prestações. Um dos objectivos do projecto NONIO-GP é que possam ser criadas referências instantaneamente, sendo esse serviço suportado pela plataforma AMA.

Um senão da criação de referências de modo instantâneo é que uma referência depois de criada será sempre válida. Isto deve-se ao facto de nas referências criadas instantaneamente a validação ser realizada utilizando um método de *check-digit* que utiliza o montante e 7 dos 9 números de uma referência para calcular os dois números restantes verificando assim se esta é válida ou não. Esta é uma limitação que permitirá que uma determinada pessoa pague a mesma referência mais do que uma vez. Para evitar erros desta natureza haverá na aplicação uma funcionalidade que detecta estes casos e credita o valor pago ao aluno.

A plataforma oferece também uma integração com a entidade de pagamentos UNICRE [11], responsável pelos pagamentos com cartões de crédito.

Cada pagamento que seja realizado pelos métodos suportados será notificado à aplicação com que a plataforma está integrada. Neste processo existe a limitação de que a plataforma apenas tenta comunicar o referido pagamento uma vez independentemente da comunicação ser realizada com sucesso ou não. Como solução a este problema existe a possibilidade de realizar um pedido à plataforma que devolve todos os pagamentos realizados para um determinado intervalo de tempo.

Este serviço oferece ainda um módulo de *back office* onde será possível ver relatórios e gráficos de todos os pagamentos realizados.

O estudo desta plataforma teve como principal objectivo perceber quais as soluções oferecidas pela plataforma e perceber se seria exequível, em termos técnicos e temporais, criar uma solução própria para o projecto NONIO-GP. Alguma da resistência que existe em relação a esta plataforma reside no facto de o sistema NONIO ficar a depender de uma aplicação externa que, por exemplo, em caso de inactividade poderia comprometer o bom funcionamento da aplicação NONIO.

Foi chegada à conclusão que, de modo a que o projecto NONIO-GP possa estar pronto a tempo da gestão de propinas para o ano 2012/2013, teria de ser utilizada a

plataforma AMA, tendo sido tomadas as medidas para que a possível inactividade desta plataforma não cause qualquer problema à aplicação NONIO.

Arquitectura do NONIO-GP

Na secção de integração com SAP-FI foi descrito todo o procedimento de integração entre NONIO e SAP-FI. No entanto, esta é apenas uma das integrações que se realizam neste projecto. Adicionalmente, são realizadas integrações com a plataforma da AMA e também com o *PayPal*.

A figura seguinte procura esquematizar a interacção entre a aplicação NONIO e a plataforma da AMA e *PayPal*:

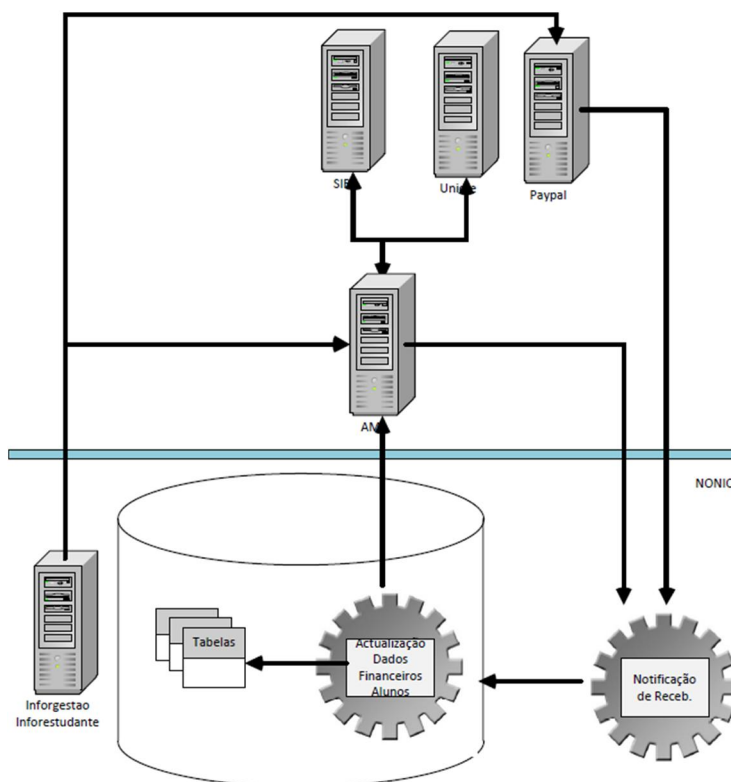


Figura 36: Arquitectura de Integração com plataforma AMA e PayPal

A aplicação comunica com a plataforma de pagamentos da AMA para o suporte de pagamento com cartões de crédito e criação de referências multibanco e directamente com o *Paypal* para o suporte deste meio de pagamento. A integração que a plataforma da AMA faz com as entidades de pagamentos SIBS (criação de referências) e UNICRE (suporte de pagamento através de cartão de crédito) é completamente transparente para a aplicação.

Na aplicação NONIO, a criação de uma prestação não implica a invocação à plataforma da AMA para a criação da referência multibanco. Deste modo, todo o processo de criação de prestações é completamente independente da plataforma AMA não ficando sujeito períodos de inactividade deste serviço.

A criação de referências multibanco será realizada por um *job* que corre periodicamente com a missão de requisitar referências multibanco para todas as prestações não anuladas existentes na base de dados. Directamente na aplicação, será requisitada uma referência multibanco cada de vez que um utilizador entre nos detalhes de pagamento de uma prestação que ainda não tenha associada uma referência multibanco.

Através da aplicação será também invocada a plataforma da AMA para o pagamento utilizando cartão de crédito.

Da base de dados serão realizadas invocações periódicas à plataforma de AMA de forma a obter-se uma lista de recebimentos para um determinado período temporal.

No que toca ao *Paypal*, serão invocados os seus serviços cada vez que um aluno pretenda pagar utilizando este meio de pagamento.

Tanto o *PayPal* como a plataforma AMA, serão responsáveis por invocar um *webservice* disponibilizado por NONIO para acusarem a recepção de um recebimento. No caso do *PayPal*, são enviadas várias tentativas de confirmação de um pagamento até que a mensagem seja recebido com sucesso. No caso da plataforma da AMA apenas será feita uma tentativa de confirmação de um pagamento. É devido a este facto que existe a necessidade de periodicamente pedir a lista de recebimentos existente para garantir que nenhum recebimento é perdido.

Como pudemos verificar existem várias operações que irão ser realizadas directamente a partir da base de dados. Isto acontece já que a arquitectura do NONIO-GP centraliza toda a lógica de negócio na base de dados, o que significa que existirá um módulo que suportará as operações de criação e anulação de prestações, criação e anulação de compensações, cálculo de juros, criação de referências, entre outras.

Esta decisão foi tomada com base na experiência dos elementos do NONIO, já que devido à complexidade do projecto aliada a uma qualidade dos dados importados que não é elevada, existirão correcções que terão de ser realizadas directamente na base de dados. Essas correcções nunca passarão apenas por editar, acrescentar ou remover os dados de uma tabela, envolvendo sempre alterações em todas as tabelas relacionadas com a original.

Na figura seguinte são identificadas as diversas formas de alteração de dados:

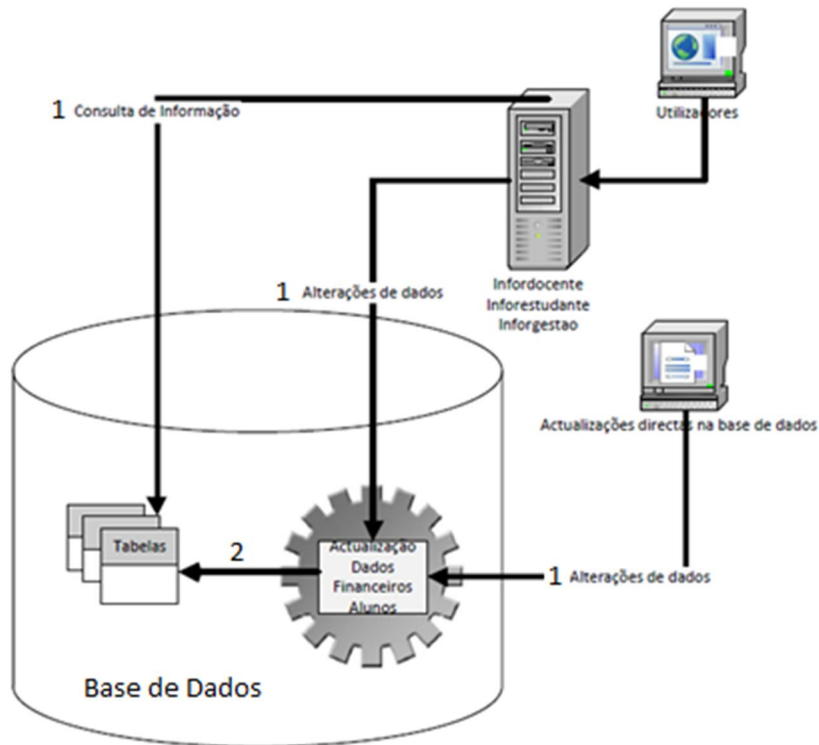


Figura 37: Processo de Actualização de Dados

Como podemos verificar, existirá um módulo na base de dados que será responsável pela actualização de dados na base de dados. Tanto as aplicações do sistema NONIO como as actualizações realizadas manualmente invocarão procedimentos do módulo referido.

No que toca à actualização de dados, é necessário garantir que apenas um e um só processo está a actualizar os dados de um aluno num determinado momento. Este é um requisito crítico que se não for cumprido há o risco de se realizarem operações sobre dados inconsistentes.

Para que se garanta a consistência dos dados irá ser implementado um mecanismo *optimistic locking* [12] com controlo de concorrência.

O controlo de concorrência irá utilizar uma particularidade do SGBD Oracle, o *System Change Number* (SCN) [13]. Este número, associado a todos os registos presentes na base de dados, permite saber a versão de um registo específico. Com essa informação é possível verificar se um determinado registo foi alterado entre o momento da consulta e o momento em que se irá realizar uma operação. Caso existam diferenças a operação não poderá continuar.

Assim, sempre que se pretende actualizar os dados é necessário cumprir duas condições: A primeira passa por obter o *lock* das tabelas onde irão ser realizadas as alterações. A segunda passa por efectuar as verificações descritas no parágrafo anterior. Apenas se estas duas condições se verificarem é que será possível actualizar os dados.

Reunião de Análise de Riscos

A análise de riscos é um aspecto que bastante importante no desenvolvimento de um projecto como o NONIO-GP. Através desta análise é possível perceber quais os aspectos que poderão por em causa ou limitar o sucesso do projecto, permitindo criar soluções e alternativas que visem minimizar o impacto que os riscos identificados poderão ter sobre o projecto.

Tendo isso em conta, todos os elementos do NONIO foram responsáveis por, com base na sua área de experiência e contexto do projecto, pensarem nos riscos com maior hipótese de comprometer o sucesso do projecto.

Como principais riscos o estagiário identificou a qualidade do processo de migração dos dados dos sistemas de gestão de propinas já existentes (CXA SiGES e XPaga). Este risco advém do facto de ser notório que a informação presente nos sistemas referidos não terá um grau elevado de consistência pelo que a importação da informação necessitará de ser tratada com rigor para que não existam erros que no futuro possam criar problemas de difícil resolução.

A dependência da plataforma de pagamentos da Agência para a Modernização da Administração foi também um dos riscos apontados pelo estagiário. A dependência deste sistema para a criação de referências e para a notificação de pagamentos realizados, poderá ser um factor que condicione o sucesso da aplicação caso não sejam encontradas soluções para prevenir por exemplo, um tempo de inactividade do deste serviço.

Outro risco apontado pelo estagiário foi a constante mudança da regulamentação do processo de gestão de propinas. Tendo em conta que toda a análise de requisitos é feita para acomodar a regulamentação em vigor, uma alteração substancial desta poderia comprometer o sucesso do projecto no futuro.

Num grupo onde existem elementos de áreas completamente distintas, corre-se o risco de que a mensagem transmitida por um membro de uma determinada área não seja bem interpretada por um membro de uma área distinta. Este problema de comunicação poderá ser considerado um risco já que poderá levar a uma análise de requisitos deficiente ou incompleta o que poderá por em causa o sucesso do projecto.

O último risco identificado pelo estagiário passa pela resistência à mudança que é uma característica presente na maioria dos seres humanos. As aplicações CXA SiGES e XPaga são já integrantes do dia-a-dia das pessoas que trabalham na área académica e financeira pelo que, apesar de o projecto NONIO-GP ter o objectivo de melhorar e facilitar o trabalho a essas pessoas, poderá haver alguma resistência por parte das mesmas.

Na opinião do estagiário, esta actividade foi bastante interessante pois permitiu ver as prioridades e preocupações de elementos de áreas distintas. Curioso verificar que, no entanto, a migração de dados é um risco apontado por todos os elementos.

Reuniões de Acompanhamento na Reitoria da Universidade Coimbra

Tendo em conta a importância do projecto NONIO-GP, foram realizadas ano lectivo cinco reuniões de acompanhamento do desenvolvimento do projecto (três no primeiro semestre e duas no segundo semestre). Nestas reuniões para além dos membros das áreas académica, financeira e informática, que normalmente participam nas reuniões de análise de requisitos, participam também os chefes de divisão das áreas referidas bem como

o Reitor da Universidade de Coimbra, o Professor João Gabriel ou, em sua representação, o Vice-reitor Henrique Madeira.

A participação nestas reuniões foi uma oportunidade tremenda e extremamente motivadora para o estagiário já que nestas reuniões para além do acompanhamento que é feito ao progresso do projecto são tomadas as decisões de alto nível que têm como objectivo simplificar e homogeneizar todo o processo de gestão de propinas de todas as faculdades da Universidade de Coimbra.

Capítulo 4

Conclusões

“With every effort, I learned a lot. With every mistake and failure, not only mine, but of those around me, I learned what not to do. I had more than a healthy dose of fear, and an unlimited amount of hope, and more importantly, no limit on time and effort.” – Mark Cuban

Com o término do estágio é possível retirar algumas conclusões não só do trabalho desenvolvido mas também de toda a evolução técnica e pessoal do estagiário.

Em termos pessoais esta foi uma experiência que permitiu ao estagiário melhorar o seu relacionamento interpessoal, já que durante todo o estágio contactou com diferentes tipos de pessoas, com diferentes tipos de conhecimento e de diversas áreas, desde funcionários da tesouraria, a elementos da área académica e financeira, bem como chefes das divisões referidas, elementos de empresas externas e até próprio Reitor e Vice-Reitor da Universidade de Coimbra. Para além disso, toda a experiência de trabalhar numa equipa tão numerosa como o NONIO, onde cada pessoa tem as suas particularidades, foi também um aspecto novo para o estagiário e que certamente irá ajuda-lo em situações futuras.

A experiência de trabalhar num ambiente regrado e com grande exigência a nível de qualidade foi também uma situação nova para o estagiário. Nunca tendo trabalhado num ambiente deste género, o estagiário percebeu a diferença entre o desenvolvimento de trabalhos académicos e o desenvolvimento num projecto real, onde o mínimo pormenor conta e não pode ser negligenciado.

Existiu, sem dúvida, uma evolução na capacidade de trabalho e capacidade de gestão de tempo. Isto deveu-se ao facto ao longo de todo o ano o estagiário ser confrontado com diversas exigências simultâneas que exigiam uma organização de tempo adequada. Todas essas exigências resultaram em que, por vezes, o estagiário cometesse erros, procurando este, no entanto, tirar lições e crescer a partir deles.

Em termos técnicos o estagiário melhorou todos os seus conhecimentos relativamente às tecnologias associadas ao projecto. Nunca tendo trabalhado com as tecnologias *Struts* e *Hibernate*, o estagiário pesquisou e estudou o funcionamento e boas práticas do uso das tecnologias, estando agora à vontade com grande parte das especificidades das mesmas. Participando também no processo de criação da arquitectura do projecto, o estagiário adquiriu conhecimentos sobre quais as principais preocupações e pormenores a ter conta aquando a definição da mesma. O processo de integração com SAP-FI permitiu ao estagiário relembrar conhecimentos sobre criação e gestão de *webservices* bem como a criação de *jobs* automáticos.

No que toca ao trabalho desenvolvido, o estagiário sente que todos os objetivos que lhe foram propostos ao longo do ano foram atingidos com sucesso, desenvolvendo a todos os níveis, desde a especificação de requisitos e prototipagem, escrita de documentação, criação de modelo de dados e implementação de funcionalidades.

Ao longo deste documento foram sendo apresentadas contagens de linhas de código produzidas em cada uma das diferentes funcionalidades ou módulos de que o estagiário foi responsável. Como pudemos verificar, a elevada complexidade do módulo de gestão de conta corrente, no que toca a todos os processos que tem de suportar, levou a que isso também se reflectisse no volume de trabalho necessário na implementação deste módulo.

A integração entre NONIO e SAP-FI foi uma oportunidade tremenda para o estagiário aprender e crescer já que lhe foi dada uma grande autonomia e depositado um elevado grau de confiança para a resolução de todos os problemas que foram surgindo. O estagiário foi responsável por representar o NONIO em toda a comunicação com a empresa

externa e com os membros da área de gestão financeira, estando sempre disponível para pensar e colaborar na resolução das dificuldades encontradas.

Apesar dos problemas referidos que motivaram alguns atrasos, a integração de NONIO com SAP-FI já é uma realidade (e estará em funcionamento na sua plenitude a tempo da gestão de propinas de 2012/2013), correspondendo o estagiário a toda a confiança que lhe foi depositada.

A mudança de planeamento para o segundo semestre acabou, na opinião do estagiário, por ser benéfica já que ofereceu desafios mais estimulantes visto que o módulo de gestão de conta corrente oferece um grau de complexidade mais elevado, exigindo assim mais criatividade e conhecimento aprofundado do tema de gestão de propinas, não só no levantamento de requisitos mas também na prototipagem e implementação. A implementação do módulo de listagens e estatísticas seria, sem dúvida, uma tarefa menos estimulante já que não existiria espaço à criatividade visto que o trabalho passaria apenas por apresentar um conjunto de listagens pedidas pelas áreas de gestão académica e financeira.

A implementação das alterações necessárias no módulo de alunos apresentou-se como um desafio novo já que, pela primeira vez durante todo o ano, o estagiário teve de alterar código já existente e não criar uma solução de raiz. Sendo esta uma situação que certamente acontecerá várias vezes no futuro, foi mais uma oportunidade para o estagiário evoluir.

No que toca à consulta dos dados financeiros na *InforEstud@nte* (módulo de propinas e emolumentos) esta irá ser uma versão menos detalhada do módulo de gestão de conta corrente pelo que a construção do mesmo será um reaproveitamento total do trabalho realizado pelo estagiário ao longo do ano.

A nível da especificação é claro agora para o estagiário que esta é sem dúvida uma das fases mais importantes de todo o processo de desenvolvimento. A tarefa não é, no entanto, fácil. Todo o processo de lidar com cliente e perceber exactamente o que se pretende e quais as exigências deste, é um processo moroso e complexo que se não for feito com qualidade poderá comprometer todo o futuro de um projecto.

Todo o trabalho desenvolvido procurou por, primeiro dar uma base sólida ao projecto e segundo implementar as funcionalidades base que garantam que o NONIO possa gerir as propinas para o ano lectivo de 2012/2013. Estes dois grandes objectivos foram, na opinião o estagiário, concluídos com sucesso.

Anexos

Nesta secção estão enumerados os anexos entregues juntamente com o presente documento.

Anexo 1 Documentos de Processo

Nonio_ProcessoDesenvolvimento_2009-06-11_v1.2r.pdf

Nonio_PlanoGarantiaQualidade_2009-01-10_v1.5r.pdf

Nonio_PlanoControloAlteracoes_2009-06-11_v1.2r.pdf

Anexo 2 Arquitectura e Design Genérico

Nonio_ArquitecturaDesignGenerico_2009-01-24_v1.4r.pdf

Anexo 3 Documento de Especificações Adicionais

Nonio_DocumentoEspecificacoesAdicionais_PlanosPagamento.pdf

Nonio_DocumentoEspecificacoesAdicionais_GestaoContaCorrente.pdf

Anexo 4 Pedido de Alteração

NONIO_PA_2012-01-17_442_ModuloPlanosPagamento.pdf

NONIO_PA_2012-04-19_554_ModuloAlunos.pdf

Anexo 5 Princípios Gestão Financeira

NONIO_Principios_Gestao_Financeira.pdf

Anexo 6 Listagem de Reuniões

Listagem_Reunioes.pdf

Anexo 7 Modelo de Dados

NONIO_ModeloDadosConceptual.pdf

Anexo 8 Protótipo

Referências

No momento da entrega do presente documento, todas as hiperligações abaixo encontravam-se disponíveis.

- [1] **MVC** - <http://msdn.microsoft.com/en-us/library/ff649643.aspx>
- [2] **Struts** - <http://struts.uw.hu/>
- [3] **Spring** - <http://www.springsource.org/>
- [4] **Hibernate** - <http://www.hibernate.org/>
- [5] **CXA SiGES** - <http://www.digitalis.pt/index.php/produtos/siges/77#cxa>
- [6] **SAP** - <http://www.sap.com/portugal/index.epx>
- [7] **Regulamento Propinas**
http://www.uc.pt/academicos/regulamentos/docs_uc/regulamento_propinas_UC
- [8] **AMA** - <http://www.ama.pt/>
- [9] **PT-SI** - <http://www.ptsi.pt/PTSI/Default.html>
- [10] **SIBS** - <http://www.sibs.pt/pt/>
- [11] **UNICRE** - <http://www.unicre.pt/index.html>
- [12] **Optimistic Locking**
http://docs.jboss.org/jbossas/docs/Server_Configuration_Guide/4/html/TransactionJTA_Overview-Pessimistic_and_optimistic_locking.html
- [13] **System Change Number**
<http://www.databasesecurity.com/dbsec/oracle-forensics-scns.pdf>
- [14] **Enterprise Integration Patterns** - <http://eapatterns.com/>