

Pedro Vinhas Laginha dos Ramos

SGIP – Sistema de Gestão de Informação Pedagógica
Relatório de Atividade Profissional



2016

Pedro Vinhas Laginha dos Ramos

SGIP – Sistema de Gestão de Informação Pedagógica

Relatório de Atividade Profissional

Mestrado em Engenharia Elétrica e Eletrónica
Ramo de Tecnologias de Informação e Telecomunicações

Trabalho efetuado sob a orientação de:
Prof. Doutor João Rodrigues

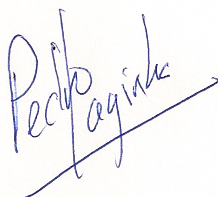


2016

SGIP – Sistema de Gestão de Informação Pedagógica

Declaração de autoria do trabalho

Declaro ser o autor deste trabalho, que é original e inédito. Autores e trabalhos consultados estão devidamente citados no texto e constam da listagem de referências incluída.



(Pedro Vinhas Laginha dos Ramos)

© 2016, *Pedro Vinhas Laginha dos Ramos*

A Universidade do Algarve reserva para si o direito, em conformidade com o disposto no Código do Direito de Autor e dos Direitos Conexos, de arquivar, reproduzir, e publicar a obra, independentemente do meio utilizado, bem como de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição para fins meramente educacionais ou de investigação e não comerciais, conquanto seja dado o devido crédito ao autor e editor respetivos.

*À minha esposa Angela
e aos nossos filhos Fernando e Marco*

Se hace camino al andar
(O caminho faz-se caminhando)
António Machado, Proverbios y Cantares, Campos de Castilla

Agradecimentos

A primeira nota de agradecimento vai diretamente para a minha esposa, Ângela, pelo apoio incondicional que sempre tem dado na prossecução do meu caminho. Usualmente o caminho não se faz sozinho e caminhar acompanhado torna tudo mais fácil.

A segunda nota de agradecimento é para o meu colega de trabalho de quase uma década, Luís Sousa. Companheiro de desenvolvimento do SGIP desde o primeiro momento, não tenho adjetivos suficientes para qualificar o seu desempenho e dedicação profissional. E, ainda muito mais que isso, a amizade que conseguimos construir. Sem ele dificilmente estaria a escrever o presente relatório.

Uma palavra de gratidão também para com os restantes colegas que colaboraram no projeto SGIP, Sérgio Baltazar e Carlos Gomes, pela paciência e amizade, e ainda pelo contributo no desenvolvimento do projeto, ainda que por apenas alguns meses, mas tornando possível a realização do mesmo. Ao Carlos ainda agradeço a forma como me incentivou a prosseguir com o presente relatório.

Um obrigado muito especial à Maria João Cruz e à Célia Ramos pela insistência na realização deste trabalho, pela sua amizade, por toda a sua paciência, mas também por todas as palavras de incentivo e pela colaboração e espírito colaborativo. Agradeço também à Emília Pacheco pela sua paciência e pela sua sapiência, a quem devo a utilização do *Mendeley*.

Uma nota de gratidão também para o meu amigo Joel Guerreiro, Diretor dos Serviços de Informática da Universidade do Algarve, por todo o seu apoio e também pelas muitas palavras de incentivo.

Ainda uma palavra de agradecimento ainda para os colegas da equipa de informática da Universidade do Algarve, com quem tenho o prazer e a sorte de partilhar o meu dia-a-dia profissional e ainda a sua amizade. Agradeço também aos restantes colegas docentes de informática da Escola Superior de Gestão, Hotelaria e Turismo da Universidade do Algarve, com quem tenho o prazer de partilhar alguns almoços, muita conversa e boa disposição e sobretudo muita amizade.

Um enorme obrigado ao colega e amigo do Hospital de Faro, Paulo Simões, por toda a ajuda que nos forneceu sobretudo no arranque do projeto, tendo, inclusivamente, proporcionado uma pequena formação no local de trabalho, transmitindo-nos parte dos seus conhecimentos em Symphony.

Por fim, sem que esta seja uma escala de importância, agradeço ao meu orientador, João Rodrigues, pela paciência, pelo seu profissionalismo e pela sua amizade. A sua resposta, sempre pronta, ajudou muito a que fosse possível concluir este trabalho.

A todos sem exceção que de alguma forma colaboraram e interagiram pessoal ou profissionalmente comigo e partilham a minha amizade, um grande obrigado.

Resumo

Toda a tomada de decisões carece de informação atempada e confiável. A informação pedagógica é muito importante para uma organização de ensino superior no sentido em que permite perspetivar a alocação de recursos para a prática da atividade letiva.

A inexistência de soluções internas e externas adequadas, com o custo e características pretendidas, determinaram que o caminho passaria pelo desenvolvimento de uma aplicação interna. Assim, a Universidade do Algarve colocava o desafio de tornar a informação e os procedimentos, existentes de uma forma dispersa, desorganizada e diferente em cada unidade orgânica, numa base universal normalizada, transversal para toda a organização.

Esse desafio foi corporizado numa aplicação denominada Sistema de Gestão de Informação Pedagógica, conhecida simplesmente por SGIP. Essa aplicação deveria ser composta por três vetores essenciais: a gestão do serviço docente, a gestão dos espaços e a gestão das agendas e sumários. No final do desenvolvimento, apenas o primeiro foi completado e colocado em produção.

O presente documento aborda o processo de escolha da solução a adotar pela equipa e das ferramentas de base utilizadas no desenvolvimento da aplicação. Também dá a conhecer algumas características da aplicação SGIP, essencialmente aquelas em que o autor teve uma participação preponderante.

Por fim, apresenta as conclusões da implementação da aplicação na organização. A aplicação apresentou alguns problemas devido à sua complexidade interna e ao fim do seu desenvolvimento prematuro, no entanto dotou a organização de uma linguagem e procedimentos normalizados no que respeita à gestão do serviço docente e tornou-se uma aplicação verdadeiramente charneira no fluxo de informação da UAlg. Tais características revelaram-se muito importantes na implementação da nova aplicação de gestão académica e pedagógica.

Termos-chave: *SGIP, informação pedagógica, aplicação web, desenvolvimento software.*

Abstract

The whole decision-making needs timely and reliable information. The educational information is very important for an organization of higher education, allowing to preview the allocation of resources for the practice of teaching activity.

The lack of appropriate internal and external solutions at a cost and with the desired characteristics, determined that the development of an internal application was the way. Thus, the University of the Algarve challenge was to transform the existing information and procedures, dispersed, unorganized and different in each organizational unit, on universal standardized platform, transversal throughout the organization.

This challenge has been embodied in an application called Educational Information Management System, known simply as SGIP. This application should consist of three main vectors: the teaching service management, space management and the management of agendas and summaries. At the end of development, only the first has been completed and put into production.

This paper states the solution process and the team choices that were taken and the basic tools used in application development. It also presents some application features of SGIP, essentially those where the author had a preponderant participation.

Finally, it puts forward the conclusions of application implementation in the organization. The application presented some problems due to its internal complexity and the early end of its development, however provided the organization with standard terminology and procedures for the teaching service management and became a unavoidable information provider in the UAlg information flow. Such features proved to be very important in the implementation of the new application of academic and teaching management.

Keywords: *SGIP, pedagogical information, web application, software development.*

Índice Geral

AGRADECIMENTOS	vi
RESUMO	vii
ABSTRACT	viii
ÍNDICE GERAL	ix
ÍNDICE DE FIGURAS E QUADROS	xiii
LISTA DE ABREVIATURAS, SIGLAS E ACRÓNIMOS	xiv
1. INTRODUÇÃO	1
2. VISÃO GERAL	4
2.1. A GESTÃO PEDAGÓGICA DE 2008 A 2016	4
2.2. TIPOLOGIAS DE APLICAÇÕES	6
2.3. SERVIDORES WEB	7
2.4. LINGUAGENS E <i>FRAMEWORKS</i>	8
2.5. SERVIDOR DE BASE DE DADOS	9
2.6. EDITORES DE DESENVOLVIMENTO INTEGRADO (IDE)	10
2.7. FERRAMENTAS DE COLABORAÇÃO E GESTÃO DE PROJETO	12
2.8. SISTEMAS DE CONTROLO DE VERSÃO	13
2.9. METODOLOGIAS	14
2.10. ESTRUTURA DE DESENVOLVIMENTO DE APLICAÇÕES	20
3. GRUPO DE TRABALHO E METODOLOGIAS	22
3.1. GRUPO DE TRABALHO	22
3.2. TIPOLOGIA DA APLICAÇÃO	23
3.3. SERVIDOR WEB	24
3.4. LINGUAGEM E FRAMEWORK	24
3.5. SERVIDOR DE BASE DE DADOS	26
3.6. EDITOR DE DESENVOLVIMENTO INTEGRADO (IDE)	26
3.7. FERRAMENTAS DE COLABORAÇÃO E GESTÃO DE PROJETO	27
3.8. SISTEMA DE CONTROLO DE VERSÃO	29
3.9. METODOLOGIA	29

3.10. ESTRUTURA DE DESENVOLVIMENTO DA APLICAÇÃO	32
4. SISTEMA DE GESTÃO DE INFORMAÇÃO PEDAGÓGICA	35
4.1. DADOS EXTERNOS DE BASE	35
4.1.1. INTRODUÇÃO	35
4.1.2. IMPORTAÇÃO	36
4.1.3. TRANSFORMAÇÃO	36
4.1.4. SINCRONIZAÇÃO DE CURSOS E DISCIPLINAS	38
4.2. APLICAÇÃO E CONCEITOS	39
4.2.1. INTRODUÇÃO	39
4.2.2. CONTROLO DE UTILIZADORES	40
4.2.2.1. Acesso ao SGIP	40
4.2.2.2. Sistema de permissões	41
4.2.3. GRUPOS DE DOCENTES	43
4.2.4. PLANOS DE ESTUDO E UNIDADES CURRICULARES	45
4.2.5. TURNOS E SUB-TURNOS	47
4.2.6. LECIONAÇÃO CONJUNTA	48
4.2.7. AGREGAÇÃO DE DISCIPLINAS	48
4.2.8. ÁREAS CIENTÍFICAS	50
4.2.9. DISTRIBUIÇÕES DE SERVIÇO DOCENTE	51
4.2.9.1. Processo de serviço docente	51
4.2.9.2. Página de gestão da Distribuição de Serviço Docente	52
4.2.9.3. Criação	54
4.2.9.4. Construção e alteração	55
4.2.9.4.1. Organização da página de edição da DSD	55
4.2.9.4.2. Coeficiente de trabalho noturno	57
4.2.9.4.3. Horas semanais, horas totais e horas equivalentes	57
4.2.9.4.4. Cálculo das horas equivalentes	58
4.2.9.4.5. Atribuição de turnos	61
4.2.9.4.6. Definição de outras atividades	62
4.2.9.5. Anulação	62
4.2.9.6. Visualização	63
4.2.9.7. Conclusão	63
4.2.10. WORKFLOW DE APROVAÇÃO DAS DISTRIBUIÇÕES DE SERVIÇO DOCENTE	63
4.2.11. ABERTURA DO ANO LETIVO	65
4.2.11.1. Procedimento de abertura do ano letivo	65

4.2.11.2. Interface de lançamento de ano letivo	68
4.2.11.3. Interface de períodos de funcionamento	69
4.2.12. INTERFACES DE GESTÃO DE DADOS	70
4.2.12.1. Introdução	70
4.2.12.2. Interface de criação e associação de cursos	71
4.2.12.3. Interface de criação e associação de disciplinas	73
4.2.12.4. Interface de anulação de operações sobre cursos	74
4.2.12.5. Interface de anulação de operações sobre disciplinas	75
4.3. EXPLORAÇÃO DOS DADOS	75
4.3.1. CONSULTA E IMPRESSÃO DE DISTRIBUIÇÃO DE SERVIÇO DOCENTE	75
4.3.1.1. Introdução	75
4.3.1.2. Distribuição de Serviço Docente do docente	76
4.3.1.3. Distribuição de Serviço Docente do curso	77
4.3.1.4. Impressão de DSD composta	79
4.3.2. WEB SERVICES DE LEITURA	80
4.3.2.1. Introdução	80
4.3.2.2. Métodos desenvolvidos – 1ª fase	81
4.3.2.3. Desempenho e sistema de cache	82
4.3.2.4. Métodos desenvolvidos – 2ª fase	83
4.4. TRABALHOS INACABADOS OU NÃO INICIADOS	84
4.4.1. GESTÃO DE ESPAÇOS	84
4.4.2. AGENDAS	85
4.4.3. HORÁRIOS E SUMÁRIOS	86
<u>5. CONCLUSÃO</u>	<u>87</u>
<u>REFERÊNCIAS</u>	<u>90</u>
<u>ANEXOS</u>	<u>94</u>
ANEXO I. DESPACHO REITORAL RT.08/2007	94
ANEXO II. DOCUMENTOS DA ANÁLISE DA INFORMAÇÃO PEDAGÓGICA	94
ANEXO III. DESPACHO REITORAL RT.1/08	94
ANEXO IV. PRODUCT BACKLOG	94
ANEXO V. DESPACHO REITORAL RT.008/2012	94
ANEXO VI. FICHEIRO CONFIGURAÇÃO ABERTURA ANO LETIVO 2012/2013	94
ANEXO VII. PROCEDIMENTO PARA ABERTURA DE UM NOVO ANO LETIVO	94
ANEXO VIII. SGIP WEB SERVICES	94

Índice de Figuras e Quadros

FIGURA 2.1- FLUXO DO SCRUM.	18
FIGURA 3.1 - EXTRATO DO PRODUCT BACKLOG.	31
FIGURA 3.2 – EXEMPLO DE BURNDOWN CHART.	32
FIGURA 4.1 - PÁGINA DE ENTRADA DO SGIP.	40
FIGURA 4.2 - PÁGINA PRINCIPAL PARA UM GESTOR DO SGIP.	41
FIGURA 4.3 - EDIÇÃO DE ATRIBUIÇÃO.	43
FIGURA 4.4 - EDIÇÃO DE GRUPO DE DOCENTES.	44
FIGURA 4.5 - PLANO DE ESTUDOS.	45
FIGURA 4.6 - DISCIPLINA.	46
FIGURA 4.7 - ESCOLARIDADES E TURNOS DA DISCIPLINA.	47
FIGURA 4.8 - AGREGAÇÃO DE DISCIPLINAS.	49
FIGURA 4.9 - INTERFACE DE GESTÃO DA DSD PARA UM ADMINISTRADOR.	53
FIGURA 4.10 - PÁGINA DE EDIÇÃO DE DSD.	56
FIGURA 4.11- DIAGRAMA DE TRANSIÇÃO DE ESTADOS DO SGIP.	64
FIGURA 4.12 - EXTRATO DE UM EXEMPLO REAL DE UM FICHEIRO DE CONFIGURAÇÃO UTILIZADO PARA A CÓPIA DOS DADOS RELATIVOS AO ANO LETIVO DE 2011/2012 PARA 2012/2013.	67
FIGURA 4.13 - INTERFACE DE LANÇAMENTO DE ANO LETIVO.	68
FIGURA 4.14 - INTERFACE DE EDIÇÃO DE PERÍODOS DE FUNCIONAMENTO.	69
FIGURA 4.15 - INTERFACE DE CRIAÇÃO E ASSOCIAÇÃO DE CURSOS.	71
FIGURA 4.16 – EM CIMA, EXEMPLO DE CRIAÇÃO DE CURSO. EM BAIXO, EXEMPLO DE ASSOCIAÇÃO DE CURSOS.	72
FIGURA 4.17 - INTERFACE DE CRIAÇÃO E ASSOCIAÇÃO DE DISCIPLINAS.	73
FIGURA 4.18 - EXEMPLO DE ANULAÇÃO DE CURSO CRIADO.	74
FIGURA 4.19 - EXEMPLO DE DSD DO DOCENTE.	77
FIGURA 4.20 - EXEMPLO DE DSD DE CURSO.	78
FIGURA 4.21 - INTERFACE DE IMPRESSÃO DE DSD COMPOSTA.	79

Lista de abreviaturas, siglas e acrónimos

- AD *Active Directory*, termo inglês que usualmente não é traduzido para português.
- CLI *Command Line Interface*, termo inglês para designar uma instrução em linha de comandos.
- CRUD *Create, Read, Update and Delete*, termo inglês para Criar, Ler, Atualizar e Apagar.
- DSD Distribuição de Serviço Docente.
- FCT Faculdade de Ciências e Tecnologia.
- HTTP *HyperText Transfer Protocol*, termo inglês para designar Protocolo de Transferência de Hipertexto, protocolo de transferência de informação na internet.
- IDE *Integrated Development Environment*, termo inglês para designar Ambiente de Desenvolvimento Integrado.
- PDF *Portable Document Format*, termo inglês para designar um formato de documento portátil ou universal criado pela Adobe.
- PE Plano de Estudos.
- REST *REpresentational State Transfer*, termo inglês para designar Transferência de Estado Representacional.
- SGBD Sistema de Gestão de Bases de Dados.
- SGIP Sistema de Gestão de Informação Pedagógica.
- SOAP *Simple Object Access Protocol*, termo inglês para designar Protocolo Simples de Acesso a Objetos.
- UAlg Universidade do Algarve.
- UC Unidade Curricular.
- UO Unidade Orgânica.
- VCS *Version Control System*, termo inglês para designar Sistema de Controlo de Versão.

1. Introdução

Desde o início do século que as instituições de ensino superior, sobretudo as que existem nas periferias fora dos grandes centros urbanos, apresentam grandes dificuldades no equilíbrio das suas contas e na geração de receitas. Não será novidade que tal se deve à maior dificuldade em captar alunos em determinadas áreas, na quebra das dotações orçamentais, na diminuição da capacidade de gerar receitas próprias, entre algumas outras razões que poderiam ser aqui enumeradas, mas que não importa, de todo, dirimir.

A Universidade do Algarve (UAlg) não foi imune a tais dificuldades, sendo que uma gestão racional tem ajudado a ultrapassá-las ao longo destes anos. Nesse contexto de racionalização, surge em princípios de 2007 a necessidade de uniformização da informação pedagógica e da respetivas práticas como forma de dotar a organização de informação de melhor qualidade e de obtenção mais rápida.

De forma a responder à referida necessidade, tal como outras identificadas na altura e não importantes no contexto do presente documento, a UAlg avançou com a constituição de grupos de trabalho para estudar cada uma dessas necessidades específicas. A formalização dos grupos de trabalho foi oficializada sob a forma de equipas de projeto, em Despacho Reitoral RT.08/2007 (ANEXO I). É importante desde já referir, que todos os Anexos serão fornecidos em formato digital, a menos que seja referido o contrário no corpo deste relatório.

A equipa de projeto da informação pedagógica é designada por “Desenvolvimento do Sistema de Informação para a Avaliação e Qualidade do Ensino”. Essa equipa de projeto será doravante designada de Equipa de Projeto 2.

Dos trabalhos subsequentes da referida equipa de projeto é produzido um conjunto de documentos, nomeadamente os documentos SEP-2.0-H-01, SEP-2.0-A-02, e outros, que se podem consultar no ANEXO II e que representam uma amostra significativa do trabalho produzido. Estes são o resultado da análise funcional por procedimentos, tendo sistematizado o sistema a desenvolver em 3 vertentes principais: a gestão da informação de serviço docente, a gestão dos espaços e a gestão de agendas. A vertente de gestão de

agendas possibilitaria a cada docente, a cada aluno, a cada espaço, a gestão da sua própria agenda, sendo que estas teriam sempre por base os horários.

No contexto das propostas apontadas pelo documento elaborado no seio dos trabalhos da Equipa de Projeto 2, nasce a constituição do grupo de trabalho para o “Desenvolvimento do Sistema de Gestão de Informação Pedagógica”, formalizado em Despacho Reitoral RT.1/08 (ANEXO III). A constituição desse grupo de trabalho tem em vista a criação de uma aplicação informática, transversal à organização, que será designada por Sistema de Gestão de Informação Pedagógica, ou SGIP, como é por demais conhecida.

O início dos trabalhos ocorreu nos primeiros dias de 2008 e consistiu, em primeiro lugar, na leitura e análise do documento produzido pela Equipa de Projeto 2, tendo sido seguida por uma fase muito rápida de perceção do mercado, designadamente das soluções existentes. Como a UAlg não tinha uma situação financeira que permitisse a aquisição de uma aplicação nem queria dispor do tempo necessário a efetivar todos os procedimentos burocráticos com vista a tal – recorda-se que o despacho RT.1/08 refere que a “aplicação a elaborar deverá permitir acolher informações respeitantes ao início do ano letivo 2008/09” –, o grupo de trabalho concentrou os primeiros meses do ano na definição e configuração de todo o ambiente necessário ao desenvolvimento da aplicação.

A aplicação teve o final do seu desenvolvimento no início do ano de 2011, sendo que, apesar de ter sido cumprido o ano de entrada em funcionamento em 2008/2009, este apenas serviu parcialmente a organização e deverá ser encarado com o ano 0 do SGIP. Na altura do término do desenvolvimento, a equipa de desenvolvimento encontrava-se sensivelmente a meio da produção da gestão de espaços e acumulava o desenvolvimento com o suporte a todo o sistema, incluindo aplicações dependentes. Os anos letivos subsequentes foram de funcionamento pleno da aplicação, até ao final do ano de 2014/2015.

As únicas exceções nesta paragem do desenvolvimento foram a correção de erros críticos e ainda a produção de meios de exploração dos dados por aplicações terceiras. Estes meios foram produzidos e disponibilizados sob a forma de *web services* que

ficaram acessíveis com mecanismos de autenticação e autorização, bem como a transmissão em canal seguro.

O presente documento visa a apresentação de todo o procedimento de desenvolvimento da aplicação e dos seus meios de exploração. No capítulo 2 apresenta-se uma breve descrição do estado da arte na altura em que se iniciaram os trabalhos e no presente no que diz respeito às ferramentas e às metodologias, no capítulo 3 está descrita a constituição do grupo de trabalho e as suas opções relativas à escolha das ferramentas e da organização da equipa, o capítulo 4 descreve o SGIP com todos os seus aspetos e, por fim, as conclusões e o trabalho a realizar apresentam-se no capítulo 5. Reforça-se mais uma vez que, devido à dimensão dos mesmos, todos os Anexos estão disponível no formato digital, tendo sido entregues em conjunto com este relatório. Por último, no ANEXO IX é apresentado o Curriculum Vitae do autor deste Relatório de Atividade Profissional.

2. Visão Geral

2.1. A gestão pedagógica de 2008 a 2016

No presente capítulo, pretende-se fazer um estado da arte resumido à data em que se iniciou o projeto até aos dias de hoje. Uma realidade é inquestionável e deverá ser transmitida: a gestão pedagógica tal como era entendida na Universidade do Algarve, na qual se inclui o serviço docente, não existia como software comercial que se adquirisse. Entenda-se por software comercial aquele que se pode adquirir por qualquer meio, numa loja ou vendedor, pronto a instalar e configurar para funcionar num ambiente individual ou institucional.

Este facto limitava muito a procura uma vez que, não existindo aplicação comercial de suporte ao serviço docente, as soluções que existiam residiam no meio universitário. O que era conhecido à data era o software SIGARRA (SIGARRA, 2016) desenvolvido no seio da Universidade do Porto agregando um conjunto de módulos iniciais. Na página indicada do produto pode constatar-se que a adoção da aplicação foi gradual ao longo dos anos pelas várias unidades orgânicas e serviços, sendo que as instâncias do SIGARRA mais recentes datam de 2013. Deve referir-se que este não foi adquirido, nem sequer existiu qualquer tentativa de assistir a uma demonstração, por não existir interesse da organização (UAlg) para tal.

De resto, sabia-se que a maioria das unidades orgânicas da UAlg elaboravam as distribuições de serviço docente de forma *ad-hoc*, de acordo com as necessidades específicas de cada uma e seguindo os conceitos culturalmente desenvolvidos no seu seio. A maior parte dessa gestão assentava em folhas de cálculo, desenvolvidas e completadas ao longo dos anos e que serviam, recursivamente, para alimentar o serviço docente ano letivo após ano letivo.

Não se pode considerar que o trabalho de estado da arte tenha sido exaustivo na altura em que constituiu o grupo de trabalho, uma vez que a decisão de desenvolver um produto de raiz estava quase formada na mente dos decisores, sendo que o tempo disponibilizado à equipa para cumprir o objetivo de desenvolvimento foi muito pequeno. Podem apresentar-se, assim e de uma forma sintética, as principais razões para que o estudo do estado da arte não ter sido mais completo:

1. Algum conhecimento que a equipa de projeto detinha sobre a realidade da gestão de serviço docente e que, de uma forma informal, apontava para a inexistência de soluções satisfatórias.
2. A constatação da multiplicidade de realidades internas, no seio das unidades orgânicas da UAlg, e da grande dificuldade em fazer uma implementação de um produto que não fosse ao encontro de uma realidade semelhante; ou seja, a vontade de adoção de uma aplicação que conseguisse refletir exatamente essas realidades e que fosse inclusivo na ótica da adoção por parte das unidades orgânicas.
3. O *deadline* colocado para a colocação em produção de uma solução foi o principal motivo para tal, uma vez que impedia um estudo exaustivo e impediria certamente a colocação em produção de uma solução de software profissional em tempo.
4. A vontade expressa transmitida em não despender qualquer quantia para compra de produtos ou serviços por parte dos responsáveis da organização.
5. A experiência negativa detida pela UAlg na aquisição de alguns produtos oriundos de outras universidades pelas constantes dificuldades de manutenção e de tempo de resposta adequado.
6. Alguma vontade da organização em, naquele momento, iniciar o desenvolvimento de soluções internas.

Ainda hoje há uma certa dificuldade em perceber o que realmente o mercado tem para oferecer neste capítulo. Sabe-se que existem aplicações que já fazem alguma gestão pedagógica, mas não é claro que o façam com a mesma qualidade e complexidade que a UAlg necessita.

A aplicação adquirida pela UAlg para substituir o SOPHIA (SOPHIA, 2016) na gestão académica e que se encontra no primeiro ano de produção, o SIGES (SIGES, 2016), tem as figuras de gestão pedagógica e de serviço docente. Mesmo sendo uma aplicação bastante completa em termos de características, este módulo em particular beneficiou muito com a experiência que a equipa de desenvolvimento do SGIP partilhou desde a elaboração do caderno de encargos.

O SOPHIA ganhou uma nova versão designada de SOPHIA PLUS e transitou da empresa Conhecer +TI, que entretanto tinha sido criada para a exploração da aplicação, para a Normática¹. Esta aplicação tem, no entanto e pelo que é dado a perceber, um conceito de serviço docente intimamente ligado ao horário, conceito que não é exatamente o seguido pela UAlg.

O SIGARRA (SIGARRA, 2016) já é uma aplicação comercial desde que foi criada a Sysnovare como uma *spin-off* de Universidade do Porto. Esta empresa adicionou a aplicação ao seu portfolio e hoje é oferecida como produto comercial. Os módulos correspondentes às funcionalidades pretendidas para o SGIP encontram-se descritos na aplicação como integrantes dos “Processos de gestão académica”.

Baseado numa *framework open-source* desenvolvida por elementos do Instituto Superior Técnico e designada por FénixEdu (FenixEdu, 2016), também se pode encontrar no mercado a solução qubEDU (qubEDU, 2016) comercializada pela empresa qubit². Esta empresa nasce tendo com base elementos oriundos do Centro de Informática daquela instituição de ensino superior. A solução está em funcionamento em alguns institutos da Universidade de Lisboa e tem toda a componente académica, desconhecendo-se a adequação em relação à gestão pedagógica.

Existem outros produtos ainda desenvolvidos no seio de consultoras como, por exemplo, a Infinite Solutions (Infinite, 2016), a CapGemini (CapGemini, 2016) ou Wiremaze (WireMaze, 2016), que terão certamente algum espaço para crescer em espaço de implementação.

2.2. Tipologias de aplicações

Em termos de arquitetura de aplicação, numa perspetiva de localização e de função, há essencialmente três tipos de paradigma (McNurlin & Sprague, 2005): *standalone*, *peer-to-peer* e *client-server*, podendo afirmar-se que estas duas últimas pertencem à categoria de aplicações distribuídas. De uma forma geral, uma aplicação *standalone* é uma aplicação que não interage com qualquer rede, funciona de forma autónoma. Não é

¹ <http://www.normatica.pt>

² <http://www.qub-it.com>

o tipo de aplicações recomendadas para partilha de informação ou de lógica do processo com vários intervenientes dispersos.

Uma tipologia em rede deverá ser considerada sempre que se pretenda partilhar recursos, sejam eles processos ou dados. Uma tipologia em *client-server* é útil quando se pretende centralizar a lógica do processo ou a localização e a fonte dos dados. Por outro lado, a tipologia de *peer-to-peer* implica que a lógica do negócio esteja em cada uma das aplicações, sendo que uma solução de partilha de dados é sempre de implementação mais complicada.

As aplicações web, ou *web applications*, são descritas de uma forma muito generalista como “um sistema de software acessível na Web” (Jablonski et al., 2004). No fundo, tratam-se de aplicações *client-server* com algumas características particulares, designadamente o facto de libertarem o desenvolvimento da construção de uma aplicação cliente para sistemas operativos diferenciados.

A aplicação cliente a usar numa aplicação *web* é o denominado *browser da web*, ou simplesmente *browser*, sendo que os mais utilizados no mundo inteiro são o *Google Chrome* (Chrome, 2016), o *Mozilla Firefox* (Firefox, 2016), o *Microsoft Internet Explorer* (MSIE, 2016), o *Apple Safari* (Safari, 2016) e o *Opera* (Opera, 2016), considerando a soma de todas as plataformas. O autor do presente relatório há muito que se refere ao *browser* como o *cliente universal*.

2.3. Servidores Web

Considerando uma lógica de aplicação cliente-servidor, as aplicações *web* já eram, no início do projeto, uma realidade muito popular e utilizada para a grande maioria das soluções. A opção de realizar uma aplicação que será consumida num *browser*, que é o que se pode chamar de “cliente universal”, tem a grande virtude de normalizar a componente de comunicação entre o cliente e o servidor e evitar o seu desenvolvimento.

Na altura em que se iniciou o projeto, existiam essencialmente dois servidores que dominavam o mercado: o *httpd* (HTTPD, 2016) da Apache, que pouco a pouco foi adotando o nome *Apache*, e o *IIS - Internet Information Server* (IIS, 2016) da Microsoft.

Se, por um lado, a Apache apresentava um servidor muito configurável e multiplataforma, com um elevado desempenho e flexibilidade, a Microsoft aproveitava o facto do servidor existir em todos os sistemas Windows e estar perfeitamente integrado com os sistemas operativos da empresa.

Se nesse início de 2008 ainda existiam alguns servidores de web mais jovens e que iniciavam a atividade tais como o *lighttpd* (LIGHTTPD, 2016) e que conquistaram o seu lugar no mercado, mais tarde começou a ter uma grande relevância um servidor que se impôs no mercado pelo seu desempenho e uma grande dose de versatilidade, apesar de existir desde 2002: o *NGINX* (NGINX, 2016).

2.4. Linguagens e frameworks

As linguagens estão intimamente ligadas às *frameworks* utilizadas, sendo que, usualmente, essas *frameworks* são escritas em determinada linguagem, utilizada também para o desenvolvimento sobre ela. É uma tarefa difícil fazer um apanhado das linguagens existentes na altura e das respetivas *frameworks* mas se se tiver em consideração estas últimas e a sua popularidade, facilmente se consegue isolar um pequeno conjunto das mais populares.

Reportando à data de início do projeto, são de destacar, pela popularidade que apresentavam, o *CakePHP* (CakePHP, 2016), o *Symfony* (Symfony, 2016), a *Zend Framework* (Zend, 2016) e o *Rubi On Rails* (Rails, 2016). Esta última desenvolvida em *Ruby* (Ruby, 2016) detinha uma boa dose de popularidade, se bem que escrita numa linguagem de programação que não tinha uma grande implantação. Por outro lado, a linguagem *PHP* (PHP, 2016) começava a consolidar o seu processo desenvolvimento como linguagem orientada a objetos, deixando a filosofia inicial de pura linguagem de *scripting*, constituindo uma excelente base de trabalho também para os criadores de *frameworks* como o *CakePHP* ou o *Symfony* indicados.

Hoje em dia, qualquer uma das *frameworks* mencionadas ainda existe e melhoraram as suas características. Do mundo PHP começou a ter muito destaque sobretudo no último ano o *Laravel* (Laravel, 2016), mencionada em vários sítios da internet como a mais

utilizada, sendo que o *Symfony*, agora a apostar na sua versão 3, continua a deter uma grande comunidade e a ser utilizada amiúde.

Não seria de desprezar a apresentação da já existente *ASP.NET* (ASP, 2016), plataforma da gigante Microsoft para o desenvolvimento Web, baseada numa *framework* estruturada designada por *.NET Framework* (Thai & Lam, 2003), podendo ser programada utilizando as linguagens *Visual Basic* ou *Visual C#* (VBVCS, 2016). Nessa altura, a Microsoft apenas disponibilizava as suas ferramentas para os sistemas operativos desenvolvidos pela empresa, perdendo alguma versatilidade por essa via. Na mesma linha, o *Java* (Gosling, 2000) também seria uma linguagem a considerar tendo em conta a sua implantação e a quantidade de sistemas operativos onde seria possível colocar a aplicação, tendo, contudo, conhecidos problemas de desempenho.

2.5. Servidor de base de dados

Por definição, um Sistema de Gestão de Bases de Dados (SGBD) ou *Database Management System* (DBMS) “é um software desenhado para ajudar na manutenção e utilização de grandes coleções de dados, e a necessidade de tais sistemas, bem como a sua utilização, está a crescer rapidamente” (Ramakrishnan & Gehrke, 2000). Tal ideia era proferida no início do século e não poderia considerar-se mais acertada: as necessidades de informação têm crescido a ritmos impressionantes nos últimos anos tendo levado grandes companhias a investir em centros de dados de grande capacidade.

Apenas como referência uma vez que o livro aborda sobretudo o modelo relacional, Ramakrishnan & Gehrke, 2000 descrevem ainda os vários tipos de modelos de SGBD existentes tais como o modelo hierárquico, o modelo de rede, o modelo orientado a objetos e ainda o modelo objeto-relacional. O mesmo autor escreve que, enquanto “há muitos bancos de dados que usam os modelos hierárquicos e de rede, e os sistemas com base nos modelos orientados a objetos e objeto-relacional estão ganhando aceitação no mercado, o modelo dominante hoje é o modelo relacional”.

Tendo em conta essa evidência, na altura em que se iniciou o desenvolvimento do SGIP existiam dois motores que se desatacavam no mundo do *open-source*: o *PostgreSQL*

(PostgreSQL, 2016) e o *MySQL* (MySQL, 2016). Os dois motores de base de dados têm mantido uma disputa interessante ao longo dos anos pela posição de SGBD relacional *open-source* mais utilizado e mais popular. Ambos existem para um leque vasto de sistemas operativos nos quais se incluem os mais populares. Contudo, o *MySQL* tem assumido uma postura mais comercial, com disponibilização de licenciamento pago embora mantendo o produto também com licenciamento *open-source*, mesmo após a sua compra pela Oracle. O *PostgreSQL* tem-se mantido com licenciamento *open-source* puro.

Também neste capítulo também deverão ser mencionados o *Microsoft SQL Server* (SQLServer, 2016) e o *Oracle Database* (Oracle, 2016), poderosos motores de base de dados, embora com licenciamento pago. Embora este último mencionado exista para as principais plataformas servidoras, o SGBD da Microsoft existia e continua a existir apenas para os sistemas operativos desenvolvidos pela empresa, embora se preveja que essa realidade se altere no curto ou médio prazo com uma versão em Linux.

Uma nota de destaque ainda para um *fork* do *MySQL*, liderado pelo principal mentor e que terá ficado pouco satisfeito com a venda do *MySQL* à ORACLE, cujo nome é *MariaDB* (MariaDB, 2016). Não se sabe ainda qual o futuro de tal proposta que ainda é algo jovem e muito idêntica ao sistema que lhe deu origem.

Há poucos anos começaram a ter algum destaque alguns motores de bases de dados capazes de lidar com grandes volumes de dados e deter elevados desempenhos. Esses motores de bases de dados, designados de forma genérica por Bases de Dados NoSQL (NoSQL, 2016), não foram considerados na altura em que se procedeu ao desenvolvimento do projeto mas poderiam bem ser parte ou a totalidade de uma solução atual. O conceito é um chapéu que engloba várias famílias de bases de dados que não serão detalhadas no presente documento.

2.6. Editores de desenvolvimento integrado (IDE)

Como é do conhecimento geral, um programador gasta a maior fatia do seu tempo em torno do código, seja a criar, refatorar, depurar ou simplesmente melhorar. Pode perceber-se então a importância que tem a criação do ambiente para desenvolver.

Importante é também a capacidade de tirar proveito das ferramentas de desenvolvimento e o ganho de tempo que a partilha de conhecimento representa com escolhas acertadas.

Compreendendo bem as necessidades de um profissional dedicado ao desenvolvimento de aplicações, os programadores estenderam o antiquado conceito de editor de texto, como sendo a ferramenta onde se pode escrever código e gerir os respetivos ficheiros, para o conceito de ambiente de desenvolvimento integrado ou IDE (*Integrated Development Environment*). De uma forma muito generalista, pode dizer-se que um IDE é uma ferramenta que é composta principalmente por um editor de texto, por várias ferramentas de automação e por um depurador de código usualmente designado pelo seu termo inglês *debugger*. Uma grande parte dos IDE mais recentes já inclui o completamento de código e verificação de semântica, realizados de uma forma inteligente e sensível à linguagem em edição.

Deve dizer-se ainda que os IDE mais modernos podem incluir muitas outras características tais como um ou vários sistemas de controlo de versão, compiladores ou interpretadores de código, ferramentas de ajuda ao desenvolvimento de um GUI (*Graphical User Interface*), entre muitas outras. Uma outra característica que se reveste de grande importância é a capacidade de integrar bases de dados e a capacidade de as definir, alterar e manipular os seus dados.

Pode afirmar-se, sem prejuízo da verdade, que não existe um grande número de IDE que sejam simultaneamente *open-source* e que se apresentem bastante completos nas suas características. Na altura em que se iniciou o projeto, existiam essencialmente dois IDE, ambos desenvolvidos na linguagem de programação *Java*, com uma comunidade expressiva, detendo suporte a várias linguagens e disponibilizados para várias plataformas: o *Eclipse* (Eclipse, 2016) e o *NetBeans* (NetBeans, 2016).

Do mundo Windows, claramente o *Visual Studio* (VisualStudio, 2016) era a proposta a reter, precisamente quando a Microsoft começou a disponibilizá-lo sem quaisquer custos. Para o desenvolvimento em tecnologias oriundas desta empresa, o IDE detinha e detém todo o tipo de características avançadas aqui especificadas e que qualquer programador procura numa ferramenta de desenvolvimento.

Desde a altura em que se iniciou o desenvolvimento do projeto até aos dias de hoje houve uma certa proliferação de ferramentas deste género, designadamente nos editores de texto mais avançados. Um dos editores de texto mais apreciados hoje em dia no mundo *open-source* é o *SublimeText* (Sublime, 2016), uma vez que detém várias características interessantes e algo inovadoras, desde a característica de ir para qualquer lado, à utilização intensiva de teclas de atalho, aos múltiplos cursores e múltiplas seleções, sem esquecer a capacidade de instalar inúmeros *plugins*, e ainda outras que o tornam um editor de texto muito popular.

Estes editores de texto são, muitas vezes, desenvolvidos pelas grandes companhias ou organizações de grandes dimensões, como o comprovam, por exemplo, o *Brackets* (Brackets, 2016) da Adobe, ou ainda o *Visual Studio Code* (VSCode, 2016) da Microsoft, ou o *Atom* (Atom, 2016) do Github. Outra característica que estes editores costumam ter é serem disponibilizados para os vários sistemas operativos de cliente, sendo que qualquer um deles existe para as 3 principais plataformas: Windows, Mac OS X e Linux.

2.7. Ferramentas de colaboração e gestão de projeto

Existe um grande número de ferramentas de gestão de projeto, sendo que dada a exiguidade do tempo disponível e a inexperiência da equipa de desenvolvimento, não foi realizada uma pesquisa apurada na sua descoberta e teste. Seja como for, na altura e no meio em que a equipa de movia, tomou conhecimento de duas aplicações que seriam das mais utilizadas: o *Redmine* (Redmine, 2016) e o *Trac* (Trac, 2016).

O *Redmine* era largamente utilizado pela comunidade *open-source* e detinha um conjunto de características avançadas, como, por exemplo, ser multi-projeto e deter um grande conjunto de *plugins*, que o tornavam uma aplicação bastante completa. O *Trac*, por outro lado, tinha a outra face, sendo de fácil instalação e gestão e simples para o utilizador.

Mais tarde, uma equipa oriunda do *Redmine* criou um fork deste, nascendo o *Chili Project* (Chili, 2016). Esta aplicação já foi descontinuada, de acordo com a mensagem que se encontra no seu *website*. Também como *fork* do *Redmine*, nasceu o *OpenProject* (OpenProject, 2016) em 2012 e mantém a sua atividade.

Para gestão de projeto apenas, existia na altura o *Gnome Planner* (Planner, 2016) que fornecia a possibilidade de fazer algumas tarefas de gestão de projeto, tais como a definição de recursos humanos, de calendário, alocação de tarefas e gráficos de Gantt, entre outros. Na mesma linha, embora no mundo Windows, o *Microsoft Project* (Project, 2016) detém uma grande popularidade por realizar o mesmo tipo de operações de uma forma muito competente.

2.8. Sistemas de controlo de versão

Os sistemas de controlo de versão (VCS – *Version Control Systems*) são usualmente divididos consoante a abordagem ou modelo em que se inserem de acordo com a filosofia decidida aquando do seu desenvolvimento. Poder-se-ão considerar 3 abordagens ou modelos diferentes, sendo que a primeira é a abordagem local, isto é, para desenvolvimento individual, razão pela qual não é, por vezes, considerada.

Otte, 2009 distingue dois modelos de sistemas de controlo de versão: os sistemas de controlo de versão centralizados (CVCS – *Centralized Version Control Systems*) e os sistemas de controlo de versão distribuídos (DVCS – *Distributed Version Control Systems*). A diferença principal entre eles é onde reside ou pode residir o repositório, sendo que no primeiro modelo o repositório existe sempre centralmente, num endereço acessível por todos os que é suposto a ele acederem, e no segundo modelo não há a necessidade de existir um repositório central, cada utilizador “dispõe do repositório completo no seu computador local, chamado repositório local” (Otte, 2009).

Na altura em que se iniciou o projeto SGIP, existiam muito disseminados o CVS – *Concurrents Version System* (CVS, 2016), e o SVN ou *Subversion* (SVN, 2016), sendo que ambos se enquadram na categoria de sistemas de controlo de versão centralizados. Nessa altura começava a existir uma atenção grande sobre um novo produto iniciado pelas mãos de Linus Torvalds: o GIT (GIT, 2016).

Pode afirmar-se empiricamente, que este último tomou conta dos VCS. Neste momento praticamente toda a comunidade de desenvolvimento utiliza esta aplicação, pelo seu potencial e pela sua versatilidade. Hoje em dia, algumas referências aos VCS surgidos na

mesma altura *Mercurial* (Mercurial, 2016) e *Bazaar* (Bazaar, 2016) poderão ser encontradas mas usualmente todos os autores falam GIT.

Do mundo Windows e da empresa mãe, a Microsoft, surgiu mais tarde o *Version Control* do *Microsoft Visual Studio* (VSVC, 2016) que permite utilizar um proprietário *Team Foundation Version Control* (TFVC) ou ainda o GIT.

2.9. Metodologias

Existem um grande número de metodologias de desenvolvimento de software. Não é objetivo do presente relatório a sua apresentação nem sequer tecer qualquer contexto crítico sobre as mesmas. Mas é importante perceber qual o contexto em que o desenvolvimento do SGIP foi realizado.

A primeira metodologia habitualmente abordada é a conhecida por *waterfall*, apresentada em 1970 por Winston Royce (Royce, 1970). O modelo inicial caracteriza-se por um conjunto de fases sequenciais, como tal o início de uma fase implica o fim da fase precedente. À medida que o passo progride e o *design* se detalha, existirá frequentemente uma interação com os passos anterior e posterior e raramente com passos a maior distância no modelo.

As fases, formalmente definidas por Royce, são as seguintes:

- ***System Requirements***: requisitos de sistema;
- ***Software Requirements***: requisitos do software;
- ***Analysis***: análise do sistema;
- ***Program design***: definição da utilização de base de dados, processadores, esquemas de alocação de memória entre outros;
- ***Coding***: escrita do código da aplicação;
- ***Testing***: fases de teste da aplicação;
- ***Operations***: conformidade e aceitação da aplicação por parte do cliente.

As fases descritas são acompanhadas da produção de muita documentação que reveste o processo de um enorme formalismo. Claramente o *waterfall* não se adequa a um projeto

de longa duração uma vez que durante o ciclo de escrita não se produz qualquer resultado aplicacional. Por outro lado, o “processo não define como responder a um resultado inesperado de qualquer dos processos intermédios” (Schwaber, 1997; Schwaber & Sutherland, 2011).

As metodologias *spiral* (espiral) e *iterative* (iterativa) visam minorar esta questão, embora o façam de uma forma diferente. A primeira metodologia, *spiral*, desenvolvida por (Boehm, 1986), considera fases de desenvolvimento, como a *waterfall*, embora percorridas numa espiral em camadas, entre tarefas sequenciais, representadas por quadrantes num esquema de espiral e que se podem traduzir como:

- Determinar objetivos, alternativas e restrições;
- Avaliar alternativas, identificar e resolver riscos;
- Desenvolver o produto do próximo nível;
- Planear próximas fases.

Contudo, as fases e os seus processos continuam a ser lineares, sendo que “o trabalho de requisitos continua a ser feito na fase de requisitos, trabalho de *design* continua a ser feito na fase de *design*, e por aí adiante, com cada uma das fases consistindo em processos lineares e definidos explicitamente” (Schwaber, 1997).

A metodologia *iterative* flexibiliza, de alguma forma, esta forma demasiado sequencial das fases, percorrendo iterativamente as fases de *waterfall* (aproximadas, sem o formalismo de Royce, 1970) em ciclos em que em cada um apenas se trata de um conjunto bem determinado de funcionalidades. As fases consideradas por Schwaber, 1997 são:

- Análise de requisitos;
- *Design* preliminar;
- *Design* detalhado;
- Escrita de código;
- Teste do módulo;
- Teste do sistema.

A abordagem do modelo iterativo “melhora o controlo de custos, assegura a entrega dos sistemas (embora subsistemas), e flexibiliza [o processo] de uma forma geral” (Schwaber, 1997). Este autor refere ainda que, apesar disso, “a abordagem Iterativa ainda espera que os processos de desenvolvimento subjacentes estejam definidos e sejam lineares”.

Basicamente, as categorias de metodologias abordadas até este ponto assumem uma característica de previsibilidade no processo de desenvolvimento. De um pequeno grupo que se juntou para discutir as falhas sistemáticas do uso de metodologias e a necessidade de encontrar uma melhor abordagem nasceu o manifesto Agile (Agile, 2016). Este movimento tem como princípios a valorização “dos indivíduos e interações sobre os processos e as ferramentas, software em funcionamento sobre documentação compreensiva, colaboração do cliente sobre negociação contratual, resposta à mudança sobre o seguimento de um plano”. Ou seja, Agile não é uma metodologia, é um movimento que procura formas diferentes para gestão de projeto que respondam à imprevisibilidade.

As metodologias que integram esta filosofia são usualmente conhecidas como modelos Agile. Duas das mais conhecidas são o XP, de *Extreme Programming*, desenvolvida por Kent Beck (Beck, 2000), e o SCRUM (Lacey, 2012; Schwaber, Sutherland, Beedle, 2013; Schwaber, 2014). Esta última metodologia foi inicialmente mencionada por Takeuchi & Nonaka, 1986, mas foi com trabalhos de Ken Schwaber e de Jeff Sutherland, que tal como Kent Beck, pertenciam ao grupo de discussão do Agile e são subscritores do manifesto, que atingiu uma certa notoriedade, sendo que hoje em dia é frequentemente utilizada em equipas de desenvolvimento de software.

Por muitos dos que debruçam sobre as questões do SCRUM, esta “não é uma metodologia nem um conjunto de práticas de engenharia” (Lacey, 2012). Segundo o mesmo autor, o SCRUM é antes “um *framework* leve desenhado para gerir o desenvolvimento de software e de um produto”, citando Schwaber & Sutherland, 2011 para o justificar.

Em termos formais, o SCRUM define papéis, fluxo das operações e ainda alguns instrumentos (Schwaber, 2014). Podem identificar-se facilmente os papeis no SCRUM (Lacey, 2012; Lacey, 2016) :

- **Product Owner:** Gere e representa os interesses das partes interessadas e dos clientes. Tem a missão principal de assegurar que a visão pedida pelos clientes está a ser executada pela equipa. Trabalha com as partes interessadas no sentido de compreender a funcionalidade do sistema e escrever ou colaborar com os mesmos no sentido de escrever *user stories*.
- **ScrumMaster:** Elemento que faz uma gestão informal da equipa. Deverá estar habilitado a ler comunicação não verbal, estar à vontade com o conflito sendo um comunicador efetivo, devendo construir confiança e ganhar respeito, entendendo a dinâmica da equipa. É um facilitador, no sentido em que deverá procurar soluções para as dificuldades encontradas pela equipa, mas não deverá integrar a equipa de desenvolvimento.
- **Development Team:** A equipa de desenvolvimento executa a visão do *Product Owner* com a ajuda do *ScrumMaster*. A equipa deverá ser multidisciplinar e integrar no seu seio os profissionais com as competências exigidas pela natureza do trabalho. Tem idealmente entre 5 a 9 elementos dedicados a tempo inteiro ao projeto. A equipa de desenvolvimento é responsável pela gestão do seu trabalho, pelos seus compromissos e pela sua execução.

O fluxo do SCRUM poderá ser descrito como um conjunto de fases descritas por Schwaber, 1997 como tendo o seguinte “grupo de fases:

1. Pregame

- a. Planeamento: Definição do novo lançamento baseado no *Backlog* (lista contendo funcionalidades desejadas, i.e., é uma lista de itens prioritizados a serem desenvolvidos para um software; ver mais abaixo mais detalhes sobre a definição) conhecido no momento, juntamente com uma estimativa da sua calendarização e do seu custo. Se estiver a ser desenvolvido um novo sistema, esta fase consiste na conceptualização do sistema e na análise. Se for apenas um melhoramento a um sistema existente, esta fase consiste numa análise limitada.
- b. Arquitetura: Projetar como os itens de *Backlog* serão implementados. Esta fase inclui modificações de arquitetura e projeto de alto nível.

2. Game

- a. *Sprints* de desenvolvimento (*sprint* é a unidade básica de desenvolvimento em Scrum, dura tipicamente entre uma semana e um mês; ver mais abaixo mais detalhes sobre a definição): Desenvolvimento de novas funcionalidades para lançamento, com respeito constante às variáveis de tempo, requisitos, qualidade, custo e concorrência. Interação com estas variáveis define o fim dessa fase. Existem múltiplos *sprints* de desenvolvimento iterativos, ou ciclos, que são utilizados para evoluir o sistema.

3. Postgame

- a. Fecho: Preparação para lançamento, incluindo a documentação final, testes de preparação de pré-lançamento e lançamento.”

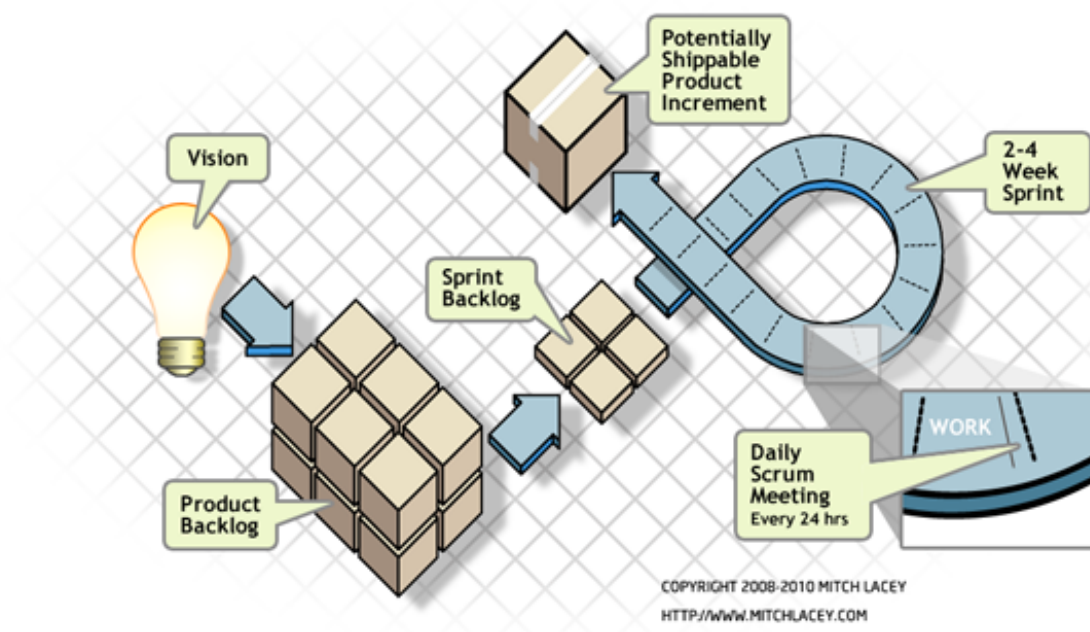


Figura 2.1- Fluxo do SCRUM (adaptado de Lacey, 2016).

Estas fases podem ser ilustradas da forma que se mostra na Figura 2.1. Nesta ilustração, Mitch Lacey ilustra sobretudo o que Ken Schwaber identifica como as fases de *Pregame* e de *Game*, identificando o caráter cíclico nesta última fase. A esses ciclos dá-se o nome de *Sprint*.

Em relação aos instrumentos, no SCRUM podem facilmente identificar-se (Lacey, 2012):

- **Product Backlog:** Contém uma lista ordenada pelo valor para o negócio e pelo risco, que contém todo o trabalho necessário para completar o projeto. Os elementos contidos nessa lista denominam-se *User Stories* e poderá também conter requisitos funcionais e não funcionais, erros e outros problemas. Cada *User Story* é estimada em unidades abstratas, denominadas *User Points*, num modelo de pesos relativos.
- **Sprint Backlog:** É o documento que resulta da reunião de planeamento do *Sprint*. Contém uma lista de tarefas que a equipa de Scrum necessita de realizar para completar um conjunto de itens selecionado do *Product Backlog* de forma a que possa terminar o *Sprint* com um determinado aumento do número de funcionalidades para um novo lançamento do produto. As tarefas presentes no *Sprint Backlog* devem ser estimadas em horas.
- **Sprint Burndown Chart:** O *Sprint Burndown Chart* é um gráfico que deverá ser preenchido diariamente e que representa as tarefas completadas do *Sprint*. Ele está orientado no sentido do trabalho em falta no *Sprint* e poderá ter representada uma linha reta desde o início até ao fim que servirá de guia para avaliar o estado de desenvolvimento a cada dia em relação ao expectável.

Em relação ao descrito sobre as características do SCRUM, há vários aspetos a destacar e a clarificar:

1. O *Product Owner* é o gestor do produto na companhia de desenvolvimento e trabalha com os clientes no sentido de construir o *Backlog* do produto, a partir de uma visão e de um pedido deste.
2. O *ScrumMaster* é o responsável pela equipa mas não toma decisões nem faz parte dela, age como um facilitador a ultrapassar todo o tipo de problemas e a sua missão é manter a equipa saudável e garantir que a equipa responde aos seus compromissos.
3. A Equipa de Desenvolvimento desenvolve ciclicamente os itens de funcionalidade descritos no *Product Backlog* em conjuntos determinados em períodos de tempo denominados *Sprints*. A Equipa de Desenvolvimento deverá ter 5 a 9 elementos, nem mais nem menos, sendo que 6 é o número ideal.

4. Cada *Sprint* é precedido de uma reunião demorada, denominada *Sprint Planning Meeting*, na qual se efetua o planeamento do *Sprint*. Elaboram-se as tarefas referentes aos itens do *Product Backlog* determinados, estimam-se e priorizam-se. É construído o *Sprint Burndown Chart* para preenchimento diário como forma de acompanhamento do estado de realização e qual o desvio para a previsão linear.
5. Um *Sprint* não deverá ter uma duração superior a 4 semanas que, segundo Lacey, 2012, nunca deverão ser estendidas. O mesmo autor defende que quando há problemas de duração de *Sprint* deverá mudar-se para um esquema de *Sprints* com duração de apenas uma semana.
6. Todos os dias no *Sprint* deverá ser realizada uma pequena reunião, em pé e de cerca de 15 minutos de duração, denominada *Daily Scrum Meeting*. É útil para cada elemento da equipa de desenvolvimento informar *ScrumMaster* e os restantes elementos da equipa do estado do seu trabalho e de quais as dificuldades encontradas. Tal permitirá que *ScrumMaster* as possa solucionar ou contribuir para ativamente a solução.

2.10. Estrutura de desenvolvimento de aplicações

Podem encarar-se as fases de desenvolvimento de uma aplicação como um longo *pipeline*, desde a escrita do código até à colocação da aplicação em produção. Ao considerar um processo cíclico em que se vão acrescentando funcionalidades à aplicação à medida que estas vão sendo desenvolvidas, a agilização desse *pipeline* torna-se importante para toda a equipa envolvida.

Há autores que mencionam a “Estrutura de desenvolvimento de aplicações tradicional em quatro camadas” (King, 2011), ainda que existam artigos que o mencionem como “boas práticas”³, apenas para referir alguns exemplos. No âmbito do presente trabalho optou-se por utilizar o termo “ambiente” para cada uma dessas camadas especificadas por se entender que é o termo universalmente aceite em português. Poderão então enumerar-se, por ordem de posicionamento no *pipeline*:

³ <http://guides.beanstalkapp.com/deployments/best-practices.html>

1. **Ambiente de desenvolvimento:** Ambiente onde os programadores desenvolvem código para novas funcionalidades, modificam algumas outras ou corrigem erros.
2. **Ambiente de integração:** Ambiente destinado à integração das peças de software escritas no ambiente de desenvolvimento.
3. **Ambiente de teste:** também referido como *stage environment* (King, 2011). Ambiente de pré-produção, onde são colocadas as versões prontas a disponibilizar em ambiente de produção para que se realizem os testes determinados pela equipa.
4. **Ambiente de produção:** Ambiente final de colocação em funcionamento da aplicação e disponibilizado aos utilizadores da mesma.

Usualmente, aos referidos ambientes é comum considerarem-se associados servidores, pelo que em algumas publicações se referem “servidores” e não “ambientes” como se optou por considerar no presente documento. Por exemplo, Zervaas (2007) considera três “servidores” para designar os ambientes que entende ser importantes para o desenvolvimento de aplicações web com PHP.

3. Grupo de trabalho e metodologias

3.1. Grupo de trabalho

Tal como mencionado anteriormente, o grupo de trabalho do SGIP foi criado por Despacho Reitoral RT.1/08 (ANEXO I) intitulado “Desenvolvimento do Sistema de Gestão de Informação Pedagógica”. No documento, que se anexa, está expressa a composição do grupo de trabalho:

- Pedro Laginha, analista e programador, Serviços de Informática;
- Luís Sousa, analista e programador, Serviços de Informática;
- Sérgio Baltazar, analista e programador, Faculdade de Ciências e Tecnologia;
- José Paulo Pinheiro, consultor, professor líder da equipa de projeto 2 e Diretor da Faculdade de Ciências e Tecnologia;
- Miguel Domingos, consultor para a temática de Gestão de Espaços.

O grupo de trabalho de analistas e programadores era assim, inicialmente, constituído por três elementos, apesar de não representar 300% da capacidade de trabalho de um profissional completamente dedicado ao projeto. Na realidade, apenas o autor do presente relatório estava inicialmente dedicado exclusivamente à aplicação. O Luís Sousa tinha de dividir o seu tempo com a gestão, suporte e formação na tutoria eletrónica, que se estimava em 50%, e o Sérgio Baltazar, cuja dedicação a “meio-tempo” se estimava em cerca de 80%, continuando a deter as responsabilidades de gestão informática na Faculdade de Ciências e Tecnologia (FCT).

Deverá ainda realçar-se que a equipa ficou reduzida a dois elementos no verão de 2008 uma vez que o Sérgio Baltazar teve de assumir os seus compromissos na FCT a tempo integral, passando a representar uma força de trabalho de apenas 150% de profissional completamente dedicado. A equipa ainda foi ligeiramente reforçada em dois períodos de tempo. Em primeiro lugar com um estágio muito curto do João Vieira, finalista do curso de Engenharia Informática e, mais tarde e também por um curto espaço de tempo, pelo colega dos Serviços de Informática Carlos Gomes. Ambas foram passagens esporádicas com pequenos mas valiosos contributos.

Nos próximos subcapítulos detalhar-se-ão as escolhas da equipa no que diz respeito a todas as opções que envolveriam o desenvolvimento da aplicação. Será efetuado, sempre que se julgar pertinente, uma referência em relação ao exposto no capítulo anterior.

Muito importante de referir ainda que o trabalho do SGIP que se apresenta neste documento é imputado ao autor, exceto nos casos em que for referido o contrário. Deverá ser ressaltado, no entanto, o facto de, por se tratar de um trabalho de equipa, todas as ações terem sido discutidas pela equipa.

3.2. Tipologia da aplicação

Tendo em conta o explanado no item anterior no que diz respeito à necessidade de partilha e integração da aplicação nos sistemas existentes na UAlg, sabendo ainda que a tipologia de aplicação que permite uma gestão centralizada mais eficiente é a típica *client-server* e ainda que todos os elementos da equipa de desenvolvimento possuíam experiência no desenvolvimento de aplicações web, esta tipologia foi a opção natural para o SGIP.

A utilização deste tipo de tipologia de aplicação obrigou a que se elegeesse um *browser* de referência que acabou por recair no Firefox essencialmente pelas seguintes características:

- Existir versão para vários sistemas operativos, designadamente para os 3 mais utilizados na altura: Microsoft Windows, Mac OS X e Linux;
- Ser o que dispunha de um mais vasto conjunto de ferramentas de apoio ao desenvolvimento;
- Ser oriundo do mundo *open-source*, logo não ter quaisquer custos para a Universidade;
- Ser o que, declaradamente, seguia mais à risca os *standards* na altura.

3.3. Servidor web

Tendo sido decidido desenvolver uma aplicação web, tal como descrito anteriormente, tornou-se necessário decidir quais os suportes lógicos sobre a qual a mesma assentaria. Para tal, dada a escassez de recursos existente na altura, foi decidido utilizar o mesmo servidor que já estava a ser utilizado no servidor do portal da Universidade. Esse servidor, assente em tecnologia Microsoft, utilizava o *Internet Information Server* (IIS) como servidor de HTTP. Este facto não tornaria as escolhas da equipa mais limitadas pela enorme capacidade de integração deste servidor com as mais variadas tecnologias.

Contudo, uma mudança de paradigma em relação ao portal institucional foi introduzida mais tarde, em 2013. A nova plataforma utilizada uma base Linux e assentava no *Apache Httpd Server*. A partir dessa altura, foi migrada a aplicação para a nova plataforma, passando a estar integrada neste ambiente. Facilmente se constatou que as escolhas realizadas ao nível da linguagem e da base da aplicação foram acertadas uma vez que a migração foi extremamente rápida.

3.4. Linguagem e framework

Da necessidade de escolha da linguagem de base para a aplicação, decidiu-se utilizar o PHP. Tal decisão deveu-se a vários fatores:

- Por ser a linguagem web na qual o maior número de elementos da equipa de desenvolvimento tinha mais experiência;
- Por ser facilmente configurável no servidor da UALG;
- Por deter uma comunidade muito grande e ativa;
- Por ser uma linguagem com um elevado nível de maturação.

Mais difícil e morosa foi a discussão em torno da utilização de *frameworks* que auxiliassem e acelerassem o desenvolvimento. Sendo que os ganhos da sua utilização são frequentemente conhecidos e divulgados, a existência de uma curva de aprendizagem poderia inviabilizar a sua adoção.

Seja como for, a utilização de uma *framework* deveria ter sempre em consideração a aceleração efetiva que daria ao desenvolvimento, fornecendo meios de desenvolvimento rápido das tarefas mais morosas, usualmente formulários, apresentação e integração com a base de dados. Adicionalmente, a *framework* deveria estar em desenvolvimento pleno de forma a garantir que alguns eventuais erros e limitações fossem ultrapassados e possuir ainda uma comunidade que possibilitasse a colocação de questões com probabilidade de ser respondidas.

Tendo em conta a escolha da linguagem e tudo o explanado sobre este assunto no capítulo anterior, foram colocadas duas hipóteses: o *CakePHP* (CakePHP, 2016) e o *Symfony* (Symfony, 2016). A escolha acabou por recair no *Symfony* pelas seguintes razões:

- Pela possibilidade de ter uma pequena formação *on-site*;
- Por possuir uma comunidade grande, ativa e a crescer;
- Por existir alguma boa documentação de uma forma geral:
 - Livro do mentor da *framework* (Zaninotto & Potencier, 2007);
 - Um *how-to* para ir construindo um site de perguntas e respostas denominado “Askeet”, acessível na parte *legacy* do site do Symfony (Symfony, 2016);
 - Um conjunto de *Cheat Sheets* disponíveis na mesma área;
 - Muitos textos e artigos sobre temas variados.

Poder-se-á afirmar que, visto à distância dos anos que, entretanto, passaram e com os conhecimentos e as escolhas que existiam na altura, parece uma decisão acertada. Por um lado, o *Symfony* mantém-se extremamente ativo, com uma grande comunidade, muita documentação e com uma continuidade de grande desenvolvimento. Por outro lado, assegurou conhecimento de uma plataforma que seria utilizada pela equipa em mais ocasiões, designadamente na aplicação de Perceções de Ensino-Aprendizagem para Alunos e Docentes (PEAAD) e de Matrículas de Alunos nos Serviços Académicos (MASA) que realizava a recolha de dados dos alunos no ato de matrícula nos Serviços Académicos. Tais dados eram posteriormente integrados no servidor de gestão académica.

3.5. Servidor de base de dados

O motor de base de dados é, quase sempre, um componente importante de uma aplicação no sentido em que o desempenho da mesma poderá ser comprometido pela escolha realizada. Tal como aconteceu com a escolha da linguagem e da *framework* a adotar, a escolha do Sistema de Gestão de Base de Dados (SGBD) a utilizar deveria estar integrado nos padrões das escolhas da UAlg, ser conhecido pela equipa de desenvolvimento e ainda ser devidamente compatível com as restantes escolhas de servidor, linguagem e *framework*.

Tal como descrito no capítulo anterior, colocavam-se algumas hipóteses para escolha do motor de base de dados. A escolha acabou por recair no *MySQL* pelas seguintes razões:

- Tinha uma ferramenta de gestão, o *MySQL Workbench* (MySQLWB, 2016), que, apesar de estar num estado ainda pouco maduro, era mais evoluída;
- Começava a ter uma comunidade grande e muito ativa;
- Tinha bastante documentação;
- Começava a assumir-se como um *standard no open-source*.

3.6. Editor de desenvolvimento integrado (IDE)

Dada a importância de que se revestia a utilização de um ambiente integrado, a equipa de desenvolvimento do SGIP procurou os IDE mais completos e que melhor serviriam os interesses de desenvolvimento da aplicação, tendo em conta as características que estão expostas no presente relatório, mas ainda que respeitassem uma característica importante: ser de código aberto. O resultado dessa pesquisa encontra-se descrito no capítulo anterior.

Tendo em conta que, tal como apresentado, a escolha seria realizada entre dois IDE, o *Eclipse* e o *Netbeans*, a equipa optou pelo primeiro. Não existiu uma justificação “científica” para tal escolha, apenas a ideia que este detinha uma comunidade maior e uma maior dinâmica.

Assim, o *Eclipse* começou por ser inicialmente utilizado, mas tal escolha durou apenas os primeiros meses de desenvolvimento. Entretanto, o editor presente no IDE revelou-se pouco fiável quando se editavam ficheiros de média dimensão, com mais de 2000 linhas de código. Esta falta de fiabilidade revelava-se facilmente quando começava a “sujar” o écran, ou seja, quando se deixava de perceber o que se estava a editar e o que aparecia no editor já não corresponderia exatamente ao que se obtinha ao gravar o ficheiro.

Desta forma, o *Netbeans* acabou por ser o IDE escolhido para o desenvolvimento do SGIP ao longo do restante tempo de desenvolvimento da aplicação e ainda dos *web services* de exploração da respetiva informação. Este IDE tinha um suporte de edição de código *javacript* mais evoluído e não apresentou os problemas de fiabilidade do Eclipse quando editando ficheiros de maior dimensão.

3.7. Ferramentas de colaboração e gestão de projeto

Quando foi lançada, percebia-se que o SGIP seria uma ferramenta estruturante para a Universidade. Dada a importância de que se revestia, foram destacados para o projeto 3 programadores seniores que, apesar de tudo, não poderiam dedicar o seu tempo integral ao projeto. Adicionalmente, foi atribuída a sua coordenação aos Serviços de Informática, sendo que a supervisão era assegurada pelo responsável da Equipa de Projeto 2 que deu origem ao grupo de trabalho, como foi mencionado anteriormente.

Contudo, sendo uma equipa com alguma dimensão à escala do que a Universidade do Algarve podia despender, contudo uma equipa pequena quando comparada com qualquer outra equipa de desenvolvimento, existiu a necessidade de utilizar uma ferramenta de planeamento que não fosse demasiado complexa e cujo tempo de aprendizagem fosse reduzido. Tendo em conta que o servidor de desenvolvimento e a maior parte dos ambientes de trabalho dos programadores utilizavam o *Debian Linux*, ou nele baseado, e o GUI *Gnome*, decidiu-se utilizar o *Gnome Planner*. Contudo, esta necessidade surgiu num período mais avançado do desenvolvimento, quando a aplicação passou a produção e se começaram a acumular tarefas de suporte a adicionar às tarefas de desenvolvimento de novas características na aplicação. Durante esse

período, a pequena equipa ainda teve que desenvolver outros projetos como os já citados PEAAD e MASA.

O *Gnome Planner* foi a aplicação selecionada porque:

- É extremamente simples de utilizar;
- Detém todas as características que se pretendiam tais como a gestão de tarefas e sub-tarefas e respetiva duração, a gestão de recursos humanos e respetivo horário e plano anual de trabalho, a atribuição ponderada das tarefas e a apresentação em gráfico de Gantt, entre outras;
- Era de acesso fácil uma vez que os programadores utilizavam Linux com o ambiente gráfico *Gnome*.

Pode dizer-se que se revelou mesmo uma ferramenta muito importante na gestão do projeto pela facilidade com que permitiu gerir o tempo de desenvolvimento e pela contribuição para as opções de desenvolvimento tomadas.

Por outro lado, uma das ferramentas mais importantes no desenvolvimento de uma aplicação é a de documentação e gestão de tarefas. Esta aplicação permite que se documente de uma forma fácil e rápida todos os aspetos que dizem respeito à aplicação mas também criar registos de tarefas a executar, bem como toda a gestão da sua atribuição, estado e calendarização de tarefas, características de prioridade e muitas outras importantes na gestão do desenvolvimento.

Tal como mencionado anteriormente, na escolha da aplicação, talvez por falta de experiência na utilização de tais ferramentas e pela pouca disponibilidade de tempo dos elementos da equipa, não foi feita uma procura exaustiva. Assim, a aplicação escolhida foi o TRAC (Trac, 2016) pelas seguintes razões:

- Por ser a ferramenta utilizada pela comunidade para documentar e gerir o *Symfony*;
- Por se tratar de uma aplicação extremamente fácil de utilizar;
- Por permitir controlo de tarefas e tickets, e ter uma ferramenta de wiki.

A partir de uma determinada altura, quando se pretendeu documentar outros projetos, emergiu a limitação maior do TRAC: ser mono-projeto. Ou seja, o TRAC não detém forma

de albergar, nativamente e de uma forma clara, vários projetos, percebendo-se que foi pensado para apenas um projeto. Dada a limitação, mais tarde foi escolhido o *Redmine* para os restantes projetos.

3.8. Sistema de controlo de versão

Hoje em dia é impensável a gestão de um projeto de desenvolvimento sem a utilização de uma ferramenta de controlo de versões. Enquadrando a necessidade no devido tempo, isto é, no início do projeto em princípios de 2008, a decisão não era evidente, em parte por existir alguma inexperiência na utilização deste tipo de ferramentas.

Remetendo ao apresentado no capítulo anterior, pareceu importante a utilização de um sistema de controlo de versões distribuído. Desta forma, optou-se pelo GIT (GIT, 2016) em grande parte pelas seguintes características:

- Tratar-se de um sistema distribuído;
- Permitir a sua utilização em modo *offline*;
- Ter uma forte garantia de continuidade na aposta por ser a ferramenta de controlo de versão utilizada na gestão de versões do *kernel* do Linux.

O tempo acabou por revelar que a escolha terá sido a melhor. Embora hajam muitas vozes críticas em relação à qualidade e características do GIT, pode dizer-se que tem uma enorme projeção e uma comunidade grande de utilizadores. Para além disso, é uma ferramenta utilizada no seio da equipa hoje em dia de forma amiúde.

3.9. Metodologia

Usualmente os projetos que abrangem de uma forma transversal toda a organização não são projetos simples por existirem tantas formas de pensar, de desenvolver estratégias, de organizar e de realizar quantas as unidades orgânicas que compõem a UALG. Por muito completa que pareça a análise, por muito lógico que pareça determinado processo, por muito boa vontade que todos os envolvidos possam ter, há sempre um

determinado requisito que falha ou uma característica essencial a determinado departamento ou unidade orgânica.

Por ter esta perceção, a equipa determinou que seria necessária uma metodologia que se afastasse das metodologias tradicionais, em que fosse muito fácil introduzir alterações aos requisitos. Claramente e de acordo com o explicado no capítulo anterior era necessário utilizar uma abordagem Agile, pelo que se optou por escolher o SCRUM (Schawber, 1997).

Contudo, o SCRUM tal como descrito pela documentação tornava-se difícil de implementar por algumas das características descritas anteriormente sobre a metodologia. De uma forma rápida, a equipa não tinha dimensão para aplicar o SCRUM com todas as suas características, uma vez que era composta por apenas 3 elementos e nem sequer todos dedicavam 100% do seu tempo ao projeto. Outra das razões que não possibilitavam a aplicação do SCRUM, tal como este é entendido, é o facto de não existirem pessoas com as características adequadas para assumir alguns dos papéis da metodologia, designadamente de *Product Owner* e de *ScrumMaster*.

Assim, foram efetuadas as seguintes adaptações:

- O *Product Backlog* foi elaborado pelo colega de equipa Luís Sousa a partir dos procedimentos aprovados no seio da Equipa de Projeto 2 – equipa à qual pertencia – foi priorizado, mas não foi praticamente ponderado. A título de exemplo a Fig. 2 apresenta um pequeno extrato do *Product Backlog*, podendo ser consultado em detalhe no ANEXO IV.
- A figura do *Product Owner* foi apenas o detentor da visão, não existindo o trabalho importante desse papel, embora mais tarde se tivessem realizado algumas reuniões com os “clientes” através de uma criada comissão de acompanhamento para o efeito.
- A figura do *ScrumMaster* foi assegurada pelo coordenador dos Serviços de Informática que acumulava demasiadas tarefas mais, ao contrário do que aconselha a metodologia em que o papel deverá ser para um elemento a 100%.
- As reuniões diárias, os *Daily Scrum Meetings*, passaram a ser bi-semanais, às terças e às quintas-feiras e imediatamente antes da hora de almoço.

- Os *Sprints* excederam, a maior parte das vezes, as 4 semanas máximas aconselhadas, eventualmente por inexperiência da equipa, ao contrário que aconselha o SCRUM que defende o término neste período ainda que nem todas as funcionalidades tenham sido desenvolvidas.
- As tarefas eram estimadas em dias de trabalho e não em horas, que funcionou bem porque permitia que a estimativa não fosse demasiado otimista.
- Na maior parte das vezes, as tarefas não eram formalmente priorizadas e o *Sprint Backlog* era uma simples lista de tarefas, com prioridade informal, que os programadores tomavam aleatoriamente para desenvolver.

Product Backlog								
ID	ID Equipa	Nome	Importância	Estimativa	Como o demonstrar	Notas	Sprint	Término
17		Estrutura de permissões	10000		O utilizador faz login e em função do seu papel, pode: adicionar novos utilizadores, delegar permissões, e outras específicas consoante os papéis que lhe estão atribuídos		1	07-04-08
14	SEP-2.0-H-00	Serviço docente	9900			Meta user story. Filhos: 16, 18, 19, 20 e 32	2	
16		Importação de dados para o serviço docente	9800		Deve ser importado e disponibilizado na base de dados de importação o seguinte: planos de estudos, disciplinas, alunos inscritos, e docentes associados às disciplinas		2	
32		Preparação de dados para o Serviço Docente	9700		Os dados importados devem ser transformados para a estrutura de dados. Ao entrar, o utilizador em função das suas permissões, pode gerir: área científica do docente, calendário escolar, CORDIS/Área Científica FCT, planos de estudo, disciplina e disciplinas agregadas		2	
18		Proposta dos responsáveis científicos de cada disciplina	9600		O responsável de cada área científica ao fazer o login pode associar, alterar e remover os docentes de cada disciplina. A criação de uma pré-versão bloqueia todas as alterações e informa o Conselho Científico	Criação de um sistema de versões para o serviço docente	2	

Figura 3.1 - Extrato do Product Backlog.

Por outro lado, houve alguns princípios de base que a equipa não prescindiu:

- Os *Sprints* foram sempre precedidos de um *Sprint Planning Meeting*, no qual eram sempre debatidos todos os assuntos e elaborado o *Sprint Backlog*, composto por todas as tarefas planeadas.
- O final de um *Sprint* deu sempre origem a uma nova versão do produto, fosse ela uma versão maior ou versão menor.
- Quase sempre foi preenchido um *Burndown Chart*, cujo exemplo se apresenta na Figura 3.2, para perceber o andamento real do *Sprint* em relação ao estimado.
- As decisões de implementação eram decididas pela equipa.

No final, apesar de haver a consciência que a aproximação realizada pela equipa não se poderá chamar de SCRUM, poder-se-á afirmar que foi uma metodologia que resultou.

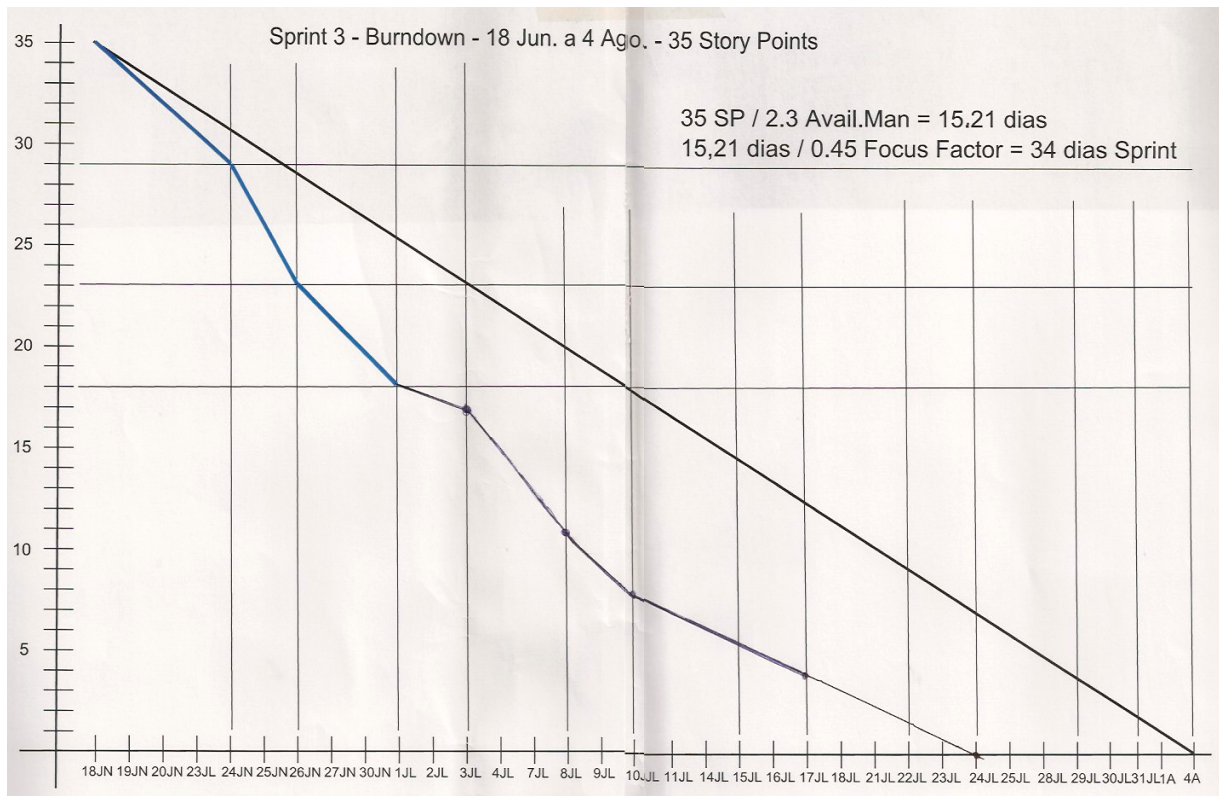


Figura 3.2 – Exemplo de Burndown Chart.

3.10. Estrutura de desenvolvimento da aplicação

Como visto no capítulo anterior, a definição de uma estrutura de desenvolvimento da aplicação é importante no sentido de agilizar o processo completo, desde o desenvolvimento até à colocação de novas versões em produção. Ou seja, esta estrutura deverá ser afinada como uma verdadeira linha de montagem de forma a tornar o processo célere e de fácil realização.

Para a estrutura do SGIP foram definidos 3 ambientes em linha com Zervaas (2007). Assim, optou-se por considerar que, dada a escassez de recursos humanos existentes, o ambiente de integração não existiria, sendo que as tarefas de integração eram realizadas por cada programador no seu ambiente de desenvolvimento. Em suma, foram implementados os seguintes ambientes:

- **Ambiente de desenvolvimento:** em cada computador dos programadores.
- **Ambiente de testes:** no servidor de desenvolvimento e de produção.
- **Ambiente de produção:** no servidor de produção.

Não se determinou que o ambiente de desenvolvimento fosse uniforme. Antes pelo contrário, tendo em conta a heterogeneidade de experiências dos elementos da equipa, o ambiente deveria ser o mais adaptável possível para evitar que se perca eficiência ou desempenho dos seus elementos. No fundo, cada um dos elementos da equipa seria responsável por instalar e configurar o ambiente nos seus computadores pessoais, de forma a responder ao escolhido por todos como *framework*, linguagem de programação, sistema de base de dados de apoio, sistema de controlo de versão, e tudo o demais necessário. Basicamente, cada programador optou pelo sistema operativo e por todo o software de apoio em que se sentia mais confortável, sendo que as opções apresentadas anteriormente eram obrigatoriamente verificadas.

Por ambiente de teste entendia a equipa que fosse um ambiente controlável e o mais semelhante possível com o sistema em produção. Dessa forma, conseguir-se-iam antecipar os problemas e limitações, prevendo-se que tal conduziria a um menor tempo de interrupção de serviço no ambiente de produção. Por razões operacionais e pela necessidade de um controlo efetivo por parte da equipa de desenvolvimento, utilizou-se o servidor de bases de dados presente na máquina de desenvolvimento da equipa para o ambiente de testes e o servidor *web* de produção como servidor de testes da aplicação.

Para a instalação e configuração do ambiente de produção foram identificadas as principais preocupações que, de uma forma genérica, se podem descrever como:

- **Partilha:** dada a escassez de recursos informáticos, o sistema deveria, o mais possível, partilhar os recursos utilizados com os restantes sistemas da UAlg.
- **Integração:** o sistema a desenvolver deveria ser o mais integrado possível com os restantes sistemas utilizados na UAlg.
- **Baixo custo:** a aplicação e todas as ferramentas de desenvolvimento e exploração deveriam basear-se o mais possível em tecnologias *open-source* de forma a onerar a instituição o menos possível.
- **Controlo de acesso:** o sistema apenas deveria estar acessível a utilizadores autorizados e a configurar internamente.

- **Responsividade:** o sistema deveria ser responsivo ao utilizador de forma que este pudesse ter uma boa experiência de utilização que não se tornasse frustrante.

A estrutura de desenvolvimento da aplicação apresentada foi replicada, com bastante sucesso, para desenvolvimentos posteriores por parte da equipa de desenvolvimento dos Serviços de Informática.

4. Sistema de Gestão de Informação Pedagógica

4.1. Dados externos de base

4.1.1. Introdução

A informação de base utilizada no Sistema de Gestão de Informação Pedagógica (SGIP) não é produzida pela própria aplicação, é carregada na aplicação. Essa informação tem duas fontes de dados importantes: a aplicação de gestão académica SOPHIA e o *Active Directory* (Desmond et al., 2013) da instituição, doravante designado simplesmente por AD. A essas fontes importantes deverá acrescentar-se o calendário escolar, diferente todos os anos, e que é manualmente lançado na aplicação.

Para disponibilizar a informação de base recorreu-se a um processo de importação e outro de transformação dos dados, embora numa fase inicial fossem apenas um só processo. As duas tarefas que compunham o processo de importação e transformação foram separadas devido à extrema lentidão do processo de importação através dos *web services* do SOPHIA. Contudo, deve referir-se que, um par de anos mais tarde, o servidor do SOPHIA foi atualizado tendo diminuído consideravelmente os tempos de importação.

De forma a materializar os processos que permitem a existência da informação de base no SGIP, foram escritos dois conjuntos de aplicações em linha de comandos (CLI⁴): um para executar a importação e outro para garantir a respetiva transformação de dados.

A conceção e a escrita destes procedimentos, à exceção das interfaces de gestão de dados e das subseqüentes intervenções no modelo e algoritmo, tiveram apenas um pequeno ou médio contributo do autor do documento, pelo que não se apresentarão com detalhe. Contudo, é importante esclarecer alguns pormenores dos processos por se considerarem importantes para a compreensão futura da aplicação e do documento.

⁴ Em inglês designa-se por *Command Line Interface* ou simplesmente por CLI.

4.1.2. Importação

Como já foi referido, o SGIP recorre a duas fontes de dados para obter os seus dados de base: a aplicação de gestão académica SOPHIA e o *Active Directory* (AD) da UAlg. Quase toda a informação tem origem no SOPHIA exceto os docentes pois apenas uma pequena percentagem destes está presente na aplicação. Os registos de docentes que existiam são apenas os correspondentes a docentes que detinham algum nível de responsabilidade, por exemplo, responsáveis de disciplinas ou responsáveis de curso.

Deve ter-se sempre presente que a informação académica fornecida pelos *web services* estava organizada de forma hierárquica. Nessa hierarquia, o topo é a unidade orgânica, como entidade responsável pela oferta de cursos com planos de estudo que eram compostos por disciplinas que existiam de forma única no plano de estudos. A esta informação juntava-se os alunos e as suas inscrições às disciplinas e ainda os registos de docentes.

Os dados de importação são registados em tabelas de importação correspondentes na base de dados, que, de alguma forma, replicam a estrutura hierárquica citada. Esta decisão permitiu que o processo de transformação passasse a executar com um melhor desempenho.

4.1.3. Transformação

O processo de transformação é autónomo do processo de importação, embora o seu agendamento deva ser definido para ocorrer imediatamente após o término da importação, conseguindo assim demorar o menor tempo na operação global de importação e transformação e também concentrá-la fora das horas de expediente. Desta forma consegue-se um menor impacto para os utilizadores da aplicação.

Tal como o processo de importação e como já foi referido, a transformação também se realiza por aplicações CLI. A transformação é substancialmente mais complexa que a importação pois contém toda a componente decisória dos campos a importar, com o seu mapeamento para a base de dados e respetivas operações de conversão, junção,

separação, desdobramento em outras tabelas ou outras que se consideraram necessárias.

Um ponto muito importante que deverá ser ressaltado é que essa transformação é ainda mais complexa pelo simples facto de determinados utilizadores com determinados perfis terem capacidade para criar planos de estudos e disciplinas no SGIP. Historicamente, a necessidade de criação de planos de estudos ou disciplinas no SGIP, por parte dos gestores de planos e disciplinas, prende-se com a inexistência de um calendário único que respeitasse o lançamento dos novos planos de estudos e respetivas disciplinas no SOPHIA e ainda com o facto do serviço docente necessitar desses planos e disciplinas para afetar os turnos das disciplinas aos docentes. A inexistência desse calendário único para o lançamento dos planos de estudos e respetivas disciplinas impossibilitava muitas unidades orgânicas de completar o lançamento do serviço docente numa data que permitisse, entre outros, a identificação de eventuais necessidades de contratação.

Contudo, desde início de 2012, com vista à preparação do ano letivo de 2012/2013, houve uma tentativa de normalização das datas limite para as várias tarefas através do Despacho Reitoral RT.008/2012, de 23 de janeiro de 2012 (ANEXO V). Esta nova realidade teve algumas dificuldades no cumprimento do calendário definido, embora se reconheça o esforço de toda a comunidade na sua prossecução.

O SGIP poderá, assim, ter planos de estudos ou disciplinas importados ou criados na aplicação. Para permitir a coerência da informação e de forma a respeitar os lançamentos dos operadores da aplicação, foram criadas tabelas de sincronização para planos e disciplinas. Entende-se como sincronização o encontro de um registo de importação com o respetivo existente no SGIP.

O algoritmo seguido não foi muito complexo. De uma forma geral, seguiram-se os seguintes princípios:

1. Os dados de base de cada ano letivo respeitantes a planos de estudos e disciplinas é uma cópia dos do ano anterior que se encontrem ativos, num procedimento que será abordado mais à frente na Secção 4.2.11 - *Abertura do ano letivo*
2. *Procedimento* de abertura do ano letivo.

3. A transformação de registos importados de unidades orgânicas, docentes, alunos e inscrições de alunos é concretizada e concluída no mesmo procedimento, sendo que:
 - a. Se o registo já existir na tabela correspondente é ignorado;
 - b. Se o registo não existir na tabela correspondente é criado.
4. A transformação de registos de cursos e disciplinas é realizada para tabelas próprias de transformação, aguardando o procedimento seguinte de sincronização, seguindo os mesmos princípios de criação de registo indicados em 2.

De notar que os cursos e as disciplinas são conhecidos na UAlg pelo seu código dos serviços académicos, usualmente designado pelos profissionais da universidade como *cod_acad*. Se um registo for criado no SGIP e não for feita a sincronização com outro registo válido proveniente da gestão académica, ele não será válido externamente. Esse facto é facilmente comprovado pela inexistência de *cod_acad*, uma vez que o SGIP apenas os designa por '0000' internamente.

4.1.4. Sincronização de cursos e disciplinas

A sincronização de cursos e disciplinas requiere um procedimento adicional como foi referido anteriormente. Esse procedimento está relacionado com a existência ou não dos registos nas tabelas definitivas e ainda pela utilização de uma interface de gestão de dados para o efeito e disponibilizada aos Gestores de Planos e Disciplinas.

A execução de tal procedimento acontece imediatamente a seguir à execução do procedimento de transformação. Na realidade, faz parte dele, sendo que apenas do ponto de vista de conceito existe de forma separada. Assim, considerando também a operação na gestão de dados, o algoritmo de sincronização de cursos e disciplinas poder-se-á definir da seguinte forma:

1. Sequencialmente para cursos e disciplinas, os registos na tabela de sincronização são marcados como:
 - a. **Resolvido**, caso o registo transformado já exista nas tabelas definitivas de cursos ou disciplinas da base de dados;

- b. ***Não resolvido***, caso o registo ainda não exista na base de dados.
- 2. Sob ação de um utilizador na gestão de dados, os registos de cursos e de disciplinas que estiverem sinalizados como ***Não resolvido***:
 - a. Poderão ser associados a cursos ou disciplinas já existentes na base de dados e são sinalizados como ***Associado***;
 - b. Poderão ser criados nas tabelas de produção, sendo sinalizados como ***Criado***.

As interfaces de gestão de dados foram completamente desenvolvidas pelo autor do documento. Por se tratar de um componente da aplicação e por utilizar alguns conceitos ainda não apresentados, as interfaces serão apresentadas mais à frente no presente documento.

4.2. Aplicação e conceitos

4.2.1. Introdução

No presente subcapítulo apresentam-se os detalhes da aplicação tal como foi desenvolvida e utilizada no seio da organização. Deverá ter-se em conta que não serão abordadas todas as interfaces uma vez que tal conduziria a um relatório demasiado extenso. Por exemplo, foi regra no desenvolvimento do SGIP que todas as entidades de base de dados (tabelas) teriam uma interface CRUD, o que resultou numa grande quantidade de interfaces.

Ao invés, optou-se por apresentar as interfaces mais exemplificativas e singulares, mas também, sempre que possível, aquelas em que o autor teve o seu papel dominante na conceção e desenvolvimento. Nesse sentido, também se optou por apresentar os conceitos e algoritmos utilizados no SGIP que permitam uma boa compreensão do funcionamento da aplicação, sobretudo na sua tarefa de gestão de serviço docente.

4.2.2. Controlo de utilizadores

4.2.2.1. Acesso ao SGIP

Desde que entrou em produção, o SGIP esteve sempre disponível no seu endereço definido inicialmente *http://sgip.ualg.pt*. A página principal, que se ilustra na Figura 4.1, convida o utilizador a autenticar-se, sendo que as credenciais a utilizar serão as suas do domínio UALG. Como o SGIP não tem tabela de utilizadores, qualquer utilizador do domínio UALG poderá aceder à aplicação, mas, se não possuir as permissões necessárias, não conseguirá realizar qualquer operação. Estas permissões serão detalhadas mais à frente no presente documento.



Figura 4.1 - Página de entrada do SGIP.

Os utilizadores acedem ao SGIP através da sua janela de *login* onde deverão autenticar-se com as suas credenciais da UALG. Sempre que é solicitado um acesso a um recurso da aplicação e não existe sessão criada, o utilizador é enviado para a janela de *login* de forma a poder criar uma sessão.

A autenticação é realizada nos servidores controladores de domínio por LDAPS. Não serão fornecidos detalhes desta operação por não ter sido desenvolvida pelo autor deste documento.

Uma vez autenticado, o utilizador é remetido para a página principal autenticada. Esta página está ilustrada na Figura 4.2 e deverá observar-se que o utilizador autenticado tem um perfil de gestor da aplicação, ou seja, tem acesso a toda a barra de menus do lado esquerdo da página. Qualquer utilizador sem privilégios terá acesso apenas à caixa de mudança de ano lectivo corrente e à caixa das opções das distribuições de serviço docente.

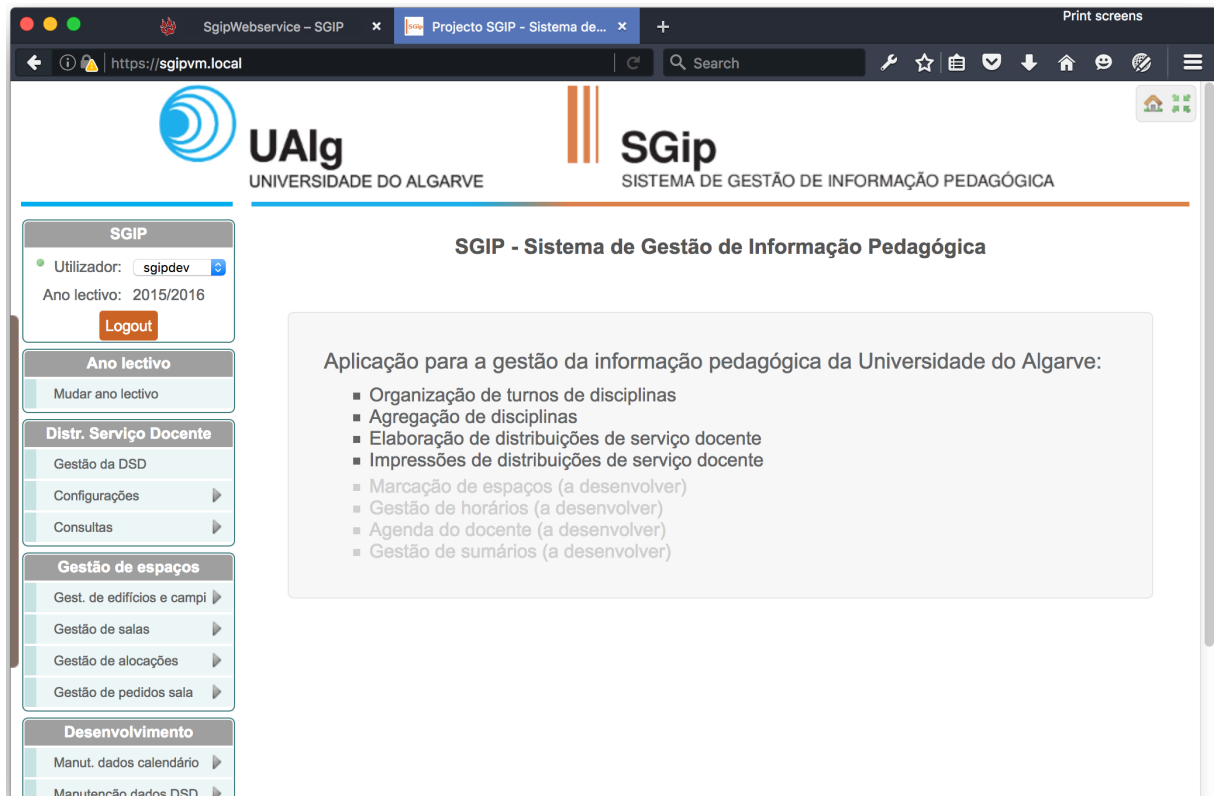


Figura 4.2 - Página principal para um gestor do SGIP.

4.2.2.2. Sistema de permissões

O sistema de permissões foi pensado como tendo as seguintes entidades:

- **Direito:** capacidade de executar determinada ação em particular;
- **Papel:** conjunto de direitos;
- **Âmbito:** enquadramento da operação;
- **Atribuição:** capacidade atribuída a um utilizador através de um papel num determinado âmbito, num determinado período de tempo e respeitante a determinados anos lectivos.

Os direitos nascem naturalmente das operações que são disponibilizadas aos utilizadores do SGIP. Simbolizam a permissão que é dada de acesso à operação não definindo, contudo, um enquadramento. Considerando o CRUD de uma tabela, foi usual definir-se um direito de LEITURA para a consulta dos registos e outro para EDIÇÃO dos registos. Foram raras as situações em que foi necessário um maior refinamento dos direitos.

Os papéis, por seu lado, foram definidos para, em conjunto, serem facilmente associados a um perfil. Ou seja, o papel, ao ser um conjunto de direitos, representa um conjunto de capacidades que fazem sentido atribuir a um utilizador conjuntamente. Por exemplo, o papel de gestor de planos e disciplinas deverá ter, pelo menos, os direitos de leitura de cursos, edição de cursos, leitura de disciplinas e edição de disciplinas, com o propósito de poder ler, criar, alterar ou apagar cursos e disciplinas.

Enquanto que o direito é a forma de permitir, ou não, que determinado utilizador execute determinada operação, o âmbito é a entidade responsável por limitar o acesso a essa informação. Dessa forma, se garante que determinado utilizador apenas tem acesso à informação que é suposto ter. No SGIP apenas foram implementados âmbitos à entidade, que representa uma unidade orgânica, um departamento, um núcleo, um serviço ou qualquer outra que faça sentido considerar. Por exemplo, um elaborador de distribuições de serviço docente responsável pelas DSD da unidade orgânica A apenas tem acesso às DSD dessa unidade orgânica. Ou, caso pretenda, poderá, como responsável de um determinado departamento de uma determinada unidade orgânica, ter acesso apenas às DSD desse departamento.

Poder-se-ão considerar as entidades apresentadas como as entidades de base do sistema de permissões. De forma a poder combiná-las, definir-lhes limites temporais e atribuí-las aos utilizadores, considerou-se uma outra entidade que se designou de Atribuição. A atribuição, cuja definição já foi apresentada acima, serve para atribuir um perfil a um determinado utilizador, filtrar de forma adequada o acesso à informação e limitar essa capacidade no tempo, sendo que esse tempo poderá ser expresso em anos letivos e em data de início e data de fim. Pode, por exemplo, pretender-se que o diretor da unidade orgânica apenas possa homologar DSD nos anos letivos em que foi diretor, entre a data em que tomou posse e a data em que foi substituído.

A Figura 4.3 representa um exemplo de uma atribuição com todas as propriedades referidas. No exemplo, é mostrado um utilizador homologador de DSD, com capacidade para homologar serviço docente de uma unidade orgânica com a sigla ISE, a partir de 1 de abril de 2014 e apenas para o ano letivo de 2014/2015 e subsequentes.

The screenshot displays the 'Atribuições - Edição de registo' page in the SGIP system. The page is divided into several sections:

- Identificação:** Papel: Homologador de DSD
- Utilizadores:** Utilizador: uhomologador, Validar utilizador
- Delegação:** Delegação: uhomologador, Validar utilizador, Igual ao username
- Âmbitos:** Entidades: ISE, Parâmetro: ISE, Acções: Novo registo
- Efectivação:** Data de início: 2014-04-01, Data de fim: 20142015, Ano lectivo início: 20142015, Ano lectivo fim: 0
- Complementares:** Observações: Homologador de DSD de teste para Mestrado.

The sidebar on the left contains navigation menus for 'SGIP', 'Ano lectivo', 'Distr. Serviço Docente', 'Gestão de espaços', and 'Desenvolvimento'.

Figura 4.3 - Edição de atribuição.

4.2.3. Grupos de docentes

Os grupos de docentes são as entidades dentro do SGIP sobre as quais se elaboram serviço docente. Considerando a heterogeneidade de organização interna de cada unidade orgânica da Universidade e ainda as diferentes lógicas de elaboração de serviço docente das mesmas, um grupo de docentes poderá ser qualquer conjunto de docentes que se pretenda. Desta forma, existiram unidades orgânicas que espelharam nos grupos de docentes os docentes que pertencem a um determinado subgrupo da unidade e existiram outras unidades orgânicas que optaram por criar os grupos de docentes relativos aos cursos a que elaboram o serviço docente.

Os grupos de docentes são entidades versionáveis, criadas pelos elaboradores de serviço docente de forma a que fosse possível representar o grupo exato em determinada DSD em particular, e ainda que compreendem um ou mais docentes. Um docente poderá pertencer a tantos grupos de docentes quantos se pretender, tornando o sistema bastante versátil uma vez que poderá permitir que o serviço docente de um determinado docente seja realizado em mais que uma Distribuição de Serviço Docente.

Grupos de docentes - Edição de registo

Identificação	
Designação:	Grupo de Docentes de Matemática para RAP - Relatórios de Atividade Profissional <small>(Clique sobre a designação para alterá-la)</small>
Docentes	
Docentes a escolher	Docentes do grupo
(a designar 1) (a_designar_1) (a designar 2) (a_designar_2) (a designar 3) (a_designar_3) (a designar 4) (a_designar_4) (a designar 5) (a_designar_5) (a designar 6) (a_designar_6) (a designar 7) (a_designar_7) (a designar 8) (a_designar_8)	Docente Dois (uddois) Docente Três (udtres) Docente Um (udum)
Áreas científicas	
Áreas científicas a escolher	Áreas científicas do grupo
DCBM_CBM - DCBM-Ciências Biomédicas DCBM_MED - DCBM-Medicina ESEC - Spoken English ESEC - ESEC_Direito ESEC - Culturas de expressão inglesa ESEC - Didática do inglês para crianças ESEC - Desenvolvimento da linguagem na criança ESEC - Inglês de nível C2	ISE_M - ISE-Matemática

Figura 4.4 - Edição de grupo de docentes.

Os grupos docentes deverão ainda ter associadas áreas científicas que, de alguma forma, representam os conteúdos que esse grupo tem competência para lecionar. Esta característica, como se detalhará mais tarde, será importante na seleção das disciplinas que determinado docente poderá lecionar uma vez que a estas também será associada uma ou mais que uma área científica. A Figura 4.4 ilustra a composição do grupo de docentes e a associação a apenas uma área científica.

4.2.4. Planos de estudo e unidades curriculares

No SGIP, um plano de estudos significa o plano de estudos ativo (Figura 4.5) de um determinado curso. É composto por unidades curriculares de duração predefinida e que se distribuem ao longo de anos escolares. Essas unidades curriculares são usualmente designadas por disciplinas, termo que passará a ser mais utilizado no presente documento por ter sido o utilizado na conceção do SGIP.

Plano de estudos do curso "Curso Bietápico em Engenharia Civil - 1.º Ciclo - (diurno)"										
Filtros										
Cód.S.Acad.	Designação	Ano Curr.	Docente Resp.	Período	Início Per.Func.	Fim Per.Func.	Opção	Semanal	Acções	
119C15823	Álgebra Linear e Geometria Analítica	1	(sem nome atribuído)	1º semestre	-	-		✓	✗ 📄	
119C15822	Análise Matemática I	1	(sem nome atribuído)	1º semestre	-	-		✓	✗ 📄	
119C15824	Física	1	(sem nome atribuído)	1º semestre	-	-		✓	✗ 📄	
119C15825	Geometria Descritiva	1	(sem nome atribuído)	1º semestre	-	-		✓	✗ 📄	
119C15826	Informática	1	(sem nome atribuído)	1º semestre	-	-		✓	✗ 📄	
119C15828	Língua Estrangeira	1	(sem nome atribuído)	1º semestre	-	-		✓	✗ 📄	
119C15827	Técnicas de Comunicações e Relações Humanas	1	(sem nome atribuído)	1º semestre	-	-		✓	✗ 📄	
119C15829	Análise Matemática II	1	(sem nome atribuído)	2º semestre	-	-		✓	✗ 📄	
119C15832	Desenho de Construção	1	(sem nome atribuído)	2º semestre	-	-		✓	✗ 📄	
119C15835	Engenharia Civil e Ambiente	1	(sem nome atribuído)	2º semestre	-	-		✓	✗ 📄	

Figura 4.5 - Plano de estudos.

Como já foi referido, os planos de estudos são importados através de processos de importação e exportação. O SOPHIA possui o conceito de versões de planos que vão sendo atualizados à medida que sofrem alterações. O SGIP não faz esse tipo de controlo disponibilizando todas as disciplinas passíveis de terem ou terem tido serviço docente. O plano de estudos é, dentro do SGIP, um mero contendor de disciplinas que são disponibilizadas para operação em serviço docente.

Por outro lado, as disciplinas são importadas com muitas das suas propriedades. As principais características importadas das disciplinas são:

- O código dos serviços académicos;
- A designação;
- A duração;

- As datas de início e de fim;
- O docente responsável;
- As escolaridades.

As disciplinas deverão ainda ser associadas às áreas científicas que os gestores de planos e disciplinas entenderem. Só dessa forma as disciplinas serão elegíveis para serviço docente de um grupo de docentes, como se mostra na Figura 4.6.

A Figura 4.5 mostra a interface de plano de estudos em que o autor do documento teve uma grande preponderância, ao invés da interface ilustrada na Figura 4.6 respeitante à edição da disciplina em que o autor teve apenas uma colaboração parcial.

Plano de estudos do curso "Curso Bietápico em Engenharia Civil - 1.º Ciclo - (diurno)" - Edição	
Identificação	
Cód.S.Acad.:	119C15822
Designação:	Análise Matemática I
Ano Curr.:	1
Docente Resp.:	(sem nome atribuido) (fjalmeida)
Período:	1º semestre
Início Per.Func.:	
Fim Per.Func.:	
Opção:	<input type="checkbox"/>
Semanal:	<input checked="" type="checkbox"/>
Nº Alunos (S.Académicos):	0
Áreas Científicas	
Lista de Áreas Científicas	Áreas Científicas Associadas
DCBM_CBM - DCBM-Ciências Biomédicas DCBM_MED - DCBM-Medicina ESEC - Desenvolvimento da linguagem na criança ESEC - Inglês de nível C2 ESEC - ESEC_Direito ESEC - Spoken English ESEC - Didática do inglês para crianças ESEC - Culturas de expressão inglesa ESEC_01 - ESEC-Ciências da Saúde ESEC_01.02.01.13 - ESEC-Prevenção, Segurança e Emergência	ISE_M - ISE-Matemática

Figura 4.6 - Disciplina.

É importante esclarecer que a interface apresentada na Figura 4.6 começou por ser desenvolvida pelo colega Sérgio Baltazar e teve intervenções posteriores do colega Luís Sousa e do autor do documento.

4.2.5. Turnos e sub-turnos

Um turno pode ser entendido como uma turma de docente. É a entidade que é diretamente associável a um docente aquando da elaboração de serviço docente.

Os turnos são turmas de docente criadas com base nas escolaridades da disciplina. As escolaridades da disciplina representam as horas de contacto que um aluno deverá ter com o docente, para cada tipo de escolaridade. Entende-se por tipo de escolaridade o que é conhecido usualmente como tipo de aula (teóricas, práticas, teórico-práticas, etc.).

A criação de turnos no SGIP é da responsabilidade dos elaboradores de serviço docente. Deverão ser criados tantos turnos quantos os que respondam às necessidades do número de alunos previstos para cada tipo de escolaridade, informação da responsabilidade de cada unidade orgânica. Usualmente, é criado pelo menos um turno de cada tipo de escolaridade, assumindo que a disciplina terá de ter inscritos para poder ter serviço docente.

Escolaridades e turnos da disciplina				
Identificação				
Disciplina	Análise Matemática I com 0 aluno(s) previstos.			
Curso(s)	Curso de Curso Bietápico em Engenharia Civil - 1.º Ciclo - (diurno), código 119C15822, 0 aluno(s) previstos.			
Escolaridades	Semanais			
Escolaridades				
Escolaridade	Horas semanais		Acções	
Aulas Teóricas	2		[Ícone]	
Aulas Teórico-Práticas	3		[Ícone]	
[+ Novo registo]				
[✓ Submeter escolaridades]				
Turnos				
Aulas Teóricas - 1 turno(s) :				
Turno	Horas	Docência	Semanas	Acções
Turno 1	2 hora(s)	(sem docente atribuído)	15	[Ícone]
Aulas Teórico-Práticas - 1 turno(s) :				
Turno	Horas	Docência	Semanas	Acções
Turno 1	3 hora(s)	(sem docente atribuído)	15	[Ícone]
[✓ Gerar turnos]				

Figura 4.7 - Escolaridades e turnos da disciplina.

A Figura 4.7 mostra um exemplo da interface de escolaridades e turnos. É importante referir que o autor não teve quase nenhuma intervenção relevante na interface, pelo que esta apenas é mostrada para uma boa compreensão do texto e das funcionalidades.

Um sub-turno foi definido como uma divisão do turno. Serve para representar situações em que um docente inicia um ano letivo lecionando um determinado número de

semanas e, por estratégia pedagógica ou por baixa médica ou algum outro tipo de licença que o impeça de lecionar todas as semanas, é necessário que outro ou outros docentes lecionem o mesmo turno. É útil ainda para representar algumas situações em que o mesmo turno corresponde a várias aulas semanais nas quais exista a necessidade de ter diferentes docentes.

4.2.6. Lecionação conjunta

O conceito de lecionação conjunta é idêntico ao conceito de agregação de disciplinas mas aplicado a turnos. A ideia por detrás do desenvolvimento do conceito é ser capaz de representar alguns turnos da mesma tipologia mas de disciplinas diferentes que são lecionados pelo mesmo docente nos mesmos horários e sala mas que tal não se estenda aos restantes turnos da disciplina.

Na perspetiva do número de horas contabilizado aos docentes que lecionam conjuntamente, apenas são contabilizados os turnos com a maior carga horária do conjunto. Tendo sido um pedido expresso de um representante de unidade orgânica à equipa de desenvolvimento e tendo sido colocado como imprescindível para uma boa representação da realidade, ele foi desenvolvido no SGIP.

Praticamente todo o conceito e interface, integrado na página de gestão de escolaridades e turnos da disciplina, foi desenvolvido pelo autor do documento, tendo sido eventualmente a sua única intervenção na página em causa. O tempo mostrou que esta funcionalidade não foi praticamente utilizada pelo que estava prevista a sua descontinuação, motivo pelo qual se opta por não introduzir mais detalhes sobre este assunto.

4.2.7. Agregação de disciplinas

O conceito de agregação de disciplinas existe na Universidade do Algarve há muitos anos. A sua origem está diretamente relacionada com a estrutura hierárquica de pai único existente na relação entre os cursos e as disciplinas na qual um curso tem muitas disciplinas e uma disciplina pertence a um único curso.

Este tipo de organização dificulta a representação do serviço docente para as disciplinas com uma estrutura de escolaridades mais clássica, tipicamente as que detêm aulas teóricas e aulas práticas. Essa dificuldade assenta no facto de que as aulas teóricas serem frequentemente otimizadas e lecionadas à mesma disciplina de mais que um curso, sendo obrigatório que as disciplinas possuam a mesma tipologia, escolaridades e programa.

Nesses casos, a prática é juntar as turmas teóricas das disciplinas dos cursos diferentes, originando apenas um turno, e deixando turnos separados para as aulas práticas ou agrupando-os também para obter um número de alunos que seja gerível numa sala de aula. Dessa forma, a responsabilidade de criar agregações de disciplinas (Figura 4.8) é dos gestores de planos e disciplinas, seguindo as indicações da sua unidade orgânica.

Disciplinas agregadoras - Edição de registo

Identificação

Designação: [ISE MI+ALGA \(Bietápico Civil + Mecânica\)](#)
(Clique sobre a designação para alterá-la)

Disciplinas

Disciplinas a escolher	Disciplinas da agregação
<div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px;"> ISE_M - ISE-Matemática </div> <div style="border: 1px solid #ccc; padding: 2px;"> <p>1446C1001 - Álgebra Linear e Geometria Analítica <small>1446 - Engenharia Topográfica (1.º Ciclo)</small></p> <p>1449C1001 - Álgebra Linear e Geometria Analítica <small>1449 - Engenharia Civil (1.º Ciclo)</small></p> <p>1451C1001 - Álgebra Linear e Geometria Analítica <small>1451 - Engenharia Civil - Regime Nocturno (1.º Ciclo)</small></p> <p>1451C1000 - Análise Matemática <small>1451 - Engenharia Civil - Regime Nocturno (1.º Ciclo)</small></p> <p>1449C1006 - Análise Matemática Aplicada <small>1449 - Engenharia Civil (1.º Ciclo)</small></p> <p>119C15822 - Análise Matemática I <small>119 - Curso Bietápico em Engenharia Civil - 1.º Ciclo - (diurno)</small></p> <p>1446C1000 - Análise Matemática I <small>1446 - Engenharia Topográfica (1.º Ciclo)</small></p> <p>1441C1010 - Cálculo Numérico <small>1441 - Engenharia Mecânica (1.º Ciclo)</small></p> </div>	<div style="border: 1px solid #ccc; padding: 2px;"> <p>119C15823 - Álgebra Linear e Geometria Analítica <small>119 - Curso Bietápico em Engenharia Civil - 1.º Ciclo - (diurno)</small></p> <p>121C15931 - Matemática I <small>121 - Curso Bietápico em Engenharia Mecânica - 1.º Ciclo</small></p> </div>
Disciplina: 119C15823 - Álgebra Linear e Geometria Analítica [119 - Curso Bietápico em Engenharia Civil - 1.º Ciclo - (diurno)]	

Figura 4.8 - Agregação de disciplinas.

Dentro do SGIP, cada agregação resulta na utilização de uma disciplina única, designada de disciplina de referência, a disponibilizar no serviço docente. As restantes disciplinas que fazem parte da agregação não estão disponíveis de forma individual nem sequer podem ter turnos criados.

Contudo, as agregações são entidades que podem ter várias versões, pelo que se admite que, ao longo do tempo, vão sofrendo alterações na sua composição. Isso significa que

poderão existir disciplinas que até determinado momento não detinham identidade própria para o serviço docente, mas a partir de algum momento passaram a ter turnos criados e áreas científicas associadas. O oposto também pode acontecer.

4.2.8. Áreas científicas

As áreas científicas são um classificador temático que se pode associar a qualquer outra entidade. No SGIP assumiram um papel de grande relevo graças às especificações transmitidas à equipa de desenvolvimento, nas quais se tinha determinado que as disciplinas deveriam ficar disponíveis para serviço docente se possuíssem uma área científica que estivesse presente no grupo de docentes a que dizia respeito o serviço docente.

Existindo várias normas ou recomendações de áreas científicas, as mais relevantes para o ensino superior e na altura a que se reporta eram as conhecidas como CNAEF (Classificação Nacional das Áreas de Educação e Formação), publicadas em Diário da República (D.R. 50 - Série I, Portaria 256/2007 de 12 de Março de 2006) e as do CORDIS⁵. As especificações transmitidas à equipa de desenvolvimento foram no sentido de adotar as áreas do CORDIS, solução que provocou alguma controvérsia no seio das unidades orgânicas. Como resultado final, sendo que a sua obrigatoriedade fazia parte da estrutura central do SGIP, concordou-se em que cada unidade orgânica poderia decidir que áreas científicas utilizariam, sendo que elas seriam carregadas na tabela de áreas científicas a pedido.

As áreas científicas do CORDIS existem numa estrutura hierárquica por vários níveis, que são numerados. Quanto maior o nível mais detalhe da área científica, sendo que a estrutura hierárquica tem como topo uma definição das 6 áreas generalistas:

- Ciências da Saúde;
- Humanidades;
- Ciências Naturais;
- Ciências Físicas;
- Ciências Sociais;
- Ciências Tecnológicas.

⁵ CORDIS versão portuguesa disponível em <https://goo.gl/75Wd45>.

Estas áreas de topo são numeradas de 01 a 06 e todos os subníveis que correspondem a maior detalhe de cada uma das áreas de topo tem um número que se junta ao número da área principal com um separador de ponto final. Por exemplo, a subárea de Ciências Médicas da área de Ciências da Saúde é numerada como 01.02.

Para além do mencionado, as áreas científicas não têm qualquer outro papel relevante no SGIP. Contudo, esse “pequeno” papel é suficientemente relevante uma vez que é necessário que as áreas científicas estejam definidas e associadas aos grupos de docentes e às disciplinas para que seja possível elaborar uma distribuição de serviço docente.

4.2.9. Distribuições de Serviço Docente

4.2.9.1. *Processo de serviço docente*

O processo completo de construção e efetivação da distribuição de serviço docente (DSD) envolve vários procedimentos e vários atores. A entrada em efeito de uma distribuição de serviço docente obriga a um determinado percurso de aprovações designado usualmente por *Workflow de aprovação de distribuições de serviço docente* que será abordado mais à frente no documento.

De uma forma geral, o SGIP consegue representar todo o processo que se sintetiza da seguinte forma:

1. As unidades orgânicas determinam que turnos (turmas de docente) das diferentes tipologias deverão ser criados para cada disciplina.
2. Um gestor de planos e disciplinas da unidade orgânica deverá agregar as disciplinas que serão lecionadas em conjunto, de acordo com o determinado.
3. Um gestor de planos e disciplinas da unidade orgânica cria os turnos pretendidos nas disciplinas respetivas.
4. Um elaborador de serviço docente deverá criar, sempre que necessário, grupos de docentes que corporizem a lógica organizativa da unidade orgânica em relação à elaboração de DSD.

5. Um elaborador de serviço docente cria uma ou várias DSD para os grupos de docentes que respeitam a lógica organizativa da unidade orgânica, atribuindo turnos criados aos docentes que se pretende que lecionem essas disciplinas.
6. Quando se considerar que a DSD está terminada, um elaborador de serviço docente deverá concluí-la, tornando-a disponível para aprovação.
7. Um aprovador de serviço docente deverá analisar as DSD concluídas que estiverem disponíveis e decidir da sua aprovação. Caso aprove a DSD, esta ficará disponível para homologação, caso contrário será remetida para o elaborador para que este proceda às alterações pretendidas.
8. Um homologador de serviço docente deverá analisar as DSD aprovadas que estiverem disponíveis e decidir da sua homologação. Caso homologue a DSD, esta entrará em efeito a partir desse momento, caso contrário será enviada para conhecimento do aprovador para que este, posteriormente, possa remetê-la para o elaborador para proceder às alterações pretendidas.

Os subcapítulos seguintes detalharão todo o processo que envolve as DSD. De notar que, apesar de ter sido o autor a implementar o *workflow* das DSD, a sua definição foi realizada pelos representantes das unidades orgânicas, em reuniões designadas para acompanhar o andamento dos trabalhos do SGIP.

4.2.9.2. Página de gestão da Distribuição de Serviço Docente

A página de *Gestão de DSD* está acessível no menu de ***Distr. Serviço Docente*** no menu geral do SGIP. Esta página está pensada para poder dar resposta aos utilizadores que são elaboradores, aprovadores ou homologadores da DSD, fornecendo caixas de DSD distintas de acordo com o estado de cada uma e, conseqüentemente, as operações disponíveis para as mesmas.

A ideia de base é que deveriam existir 3 combinações de caixas:

1. Para os elaboradores de DSD:
 - a. DSD em Elaboração e DSD Não Aprovadas ou Remetidas;
 - b. DSD em Outros Estados.

2. Para os aprovadores de DSD:
 - a. DSD Por Aprovar e DSD Não Homologadas.
3. Para os homologadores de DSD:
 - a. DSD Por Homologar e DSD Homologadas.

Se um utilizador tiver mais que um papel ser-lhe-ão mostradas todas as caixas associadas aos papéis detidos pelo utilizador.

Em cada uma das caixas serão mostradas as DSD que estejam nos estados correspondentes. Poderão existir dentro da mesma caixa dois tipos de estado: os estados em que o utilizador pode interagir de alguma forma com a DSD e outros estados em que a ideia da lista é informar o utilizador do respetivo estado. Por exemplo, um aprovador de DSD poderá ter na sua caixa de “DSD Por Aprovar e DSD Não Homologadas” várias DSD no estado *Concluído*, as quais poderá aprovar ou reprovar, e ainda várias DSD que já estejam aprovadas indicando que se encontram para homologação pelo respetivo homologador. Nesse caso, o acesso à DSD é apenas de leitura.

Gestão da Distribuição de Serviço Docente - Destaques

DSD Por Homologar

Designação	Ano lectivo	Período	Grupo docentes	Estado	Consulta	Relatórios	Acções
DSD do Grupo de Docentes de Teste II	2015/2016	Ano	Grupo de Docentes de Teste II	Aprovado			

DSD Por Aprovar e DSD Não Homologadas

Designação	Ano lectivo	Período	Grupo docentes	Estado	Consulta	Relatórios	Acções
DSD do Grupo de Docentes de Teste II	2015/2016	Ano	Grupo de Docentes de Teste II	Aprovado			
DSD do Grupo de Docentes de Teste I	2015/2016	Ano	Grupo de Docentes de Teste I	Elaborado			

DSD em Elaboração e DSD Não Aprovadas ou Remetidas

Designação	Ano lectivo	Período	Grupo docentes	Estado	Consulta	Relatórios	Acções
Curso Técnico Superior Profissional (2015/2016)	2015/2016	Ano	Curso Técnico Superior Profissional (2015/2016)	Em elaboração			

Adicionar nova DSD | Copiar DSD de ano transacto

DSD em outros estados

Designação	Ano lectivo	Período	Grupo docentes	Estado	Consulta	Relatórios	Acções
Sem registos							

Figura 4.9 - Interface de Gestão da DSD para um administrador.

A Figura 4.9 ilustra a interface de Gestão da DSD para um administrador do SGIP. Deve notar-se o exemplo da “DSD do Grupo de Docentes de Teste II”, que se encontra no estado Aprovado e que é mostrada na caixa destinada ao Homologador, na qual este poderá realizar a ação que está determinada: homologar. Contudo, a mesma DSD também é apresentada na caixa destinada ao Aprovador, sinalizando que a DSD foi aprovada e que ainda não foi sujeita à apreciação de homologação. Em qualquer das caixas se encontram disponíveis as ações de visualização e impressão da DSD e ainda a visualização do relatório de aprovação.

4.2.9.3. Criação

As distribuições de serviço docente são criadas por elaboradores de serviço docente. Estes utilizadores especiais têm a missão de definir os grupos de docentes para os quais se vai construir o serviço docente, atribuir-lhes áreas científicas, adicionar disciplinas e, posteriormente, atribuir turnos ao docente.

O processo de criação inicia-se com o fornecimento da informação base. Esta informação de base, composta maioritariamente por campos não alteráveis posteriormente, define o contexto da DSD sendo, por isso, estruturante na forma como esta se elabora. Estas informações de base são compostas por:

- **Designação:** nome pelo qual a DSD passa a ser conhecida;
- **Ano letivo:** não editável, sujeito ao ano letivo ativo escolhido pelo operador;
- **Período:** abrangência da DSD em relação às disciplinas;
- **Grupo de docentes:** grupo de docentes ao qual diz respeito a DSD;
- **Estado:** não editável, a DSD inicia-se sempre no estado *Em Elaboração*.

A designação poderá ser alterada posteriormente pelo elaborador. O estado é definido automaticamente como *Em Elaboração* e poderá ser alterado posteriormente de forma automática seguindo o *workflow* de aprovação e homologação da DSD, como será abordado mais à frente. Os restantes campos não serão alteráveis de nenhuma forma depois da DSD estar criada.

Uma outra forma de criar uma nova DSD é através da cópia de uma DSD do ano transato. Para que esta operação seja possível, o grupo de docentes tem de ter sido criado como parte do processo de cópia de ano letivo, processo detalhado mais à frente no presente relatório, o mesmo acontecendo com as disciplinas. Caso os turnos “equivalentes” ou os mesmos docentes não se encontrem no novo ano letivo, as atribuições de turnos respetivos não serão copiadas. Na página de cópia, apenas são carregadas as DSD do ano transato passíveis de ser copiadas.

Uma vez criada a DSD, poderão ser adicionadas as disciplinas que se pretende que estejam incluídas no serviço docente presente nessa DSD. A escolha das disciplinas para a DSD poderá ser executada mais tarde, sendo que poderão ser associadas ou desassociadas disciplinas em qualquer altura da fase de elaboração.

A escolha das disciplinas faz-se partindo do princípio base utilizado no SGIP: apenas estão disponíveis para serviço docente as disciplinas que tenham uma área científica associada ao grupo de docentes a que diz respeito o serviço docente. Essa escolha é executada através de um sistema de lista dupla, sendo que apenas aparecerão disponíveis para escolha na lista da esquerda as disciplinas que respeitarem o princípio indicado.

4.2.9.4. Construção e alteração

4.2.9.4.1. Organização da página de edição da DSD

A fase de construção da DSD é aquela em que o elaborador associa turnos a docentes para representar o serviço docente. Esta operação deve ser precedida pela escolha das disciplinas, como mencionado anteriormente aquando da criação da DSD.

Na fase de construção da DSD é possível também afetar outras atividades aos docentes que fazem parte do grupo de docentes. Estas atividades são as que estão devidamente regulamentadas como fazendo parte das obrigações dos docentes e que têm a ver com horas a cargos, orientações e outras. A ideia da colocação destas horas nas DSD é a possibilidade de ter uma panorâmica de todo o serviço atribuído a um docente.

Gestão da DSD - Edição de registo

Identificação
 Designação: [DSD do Grupo de Docentes de Teste I](#)
(Clique sobre a designação para alterá-la)

Dados de referência

Disciplinas

Sumário de actividade

		Eq. semanal	Eq. anual
Docente Dois	Horas de contacto	6.58	197.5
	Outras actividades	6	180
	Total	12.58	377.5
Docente Um	Horas de contacto	12	360
	Outras actividades	0	0
	Total	12	360

[+ Adicionar nova actividade](#) |
 [Gerir por disciplina](#) |
 [Gerir por docente](#) |
 [Vista da DSD por disciplina](#) |
 [Vista da DSD por docente](#)

[Agrupar por docente](#) |
 [Agrupar por disciplina](#) |
 [Incluir períodos dependentes](#) |
 [Incluir outras DSD](#)

Docente Dois Acções

Horas de contacto

Disciplina	Curso	Actividade	Turno	Horas	Acções
Testes Especificos I	Mestrado em Testes Exaustivos	Aulas Teóricas	Turno 1	1.5 Hora(s)	
Testes Especificos I	Mestrado em Testes Exaustivos	Práticas Laboratoriais	Turno 1	4.5 Hora(s)	
Testes Especificos II	Mestrado em Testes Exaustivos	Aulas Teóricas	Turno 1	1.5 Hora(s)	
Testes Especificos II	Mestrado em Testes Exaustivos	Práticas Laboratoriais	Turno 1	4.5 Hora(s)	
Testes Gerais III	Mestrado em Testes Exaustivos	Trabalho de campo	Turno 1.2	5 Hora(s)	
Testes Gerais III	Mestrado em Testes Exaustivos	Trabalho de campo	Turno 1.1	10 Hora(s)	

Outras actividades

Horas a Cargos	Vice-Presidente Conselho Diretivo	6 Hora(s)	30 Semana(s)	
----------------	-----------------------------------	-----------	--------------	--

Docente Um Acções

Horas de contacto

Figura 4.10 - Página de edição de DSD.

A construção da DSD dá-se início assim que se cria a DSD ou sempre que se escolher a opção de alteração na página de gestão de DSD. A página de edição da DSD (Figura 4.10) é composta por várias caixas:

1. **Identificação:** visualizar ou alterar a designação da DSD;
2. **Dados de referência:** informação não alterável de informação de base com o ano letivo e o período a que diz respeito a DSD, o grupo de docentes e o estado em que se encontra a DSD;
3. **Disciplinas:** lista dupla para escolha das disciplinas que fazem parte da DSD, na qual a lista da esquerda apresenta as disciplinas disponíveis para a DSD e a lista da esquerda contém as disciplinas que já fazem parte da DSD;
4. **Sumário de actividade:** sumário de horas atribuídas a cada docente;
5. **Serviço docente:** caixas para o serviço docente, por docente ou por disciplina, dependendo da opção de visualização escolhida;
6. **Zonas de botões:** botões de alteração ou visualização da DSD e botões para imprimir ou voltar à página de gestão de DSD, sendo que alguns destes botões aparecem repetidos antes e depois das caixas de serviço docente.

Assim que se entra em edição da DSD poder-se-á escolher um de dois modos de visualização: visualização por docente, a opção por defeito assim que se entra, e a visualização por disciplina. A diferença reside em como são agrupados os detalhes a apresentar na respetiva lista.

4.2.9.4.2. Coeficiente de trabalho noturno

O coeficiente ou fator de trabalho noturno, como também é mencionado, é um dos parâmetros que influencia a forma como o cálculo do total de horas de lecionação do docente é calculada. Este fator pode ser associado a uma atribuição de turno a um docente e tem um valor de 1,5 de acordo com a legislação existente na altura em que o SGIP foi desenvolvido.

Na página de edição de DSD, todas as atribuições de turnos a docentes podem ser sujeitas à atribuição deste fator. Por omissão, este fator não é aplicado mas está sempre disponível através de um botão com o ícone lua, que passa a um sol quando o fator está aplicado. Não foi possível aplicar automaticamente esse fator porque depende de condições contratuais do docente e das horas de início da lecionação do turno, informações não disponíveis no SGIP.

4.2.9.4.3. Horas semanais, horas totais e horas equivalentes

Como já foi referido, o SGIP tem como principal fornecedor de dados de base a aplicação de gestão académica SOPHIA. Como tal, há determinados factos que têm a ver com a forma como esta aplicação trabalha e fornece informação ao exterior.

Um pouco tardiamente, a equipa de desenvolvimento do SGIP deparou-se com uma realidade em relação às escolaridades das disciplinas. As horas das escolaridades poderiam estar expressas na aplicação de gestão académica como horas semanais ou horas totais, tal como foram publicadas em Diário da República. Este pequeno pormenor não fazia qualquer diferença para a gestão académica, uma vez que estes dados não são trabalhados no SOPHIA, mas representou um constrangimento importante para o SGIP.

Para não introduzir qualquer conversão que poderia conduzir a erros de grande magnitude em relação às horas, foi decisão da equipa registar as escolaridades internamente tal como foram importadas. No modelo de dados sinalizou-se esta realidade num campo booleano associado à disciplina, ou seja, em cada disciplina apenas se trabalha com horas semanais ou horas totais. No nono campo da Figura 4.6, relativa à página de edição da disciplina, pode observar-se facilmente esta propriedade.

Numa tentativa de normalização da carga horária de cada docente nas DSD criou-se o conceito de *horas equivalentes*. Este conceito nunca foi de forma alguma aprovado superiormente e representa a visão da equipa de desenvolvimento relativa a esta normalização.

4.2.9.4.4. Cálculo das horas equivalentes

As horas equivalentes, conceito apresentado no item anterior, teve um prévio trabalho de conceito e a sua forma de cálculo foi documentada face às dúvidas que usualmente se colocavam pelos vários intervenientes da aplicação. Como explicado, é um conceito da autoria exclusiva da equipa de desenvolvimento com grande responsabilidade do autor do documento, mas nunca sofreu qualquer tipo de validação por parte dos vários responsáveis da UAlg.

Este cálculo assenta num conjunto de pressupostos que é essencial apresentar ou recordar, uma vez que alguns já foram referidos no presente documento:

- O serviço docente de todas as DSD está sempre expresso numa base anual, daí o termo “equivalente” aplicado a todos os cálculos;
- De acordo com a *praxis* das UO, o número de semanas de um ano letivo é expresso num total de 30 semanas, sabendo que, utilizando esta base, a um semestre correspondem 15 semanas, a um quadrimestre 10 semanas, e por aí adiante;
- As escolaridades de uma unidade curricular podem ser expressas em número de horas totais ou em número de horas semanais;

- O total de horas de lecionação de um turno é afetado por um fator denominado coeficiente de trabalho que ponderará os casos em que o turno é considerado trabalho noturno e o docente se encontra a 100% e com exclusividade;
- Desenvolveu-se o conceito de lecionação conjunta como sendo um conjunto de turnos que são lecionados em conjunto; nestes casos, apenas é contabilizado no serviço docente o turno que possua maior carga horária;
- Foi desenvolvido o conceito de sub-turno para poder refletir de alguma forma um turno que não será integralmente lecionado por um docente; por exemplo, nos casos de substituição de docentes a meio do período da disciplina, o turno é desde logo planeado para ser lecionado por vários docentes, e outros casos que o exijam; o coeficiente de sub-turno reflete, de forma proporcional, a percentagem de horas em relação ao total do turno que esse sub-turno representa.

Desta forma, apresentam-se as fórmulas de cálculo das horas equivalentes. Na primeira fórmula considera-se t_{se} como o total de horas semanais equivalentes obtidas através da soma do total de horas semanais de contacto equivalentes t_{sce} e do total de horas semanais de outras atividades equivalentes t_{sae} :

$$t_{se} = t_{sce} + t_{sae}, \quad (1)$$

com t_{se} a carga horária semanal do docente na DSD na base do ano letivo, t_{sce} as horas de contacto contabilizadas em horas semanais do docente na DSD na base do ano letivo e t_{sae} o total de outras atividades contabilizadas em horas semanais do docente na DSD na base do ano letivo.

Também para as horas totais ou anuais, se considera t_{ae} como o total de horas anuais equivalentes cujo valor é obtido pela soma do total de horas de contacto anuais equivalentes t_{ace} e do total de horas anuais de outras atividades t_{aae} :

$$t_{ae} = t_{ace} + t_{aae}, \quad (2)$$

com t_{ae} a carga horária anual do docente na DSD na base do ano letivo, t_{ace} o total de horas de contacto do docente na DSD na base do ano letivo e t_{aae} o total de outras atividades do docente na DSD na base do ano letivo.

Utilizando:

$$t_{sce} = \sum_{\text{turnos}} \left(\frac{c_{lc} * c_t * c_{st} * n_{hp}}{n_{sr}} \right) \quad (3)$$

$$t_{ace} = \sum_{\text{turnos}} (c_{lc} * c_t * c_{st} * n_{hp}) \quad (4)$$

$$t_{sae} = \sum_{\text{atividades}} \left(\frac{n_{ha} * n_{sa}}{n_{sr}} \right) \quad (5)$$

$$t_{aae} = \sum_{\text{atividades}} (n_{ha} * n_{sa}) \quad (6)$$

Considerando que:

$$c_{lc} = \begin{cases} 1, & \text{caso não se trate de lecionação conjunta ou seja o turno de} \\ & \text{maior carga horária total da lecionação conjunta} \\ 0, & \text{caso contrário} \end{cases} \quad (7)$$

$$c_{st} = \begin{cases} \frac{n_{sst}}{n_{sd}}, & \text{caso se trate de um subturno} \\ 1, & \text{caso se trate de um turno} \end{cases} \quad (8)$$

$$n_{hp} = \begin{cases} n_{he} * n_{sd}, & \text{caso escolaridades semanais} \\ n_{he}, & \text{caso escolaridades totais} \end{cases} \quad (9)$$

E em que:

- c_{lc} : Coeficiente de lecionação conjunta;
- c_t : Coeficiente de trabalho a utilizar para contabilizar horas noturnas;
- c_{st} : Coeficiente de proporcionalidade entre a duração do turno e a carga efetiva do sub-turno atribuída ao docente;
- n_{hp} : Número de horas totais do turno no período da disciplina;
- n_{sr} : Número de semanas de referência do ano letivo (30);

n_{ha} :	Número de horas da outra atividade;
n_{sa} :	Número de semanas da outra atividade;
n_{sst} :	Número de semanas do sub-turno;
n_{sd} :	Número de semanas da disciplina;
n_{he} :	Número de horas da escolaridade da disciplina.

4.2.9.4.5. Atribuição de turnos

Para a atribuição de turnos foram construídas duas interfaces, designadas por “Gestão por docente” e “Gestão por disciplina”. A diferença entre as duas reside apenas no que está no centro da mesma, sendo que na primeira indicada o centro é o docente e na segunda o centro é a disciplina. Basicamente na primeira seleciona-se numa caixa de combinação o docente de que se pretende gerir os turnos e na segunda seleciona-se a disciplina.

Em ambas as interfaces se pode utilizar o pegar-e-largar⁶ para mover um turno que está por atribuir para a caixa dos turnos atribuídos e vice-versa. O mesmo procedimento pode ser realizado através de duplo-clique em cima de um turno, fazendo com que um turno livre se torne atribuído ao docente escolhido ou um turno atribuído fique livre.

A gestão da atribuição de turnos por docente tem essencialmente duas caixas: uma onde se encontram os turnos atribuídos ao docente e na outra onde se encontram os turnos livres. A forma de operar esses turnos é feita como explicado nos parágrafos anteriores, após a seleção do docente pretendido.

A gestão da atribuição de turnos por disciplina é um pouco mais complexa. É composto essencialmente por duas áreas, a área da disciplina e a área do docente. Na área de disciplina mostra os turnos livres da disciplina numa caixa e os turnos atribuídos dessa mesma disciplina noutra caixa. Na área de docente, permite escolher o docente pretendido mostrando os turnos atribuídos a esse mesmo docente. A forma de operar é semelhante, sabendo que duplo-clique em cima de um turno livre efetiva a atribuição

⁶ Do inglês *Drag-and-Drop*, mecanismo pelo qual se utiliza o rato para pegar num objeto em largá-lo em qualquer outro lado.

desse turno ao docente escolhido na área de docente e duplo-clique em cima de um turno atribuído ao docente resulta na libertação desse turno.

4.2.9.4.6. Definição de outras atividades

A interface de outras atividades é das mais antigas do SGIP. Não sofreu alterações de maior uma vez que cumpre o seu objetivo principal: lançar outras atividades dos docentes para que as horas possam ser contabilizadas na sua atividade profissional.

As outras atividades têm as seguintes características:

- **Designação:** aparecerá na listagem da DSD;
- **Docente:** docente a que diz respeito a atividade;
- **Tipo de atividade:** horas a cargos, orientações ou outras;
- **Número de horas:** número de horas que deverão ser contabilizadas por semana;
- **Número de semanas:** número de semanas a considerar.

Propositadamente os últimos dois campos não foram fechados para que pudessem admitir qualquer valor. A razão é simples, a não existência de uma regra na Universidade que permita aferir um conjunto de valores precisos à data da elaboração da aplicação, sendo que posteriormente houve indicações para tal. Contudo, ficaria sempre por esclarecer o número de semanas que deveriam ser contabilizadas, tendo em conta que, de uma forma empírica, é hábito determinar-se que um ano letivo tem 30 semanas de lecionação, logo que um semestre tem 15 e que todas as durações dos restantes períodos são obtidas de forma proporcional. Esses valores são definidos no SGIP na aquando da definição dos períodos para o ano letivo.

4.2.9.5. Anulação

A anulação de uma DSD é possível desde que ela se encontre no estado *Em Elaboração*. Esta exigência tem a ver com o procedimento seguido através do *workflow* de aprovação sendo que não faz sentido anular uma DSD que se encontra fora do controlo do elaborador.

Se não se pretender aprovar ou homologar uma DSD basta que se atue negativamente num qualquer desses estados. Assim, não se considerou consistente anular uma DSD aprovada ou homologada, o que deixa o contexto da anulação para a elaboração.

4.2.9.6. Visualização

Foi desenvolvido um modo de visualização para as DSD com o intuito de mostrar alguma informação extra no cabeçalho que ajudasse os elaboradores e outros utilizadores a ter uma melhor perceção das horas associadas a cada docente.

Na realidade, o detalhe dos turnos atribuídos no modo de visualização é idêntico ao utilizado na edição da DSD. A grande diferença está na apresentação da lista de docentes que fazem parte do grupo de docentes ao qual diz respeito a DSD. Nessa lista são mostrados os totais de horas de docência e de outras atividades, tornando assim a visualização muito interessante para quem pretende verificar as horas atribuídas a cada docente.

4.2.9.7. Conclusão

A conclusão da DSD é a ação pela qual o elaborador sinaliza o seu término. É extremamente importante porque é através desta ação que se inicia o *workflow* de aprovação e homologação da DSD.

A conclusão de uma DSD provoca a sua mudança de estado de *Em Elaboração* para *Elaborado*, como se poderá verificar mais à frente no presente documento. Essa simples mudança de estado, para além de inibir quaisquer modificações à DSD, coloca-a disponível ao aprovador para que este a possa apreciar. Até esse momento, o aprovador não detinha qualquer acesso à mesma.

4.2.10. Workflow de aprovação das Distribuições de Serviço Docente

O circuito de aprovação e homologação das DSD, usualmente designado por *workflow* de aprovação das Distribuições de Serviço Docente, é a sequência de estados possíveis para

uma DSD. Foi um consenso difícil de encontrar e sintetizar entre os pares representantes das unidades orgânicas atendendo à praxis de cada uma na altura em que foi apresentado e discutido.

O *workflow* considerado detinha vários tipos de entidades que importam apresentar:

- **Estados:** os estados definem o posicionamento da DSD no *workflow*;
- **Intervenientes:** os intervenientes do circuito são de três tipos:
 - *Elaborador:* constrói a DSD e faz todas as alterações necessárias;
 - *Aprovador:* o aprovador é o interveniente a quem cabe o primeiro nível de apreciação e decisão, detendo a capacidade de aprovar ou reprovar a DSD;
 - *Homologador:* o homologador é o interveniente que tem a capacidade de homologar ou não uma DSD num segundo e último nível a cadeia de apreciação e decisão.
- **Ações:** as ações diferem de estado para estado, estando ou não disponíveis consoante os intervenientes, e provocam, quase sempre, uma mudança de estado.

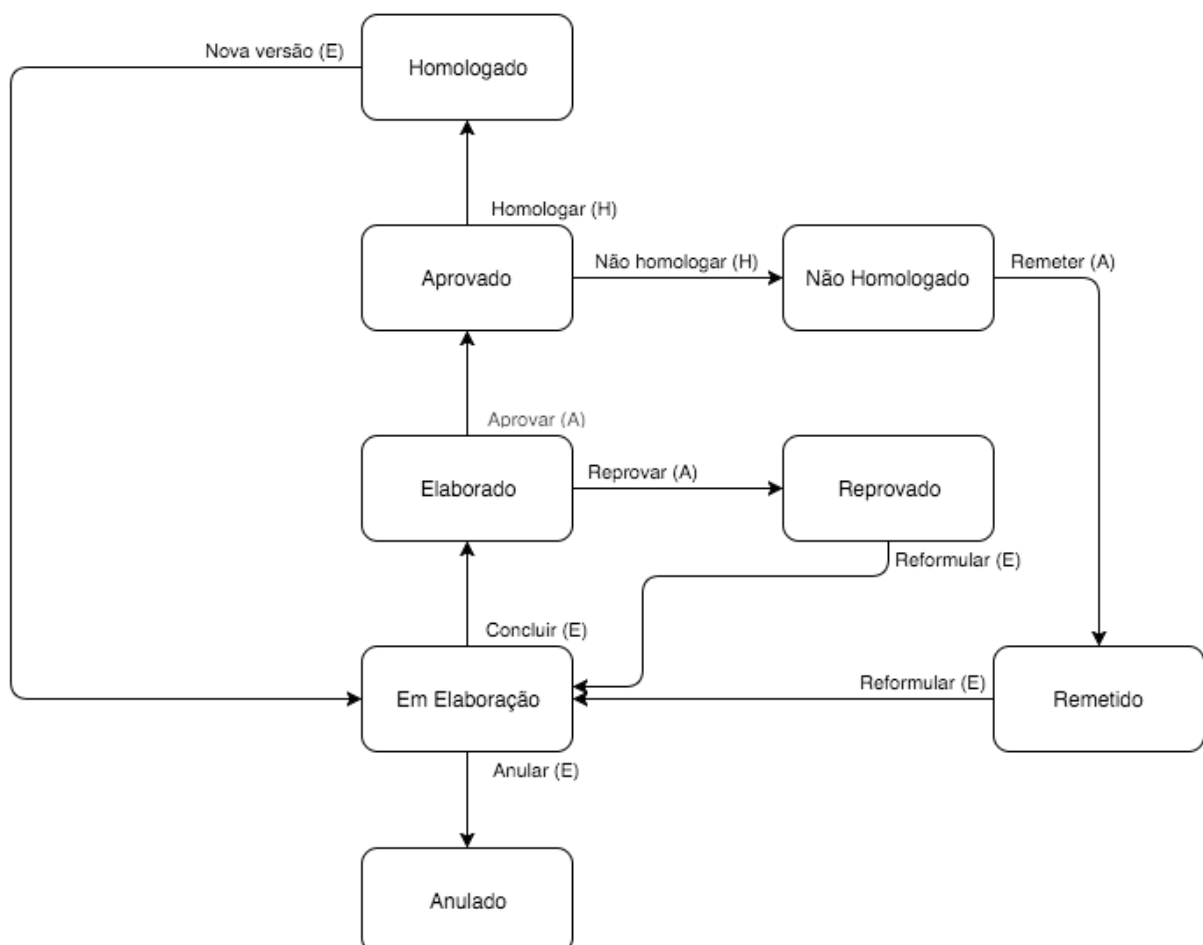


Figura 4.11- Diagrama de transição de estados do SGIP.

Apresenta-se em Figura 4.11 o diagrama de transição de estados do SGIP. Cada uma das operações passíveis de realizar em cada estado têm inscrito uma letra entre parêntesis. Essa letra representa o interveniente que pode executar a ação inscrita, sendo que **E** significa Elaborador, **A** significa Aprovador e **H** significa Homologador.

Existem duas questões que importam considerar. A primeira diz respeito ao interveniente no estado *Não Homologado* e o facto do interveniente ser o Aprovador e não o Elaborador, que talvez fosse mais lógico numa primeira apreciação. Mas esse cenário refletiria uma situação em que a DSD teria sido aprovada pelo Aprovador e depois não homologada e remetida para o Elaborador sem que o Aprovador tivesse conhecimento. Assim, para permitir que o Aprovador tivesse conhecimento da não homologação, entendeu-se que a não homologação colocaria a DSD na mão do Aprovador e não diretamente na do Elaborador.

A segunda questão diz respeito ao estado *Anulada*. Como se pode reparar no diagrama, o estado está representado como um ponto de fim embora no SGIP este estado estivesse implementado de forma a possibilitar uma nova versão. Na realidade, essa seria uma alteração a realizar posteriormente uma vez que a simples possibilidade de surgir uma nova versão a partir de uma DSD anulada induz uma complexidade enorme ao nível das consultas à base de dados.

4.2.11. Abertura do ano letivo

4.2.11.1. Procedimento de abertura do ano letivo

Na altura de desenvolvimento acelerado do SGIP, uma das especificações que foram fornecidas aos programadores é que a informação mais correta relativa a planos e disciplinas passaria a ser a do SGIP uma vez que refletiria a realidade do serviço docente nas várias unidades orgânicas. Esta ideia foi transmitida informal e oralmente à equipa de desenvolvimento e implicou que fosse desenvolvido um procedimento que garantisse que o ano letivo n seria replicado para o ano letivo $n+1$.

Para que tal fosse possível, atendendo à natureza independente de cada ano letivo no sistema de informação do SGIP, foi desenvolvido um procedimento de cópia dos registos

de um ano letivo para o seguinte. De uma forma geral, tal procedimento implica a capacidade de copiar as seguintes entidades:

- Planos de estudos;
 - Disciplinas;
 - Escolaridades;
 - Associação às áreas científicas;
 - Turnos.
- Grupos de docentes (inclui associação às áreas científicas);
- Agregações.

Adicionalmente, o sistema permite configurar a execução de alguns procedimentos antes e depois da cópia dos registos configuráveis para a preparação do ano letivo. As operações de pré ou pós processamento da cópia desenvolvidas foram:

- **ACTIVA_CURSO:** Ativação de curso desativado;
- **DEACTIVA_CURSO:** Desativação de curso;
- **DUPLICA_DISCIPLINA:** Duplicação de uma disciplina.

Por outro lado, nas seções *Params*, é permitido colocar operações de seleção dos dados a copiar com valores que poderão ser *inteiros*, *strings*, *booleanos* ou *data/hora*. Essas operações de seleção poderão selecionar com dados que respeitem determinado valor literal ou o resultado de uma *query*, utilizando de operadores de EQUAL, NOT_EQUAL, GREATER_THAN, GREATER_EQUAL, LESS_THAN, LESS_EQUAL, LIKE, NOT_LIKE, IN e NOT_IN. As operações de seleção poderão ainda ser simples ou compostas, sendo que neste último caso elas poderão relacionar-se com outras através de operadores AND, OR e NOT.

Os itens mencionados com algum detalhe são configuráveis num ficheiro em formato YAML (YAML, 2016). A Figura 4.12 ilustra um extrato de um exemplo real de um ficheiro de configuração utilizado para a cópia dos dados relativos ao ano letivo de 2011/2012 para 2012/2013 (o ficheiro completo pode ser consultado no ANEXO VI). Como se pode ver, há configurações que excluem alguns dados que, no caso apresentado, já haviam sido criados no SGIP. Tal como foi explicado, a sintaxe a utilizar no ficheiro de configuração permite diversas operações.

```

# Configuração do ano lectivo de 2012/2013
#
# IMPORTANTE
# Deverão ser excluídos os cursos e da FCT porque já foram
copiados antes
#
---
execucao: ~
# Nesta versão, para já, não há necessidade de pré e pós
execuções
config:
  # IMPORTANTE
  # ---
  # A sub-secção execucao existe para permitir a restrição do
que há a copiar
  # Ou seja, qualquer elemento NÃO mencionado aqui SERÁ SEMPRE
COPIADO
  # ---
  # Por decisão de implementação, a sub-secção params apenas
será considerada
  # para as fases de primeiro nível, isto é, planos, grupos e
agregacoes.
  execucao:
    planos:
      exec: true
      params:
        id_entidade:
          tipo: integer
          operador: NOT_EQUAL
          comparador: valor
          valor: 22
        sub:
          disciplinas:
            exec: true
            sub:
              escolaridades:
                exec: true
                sub: ~
              areas:

```

Figura 4.12 - Extrato de um exemplo real de um ficheiro de configuração utilizado para a cópia dos dados relativos ao ano letivo de 2011/2012 para 2012/2013.

Basicamente, o algoritmo de cópia do ano letivo sintetiza-se como:

1. Criação do registo de ano letivo;
2. Associação dos períodos de funcionamento com informação de datas de início e de fim;

3. Cópia do ano letivo através de um script em CLI que segue as definições do ficheiro de configuração;
4. Formalizar a abertura do ano letivo dentro do SGIP.

Esta informação foi registada com detalhe no *wiki* de desenvolvimento do projeto, incluindo os procedimentos para disponibilização da informação na tutoria e ainda pequenas definições a realizar para que a importação se comece a realizar para o ano letivo que acaba de ser copiado e aberto (v.d. ANEXO VII; este documento foi elaborado em parceria). A razão pela qual esses pormenores não são detalhados no presente documento é que, apesar do script descrito ter sido desenvolvido pelo autor do documento, as restantes configurações não foram da sua responsabilidade.

4.2.11.2. Interface de lançamento de ano letivo

Tal como mencionado anteriormente, o procedimento de abertura de ano letivo implica a criação do registo do ano letivo. Este registo reveste-se de alguma singularidade pelo facto de ser através dele que se definem as durações, em número de semanas, dos vários períodos disponíveis na aplicação. A Figura 4.13 ilustra a interface de criação e edição de um registo de ano letivo.

Anos Lectivos - Edição de registo			
Identificação			
Código:	20152016		
Designação:	2015/2016		
Períodos			
Período	Período pai	Nº Semanas	Ações
Ano		30	
1º semestre	Ano	15	
2º semestre	Ano	15	
Módulo 1	1º semestre	8	
Módulo 2	1º semestre	8	
Módulo 3	2º semestre	8	
Módulo 4	2º semestre	8	
1º Quadrimestre	Ano	10	
2º Quadrimestre	Ano	10	
3º Quadrimestre	Ano	10	
2º Semestre (MECE)	Ano	15	
1º Semestre (MECE)	Ano	15	
Todos Semestres	Ano	15	

Figura 4.13 - Interface de lançamento de ano letivo.

Tornou-se pertinente o desenvolvimento de tal interface num período em que a UAlg pretendeu acabar a lecionação por módulos e iniciou um período transitório no qual o regime de 6 módulos anuais se tornou num regime de apenas 4 módulos, ou seja, 2 módulos por semestre. Impunha-se então que a duração dos vários períodos fosse suficientemente flexível para poder representar convenientemente quaisquer alterações relativas aos vários tipos de período de funcionamento.

4.2.11.3. Interface de períodos de funcionamento

A interface de períodos de funcionamento permite realizar o detalhe de cada período de funcionamento para cada ano letivo, unidade orgânica e período de funcionamento. O nome “período de funcionamento” não terá sido o mais feliz porque se confundiu frequentemente com o tipo de período de funcionamento (ano, 1º semestre, etc). Apresenta-se na Figura 4.14 um exemplo da edição de um período de funcionamento.

Períodos de funcionamento - Edição de registo	
Identificação	
Ano Lectivo:	2015/2016
Unidade Orgânica:	Instituto Superior de Engenharia
Designação:	Ano <small>Designação opcional do período</small>
Período	
Período:	Ano
Início primeiro ano:	2015-09-14 <small>Data de início do período do 1º ano escolar no formato aaaa-mm-dd</small>
Início restantes anos:	2015-09-14 <small>Data de início do período dos restantes anos escolares no formato aaaa-mm-dd</small>
Fim primeiro ano:	2016-07-01 <small>Data de fim do período do 1º ano escolar no formato aaaa-mm-dd</small>
Fim restantes:	2016-07-01 <small>Data de fim do período dos restantes anos escolares no formato aaaa-mm-dd</small>

Figura 4.14 - Interface de edição de períodos de funcionamento.

O período de funcionamento corresponde à definição do calendário escolar vigente na altura em que o SGIP teve o seu início de desenvolvimento. Na sua definição, era importante que se pudessem lançar datas diferentes para o início e fim do primeiro ano escolar e para os restantes, distinção que era praticada em algumas unidades orgânicas.

Aquando da definição de um período de funcionamento, é colocada a possibilidade de propagação das datas de início e fim relativas a esse período de funcionamento para as disciplinas que pertençam à unidade orgânica e ao período definidos. Essas datas apenas serão gravadas no registo de disciplina caso não estejam preenchidas ou as existentes no registo não se enquadrem no ano letivo atual.

4.2.12. Interfaces de gestão de dados

4.2.12.1. Introdução

Tal como apresentado anteriormente, as interfaces de gestão de dados são a única forma de tornar novos cursos ou disciplinas “válidos” para o SGIP. A noção de “válido” apresentada tem a ver com o facto de, se os cursos ou disciplinas forem criados manualmente, ser-lhes-á atribuído o código 0000. Como se pode perceber, esse código não é utilizável para qualquer situação de referenciação uma vez que não é único nem universal.

As operações que se poderão realizar através das interfaces de gestão de dados são:

- Criação de novos cursos ou novas disciplinas, importados da aplicação de gestão académica;
- Associação de novos cursos ou novas disciplinas, importados da aplicação de gestão académica, a cursos ou disciplinas criados no SGIP, logo detentores do código 0000;
- Anulação segura das operações descritas.

Quando são mencionadas a criação ou associação de cursos, deverá entender-se como a criação ou associação de cursos e das respetivas disciplinas oriundas da aplicação de gestão académica. Ao contrário, a criação ou associação de uma disciplina apenas deverá considerar a operação sobre essa disciplina.

Para as operações descritas foram criadas 4 interfaces:

- Interface de criação e associação de cursos;
- Interface de criação e associação de disciplinas;

- Interface de anulação de operações sobre cursos;
- Interface de anulação de operações sobre disciplinas.

As próximas Secções detalharão um pouco estas interfaces.

4.2.12.2. Interface de criação e associação de cursos

Na interface de criação e associação de cursos (Figura 4.15) poderão ser criados cursos que serão acrescentados à lista de cursos disponíveis no ano letivo em operação ou ainda associados cursos obtidos por importação a cursos criados no SGIP.

A criação de um curso envolve a criação do registo do curso, mas também das disciplinas que a ele pertencem. A criação do curso é realizada em dois passos: no primeiro é selecionado o curso e utilizado o botão de “Criar novo curso” para indicar a intenção de o criar e no segundo passo são mostradas as disciplinas importadas dando hipótese ao operador de seleccionar quais as que quer criar e as que vai deixar como não resolvidas.

Cursos: Resolução de problemas de importação

Serviços Académicos

1588 Estudos Artísticos - Regime Pós-laboral (1.º Ciclo)

500 Extra-curricular

1754 Gestão de Informação

1581 Gestão de Serviços Financeiros e Bancários (esp.)

1707 Gestão e Avaliação de Tecnologias em Saúde (2.º Ciclo)

Associar

SGip

0000 Ano Zero (Economia)

0000 Ano Zero (FCT)

0000 Ano Zero (ISE)

1728 Ciências do Mar e do Ambiente (3.º Ciclo)

1315 Contabilidade Pública

Criar novo curso

Código Acad.	1754
Nome	Gestão de Informação
Unidade Orgânica	Escola Superior de Gestão, Hotelaria e Turismo
Pode ser criado?	Sim

Sem curso seleccionado

Figura 4.15 - Interface de Criação e Associação de Cursos.

A Figura 4.15 mostra a interface para a criação e associação de cursos, contendo duas listas, uma com os cursos importados da aplicação de gestão académica e a outra com os cursos que não estão resolvidos no SGIP. Por outro lado, a Figura 4.16 em cima, mostra um exemplo de criação de um curso importado.

1754 Gestão de Informação	0000 Ano Zero (ISE)
1581 Gestão de Serviços Financeiros e Bancários (esp.)	1728 Ciências do Mar e do Ambiente (3.º Ciclo)
1707 Gestão e Avaliação de Tecnologias em Saúde (2.º Ciclo)	1315 Contabilidade Pública

Criar novo curso

Criação de curso				
Cod. Acad.	Designação	Ano escolar	Período	Ação
1754	Gestão de Informação	Escola Superior de Gestão, Hotelaria e Turismo		
Disciplinas do curso				
1754C1004	Aplicações Web	1	1º semestre	Criar disciplina
1754C1008	Base de Dados	1	2º semestre	Criar disciplina
1754C1011	Desenvolvimento de Aplicações Móveis	1	2º semestre	Deixar como não resolvida
1754C1000	Desenvolvimento de Sistemas de Informação	1	1º semestre	Criar disciplina
1754C1007	Erp e Crm	1	2º semestre	Criar disciplina
1754C1010	Gestão da Inovação e Negócio Digital	1	2º semestre	Criar disciplina
1754C1009	Gestão de Processos	1	2º semestre	--- (escolha a operação) ---
1754C1001	Informática Para Organizações	1	1º semestre	--- (escolha a operação) ---
1754C1003	Noções Básicas de Programação	1	1º semestre	--- (escolha a operação) ---
1754C1005	Sistemas de Apoio à Decisão e Business Intelligence	1	1º semestre	--- (escolha a operação) ---
1754C1002	Tecnologia Multimédia	1	1º semestre	--- (escolha a operação) ---
1754C1006	Web Marketing e Redes Sociais	1	2º semestre	--- (escolha a operação) ---

1707 Gestão e Avaliação de Tecnologias em Saúde (2.º Ciclo)	0000 Línguas Aplicadas às Ciências Empresariais (2º Ciclo)
1506 História do Algarve (2.º Ciclo)	0000 Mestrado em Engenharia Civil
1596 Línguas Aplicadas às Ciências Empresariais (2.º Ciclo)	1525 Psicologia Social e das Organizações (2º Ciclo)
1738 Mecanismos de Doença e Medicina Regenerativa (3.º Ciclo)	1285 Segurança e Saúde no Trabalho

Criar novo curso

Associação de curso					
Cod. Acad.	Designação	Ano escolar	Período	Ação	Disc. Assoc.
1596	Línguas Aplicadas às Ciências Empresariais (2.º Ciclo)	Escola Superior de Gestão, Hotelaria e Turismo			
- associar com -					
0000	Línguas Aplicadas às Ciências Empresariais (2º Ciclo)	Escola Superior de Gestão, Hotelaria e Turismo			
Disciplinas do curso					
1596C1003	Análise Estatística Para as Ciências Sociais	1	1º Trimestre	Associar disciplina	0000 - Análise Estatística para as Ciências Sociais
1596C1007	Comunicação Intercultural	1	3º Trimestre	Associar disciplina	0000 - Comunicação Intercultural
1596C1019	Dissertação	2	Ano	Criar disciplina	
1596C1006	Espanhol Para Ciências Empresariais I	1	3º Trimestre	Associar disciplina	0000 - Espanhol I para Ciências Empresariais I
1596C1009	Espanhol Para Ciências Empresariais II	1	4º Trimestre	Associar disciplina	0000 - Espanhol I para Ciências Empresariais II
1596C1016	Estágio	2	Ano	Deixar como não resolvida	
1596C1005	Francês ou Alemão Para Ciências Empresariais I	1	2º Trimestre	--- (escolha a operação) ---	
1596C1008	Francês ou Alemão Para Ciências Empresariais II	1	3º Trimestre	--- (escolha a operação) ---	

Figura 4.16 – Em cima, exemplo de criação de curso. Em baixo, exemplo de associação de cursos.

Ao solicitar a criação de um curso, o operador é sempre confrontado com uma lista de disciplinas importadas, pertencentes ao curso em causa. Nessa lista, o operador é livre

de escolher individualmente entre a criação das disciplinas ou deixá-las como não resolvidas. Contudo, é obrigatório esta indicação explícita para todas as disciplinas para que seja possível efetivar a criação do curso.

Por outro lado, a associação é idêntica mas implica escolher um curso na lista de cursos oriundos da aplicação de gestão académica e o curso da lista de cursos não resolvidos do SGIP com o qual se pretende associar o primeiro. A operação inicia-se ao dar um clique no botão “Associar”, após o qual é mostrado uma lista das disciplinas do curso importado, tal como se mostra na Figura 4.16 em baixo.

No exemplo acima, deixa-se explícito que as disciplinas importadas poderão sofrer uma de três operações possíveis: ser deixada como não resolvida, ser criada ou ainda ser associada com alguma disciplina que pertença ao curso do SGIP a associar.

4.2.12.3. Interface de criação e associação de disciplinas

A interface de criação e associação de disciplinas é quase idêntica à interface de criação e associação de cursos. Contudo, existem duas diferenças substanciais entre as duas: na interface de disciplinas é necessário escolher o curso das disciplinas que são mostradas em cada uma das listas, tal como se mostra na Figura 4.17. Por outro lado, a operação de criação ou associação de disciplina não é detentora da complexidade apresentada para as mesmas operações nas interfaces de cursos.

Disciplinas: Resolução de problemas de importação

Escolha curso Serviços Académicos

1522 - Análises Clínicas e Saúde Pública (1.º Ciclo)

Serviços Académicos

1522C1082
Microbiologia

1522C1080
Oficina de Sociologia I

1522C1063
Seminário de Introdução à Escrita Científica

1522C1083
Técnicas de Laboratório e Segurança a

1522C1084
Terapia Génica e Celular

Código Acad.	1522C1063
Nome	Seminário de Introdução à Escrita Científica
Ano escolar	3
Tipo	Opcional
Período	1º Semestre
Responsável	-
Unidade Orgânica	Escola Superior de Saúde UALG

Escolha curso SGip

1522 - Análises Clínicas e Saúde Pública (1.º Ciclo)

SGip

Psicologia da Saúde - Análises Clínicas e Saúde Pública

1522C1054
Psicologia Social

1522C1055
Seminário

0000
Seminário de introdução à escrita científica

1522C1000
Seminários em Análises Clínicas e Saúde Pública

1522C1062
Sociologia da Saúde

Código Acad.	0000
Nome	Seminário de introdução à escrita científica
Ano escolar	3
Tipo	Opcional
Período	1º semestre
Responsável	Ana Patrícia Gago Mateus - apmateus
Unidade Orgânica	Escola Superior de Saúde UALG

Figura 4.17 - Interface de criação e associação de disciplinas.

4.2.12.4. Interface de anulação de operações sobre cursos

A interface de anulação de operações de criação e associação de cursos permite que o operador anule operações que ele, e apenas ele, executou sobre o curso. A anulação das operações sobre o curso poderá ser realizada ou não, dependendo de quais os relacionamentos que cada uma das disciplinas incluídas na operação sobre o curso tenha com outras entidades do SGIP.

As condicionantes à anulação sobre cada disciplina do curso são:

- A disciplina foi incluída numa DSD;
- A disciplina já faz parte de uma agregação de disciplinas;
- A disciplina já tem turnos gerados;
- A disciplina já tem áreas científicas associadas.

A existência de uma destas condicionantes em uma ou mais disciplinas do curso inibe a anulação da operação sobre o curso. Contudo, as restantes disciplinas podem ser alvo de anulações individuais na interface de anulação de operações de criação ou associação de disciplinas. Na Figura 4.18 ilustra-se uma anulação de um curso mal sucedida. O exemplo presente na figura demonstra uma operação de anulação que não pode ser desfeita porque, entretanto, uma das disciplinas criada já tinha sido associada a uma área científica, precisamente uma das operações inibidoras da anulação.

ANULAR OPERAÇÃO: Criação de Curso						
1735	Operações e Gestão de Spa	Escola Superior de Gestão, Hotelaria e Turismo				
Síntese da operação						
A operação não pode ser realizada por se tratar de uma operação desconhecida por existirem disciplinas criadas com impedimento! Esse impedimento pode tratar-se de: - A disciplina está associada a uma DSD; - A disciplina faz parte de uma agregação de disciplinas; - A disciplina já tem turnos gerados; - A disciplina está associada a áreas científicas.						
Disciplinas do curso						
Cod. Acad.	Designação	Ano escolar	Período	Operação	Pode anular?	Ação
1735C1003	Criação, Conceção e Aberturas de Spa	1	1º semestre	Criada	●	Apagar ▾
1735C1006	Gestão Operacional de Spa e Dinâmica Financeira	1	2º semestre	Criada	●	Apagar ▾
1735C1005	Marketing e Vendas	1	2º semestre	Criada	●	Apagar ▾
1735C1001	Noções Gerais de Saúde	1	1º semestre	Criada	●	Apagar ▾
1735C1007	Projeto de Investigação	1	2º semestre	Criada	●	Apagar ▾
1735C1004	Recursos Humanos	1	2º semestre	Criada	●	Apagar ▾
1735C1002	Serviços Spa e Medical Spa	1	1º semestre	Criada	●	Apagar ▾
1735C1000	Turismo de Saúde, Historia e Cultura dos Spa	1	1º semestre	Criada	●	Ignorar ▾

Figura 4.18 - Exemplo de anulação de curso criado.

4.2.12.5. Interface de anulação de operações sobre disciplinas

À semelhança do que foi escrito para a interface de criação e associação de disciplinas, a interface de anulação não se reveste da complexidade da interface equivalente das operações dos cursos. De uma forma muito geral, pode afirmar-se que as condições que possibilitam a anulação da operação sobre a disciplina são as mesmas que as indicadas para cada uma das disciplinas da anulação das operações sobre os cursos. A interface é idêntica pelo que não se entrará em mais detalhes.

4.3. Exploração dos dados

4.3.1. Consulta e impressão de Distribuição de Serviço Docente

4.3.1.1. Introdução

O SGIP disponibiliza um conjunto pequeno de consultas do serviço docente, mas que devem ser mencionadas pois foram pedidos explícitos da comunidade. Podem resumir-se essas consultas e respetiva impressão às seguintes:

- Consulta da DSD do grupo de docentes, conhecida simplesmente como DSD;
- Consulta da DSD do docente;
- Consulta da DSD do curso.

Para além destas consultas que incluem a opção de impressão integrada, foi ainda desenvolvida uma interface que permite fazer a impressão de várias DSD de grupo, sequencialmente, ou agrupando o serviço docente das DSD escolhidas e imprimir uma DSD resultante. Deve entender-se “impressão” num sentido mais lato, uma vez que o que é gerado são ficheiros em formato PDF que poderão ser impressos ou simplesmente consultados ou transportados.

A impressão pode ser considerada outra forma de exploração dos dados, sendo que se entende por impressão como a formatação de um documento PDF que permita ser transportado em formato digital ou, eventualmente, em formato papel. Este conceito de impressão tem em vista, e já tinha na época de desenvolvimento da aplicação, a

tendência de imprimir cada vez menos em papel e disseminar mais os documentos em formato digital.

O que até este momento foi designado simplesmente “DSD” poder-se-ia chamar “DSD de grupo de docentes” uma vez que se trata claramente da consulta e organização do serviço docente para um conjunto de docentes a que se chamou “Grupo de docentes”. Na verdade, trata-se de uma vista do serviço docente, importante por ser a que é utilizada na gestão do mesmo.

Contudo, a pedido de vários utilizadores, tornava-se demasiado importante deter a perspectiva do serviço docente do lado do docente ou mesmo do lado do curso. A essas vistas de serviço docente chamaram-se “DSD do docente” e “DSD do curso”, respetivamente. Essas duas vistas são muito importantes para consultar ou imprimir, o serviço docente de um único docente ou de um curso completo, respetivamente. Essas duas vistas serão detalhadas nos subcapítulos subsequentes.

Por fim, como forma de agilizar a impressão de serviço docente disperso em mais que uma DSD, foi desenvolvida uma forma de impressão sequencial de várias DSD ou ainda a composição de várias DSD em apenas uma, logo produzindo um único ficheiro PDF. Este tema será detalhado um pouco mais à frente no documento.

4.3.1.2. Distribuição de Serviço Docente do docente

A DSD do docente possibilita a consulta e impressão do serviço docente associado a um único docente. No SGIP, um docente pode estar presente em vários grupos de docente, o que torna a aplicação muito versátil no sentido em que cada grupo poderá representar qualquer associação de docentes que os elaboradores da DSD pretendam.

A Figura 4.19 apresenta um exemplo de uma DSD de docente. De notar que ela possui informação muito útil para o docente e para a gestão do serviço docente, com um sumário de horas em primeiro plano, bem como informação dos grupos de docente em que se encontra integrado e o detalhe de serviço docente e das outras atividades. Nestes detalhes, são identificados o tipo de escolaridade e a DSD em que o detalhe se encontra lançado, entre outras informações.

DSD do docente

Escolher o docente				
Docente:	<input type="text" value="Docente Dois (uiddois)"/>			
Sumário de actividade				
	Equivalente semanal		Equivalente anual	
Horas de contacto	6.58		197.5	
Outras actividades	6		180	
Total	12.58		377.5	
Grupos de docentes				
Grupo de Docentes de Matemática para RAP - Relatórios de Atividade Profissional				
Composição:	Docente Dois Docente Três Docente Um			
Grupo de Docentes de Teste I				
Composição:	Docente Dois Docente Um			
Detalhe das horas de contacto e restantes actividades dos docentes				
0000 - Testes Especificos I [Mestrado em Testes Exaustivos] - 1º semestre				
Aulas Teóricas	Turno 1	1.5 Hora(s)	DSD do Grupo de Docentes de Teste I	Coef. 1.0
Práticas Laboratoriais	Turno 1	4.5 Hora(s)	DSD do Grupo de Docentes de Teste I	Coef. 1.0
0000 - Testes Especificos II [Mestrado em Testes Exaustivos] - 2º semestre				
Aulas Teóricas	Turno 1	1.5 Hora(s)	DSD do Grupo de Docentes de Teste I	Coef. 1.0
Práticas Laboratoriais	Turno 1	4.5 Hora(s)	DSD do Grupo de Docentes de Teste I	Coef. 1.0
0000 - Testes Gerais III [Mestrado em Testes Exaustivos] - 1º semestre				
Trabalho de campo	Turno 1.2	5 Hora(s)	DSD do Grupo de Docentes de Teste I	Coef. 1.5
Trabalho de campo	Turno 1.1	10 Hora(s)	DSD do Grupo de Docentes de Teste I	Coef. 1.0
Outras actividades: Horas a Cargos				
Vice-Presidente Conselho Diretivo	6 Hora(s)	30 Semana(s)	DSD do Grupo de Docentes de Teste I	
Outras actividades: Orientações				
Sem actividades lançadas				
Outras actividades: Outras				
Sem actividades lançadas				

Figura 4.19 - Exemplo de DSD do docente.

Quando se desenvolveu a DSD do docente, a ideia era dar a possibilidade de cada docente aceder e consultar o seu serviço docente. Dado o término algo abrupto do desenvolvimento do SGIP, nunca foi terminada a programação necessária para tal.

4.3.1.3. Distribuição de Serviço Docente do curso

A DSD do curso reveste-se de uma complexidade que não foi colocada na DSD do docente. O público-alvo da DSD do curso era composto essencialmente pelos elaboradores da DSD que, desta forma, poderiam ter uma ideia clara do estado em que se encontrava o serviço docente desse curso. Nesse sentido, considerando que se tratava mais de uma página de gestão do que uma DSD de consulta, não foi desenvolvida a capacidade de impressão.

A Figura 4.20 ilustra uma DSD de curso. Poderão facilmente identificar-se várias zonas de informação e de ação, sendo composta pelas seguintes áreas:

1. Informação do plano do curso;
2. Informação das DSD de grupo envolvidas;
3. Detalhe dos turnos atribuídos.

DSD do curso					
Escolher o curso					
Entidade:	Instituto Superior de Engenharia				
Curso:	0000 - Mestrado em Testes Exaustivos				
Disciplinas do plano					
1º ano	1º semestre	0000	Contra-Testes I	3 turnos: 3 atribuídos, 0 livres	
		0000	Testes Específicos I	3 turnos: 2 atribuídos, 1 livres	
		0000	Testes Gerais I	2 turnos: 2 atribuídos, 0 livres	
	2º semestre	0000	Contra-Testes II	3 turnos: 0 atribuídos, 3 livres	
		0000	Testes Específicos II	3 turnos: 2 atribuídos, 1 livres	
		0000	Testes Gerais II	2 turnos: 2 atribuídos, 0 livres	
2º ano	1º semestre	0000	Dissertação I	0 turnos: 0 atribuídos, 0 livres	
		0000	Testes Específicos III	1 turnos: 0 atribuídos, 1 livres	
		0000	Testes Gerais III	2 turnos: 0 atribuídos, 2 livres	
	2º semestre	0000	Dissertação II	0 turnos: 0 atribuídos, 0 livres	
		0000	Testes Específicos IV	1 turnos: 0 atribuídos, 1 livres	
		0000	Testes Gerais IV	1 turnos: 0 atribuídos, 1 livres	
Distribuições de Serviço Docente do plano					
DSD do Grupo de Docentes de Teste I		Em elaboração		Grupo de Docentes de Teste I	ISE
DSD do Grupo de Docentes de Teste II		Em elaboração		Grupo de Docentes de Teste II	ISE
Contra-Testes I (0000)					
Horas de contacto					
Aulas Teóricas	Turno 1	3 Hora(s)	Docente Três	(DSD: DSD do Grupo de Docentes de Teste II)	
Aulas Práticas	Turno 1	4.5 Hora(s)	Docente Três	(DSD: DSD do Grupo de Docentes de Teste II)	
Aulas Práticas	Turno 2	4.5 Hora(s)	Docente Três	(DSD: DSD do Grupo de Docentes de Teste II)	
Testes Específicos I (0000)					
Horas de contacto					
Aulas Teóricas	Turno 1	1.5 Hora(s)	Docente Dois	(DSD: DSD do Grupo de Docentes de Teste II)	
Práticas Laboratoriais	Turno 1	4.5 Hora(s)	Docente Dois	(DSD: DSD do Grupo de Docentes de Teste II)	
Práticas Laboratoriais	Turno 2	4.5 Hora(s)	-	(turno não atribuído)	

Figura 4.20 - Exemplo de DSD de curso.

A área de informação do plano do curso apresenta essencialmente o currículo do curso com indicação por disciplina dos turnos criados, dos turnos atribuídos e dos turnos livres. Tem ainda a possibilidade de, caso o perfil do utilizador o permita, editar os turnos da disciplina a partir da interface e possui um código de cores que indica o estado de turnos. Estas cores poderão ser:

- **Verde:** caso existam turnos criados e todos se encontrem atribuídos;
- **Laranja:** caso existam turnos criados e apenas alguns se encontrem atribuídos;
- **Vermelho:** caso existam turnos criados, mas nenhum turno esteja atribuído;
- **Preto:** caso a disciplina não tenha turnos criados, o que poderá significar que a disciplina não funcionará ou ainda não tem serviço docente atribuído.

A área de informação das DSD de grupo de docentes que estão na base da DSD de curso permite consultá-las pela sua designação e ainda ter acesso a uma de 3 tarefas, sendo apenas serão acessíveis se o utilizador tiver permissões para tal:

- Visualizar a DSD;
- Imprimir a DSD;
- Editar a DSD.

A área de detalhe possibilita, disciplina a disciplina, visualizar os turnos atribuídos a cada docente, sendo fornecida alguma informação adicional como, por exemplo, o número de horas, o tipo de horas de contacto e o docente a quem está atribuído o turno.

4.3.1.4. Impressão de DSD composta

A ideia de impressão conjunta das DSD foi desenvolvida pela preocupação de alguns utilizadores ao referir a dispersão de ficheiros das várias distribuições de docente de grupos de docente. Quer pela necessidade de impressão individual, quer pela ausência de uma forma agrupada de ver o serviço docente, as opções de composição da impressão de várias DSD de grupo de docente são uma forma ágil de juntar as atribuições de turnos a docentes.

Gestão da Distribuição de Serviço Docente - Impressão múltipla							
* Filtros							
Lista de DSD							
Designação	Ano lectivo	Período	Grupo docentes	Estado	Consultar	Relatórios	Ações
DSD do Grupo de Docentes de Teste II	2015/2016	Ano	Grupo de Docentes de Teste II	Em elaboração			<input checked="" type="checkbox"/>
DSD do Grupo de Docentes de Teste I	2015/2016	Ano	Grupo de Docentes de Teste I	Em elaboração			<input checked="" type="checkbox"/>
Curso Técnico Superior Profissional (2015/2016)	2015/2016	Ano	Curso Técnico Superior Profissional (2015/2016)	Em elaboração			<input type="checkbox"/>
3 registos							

Imprimir DSDs sequencialmente |
 Imprimir DSD resultante

Figura 4.21 - Interface de impressão de DSD composta.

A Figura 4.21 ilustra a interface de impressão de DSD composta, onde se podem identificar as duas formas diferentes de impressão de DSD composta:

- Impressão sequencial de DSD;
- Impressão de DSD resultante.

Como o próprio nome indica, a impressão sequencial de DSD permite a junção de várias DSD de grupo de docentes num único documento de impressão. Por outro lado, a impressão da DSD resultante permite misturar o serviço docente de várias DSD como se tratasse apenas de uma, num único documento.

4.3.2. Web Services de leitura

4.3.2.1. Introdução

Existem duas formas principais de disponibilizar informação por *web services*: utilizando SOAP (SOAP, 2016) ou REST (Fielding, 2010). O último foi desenvolvido numa tese de doutoramento de Roy Thomas Fielding. Sem pretender apresentar demasiados detalhes sobre os dois uma vez que sairá do âmbito do presente documento, poder-se-á afirmar de uma forma ligeira que o primeiro se trata de um protocolo e o segundo é um estilo de arquitetura de software.

Com o objetivo de criar uma forma fácil de explorar os dados do SGIP, foram criados os *web services* de leitura, módulo ao qual se deu o nome de SGIPWS. Este módulo disponibiliza métodos de leitura em SOAP para consultar informação presente no SGIP.

Numa perspetiva histórica, os *web services* surgiram no contexto de uma necessidade de ter informação dos tempos de docência na implementação da aplicação integrada de recursos humanos e financeiros SIGEST, sobre uma plataforma SAP (SAP, 2016). A proposta inicial da equipa de implementação era a transferência por ficheiros pelo que a equipa do SGIP contrapôs com a necessidade de que essas transferências de informação se fizessem da forma mais automatizada e direta possível. Estes métodos serão doravante referidos como métodos desenvolvidos na 1ª fase.

Cerca de ano e meio mais tarde, tendo em conta a atualização do Moodle⁷ eminente e a vontade da equipa em não alimentar os dados da aplicação com injeção direta às tabelas da base de dados, resolveu-se desenvolver métodos que permitissem a integração dos

⁷ <http://www.moodlelivre.com.br/tutoriais-e-dicas/974-o-que-e-moodle>

dados do SGIP na aplicação de Tutoria Eletrónica. Desta forma, seria possível desenvolver um módulo de integração que lesse dados do SGIP por *web service* e os integrasse no Moodle pelos *web services* dessa aplicação, entretanto desenvolvidos para a sua versão 2. Estes serão referidos como métodos desenvolvidos na 2ª fase.

A decisão da equipa desenvolver em cima de SOAP teve várias razões, mesmo sabendo do formalismo quase excessivo que envolve o protocolo quando comparado com as implementações em REST. Basicamente, as razões resumem-se à necessidade de desenvolver os *web services* de forma célere, algum conhecimento que existia no seio da equipa, apesar da completa inexperiência em relação ao tema, e ainda pela quantidade de informação disponível.

4.3.2.2. Métodos desenvolvidos – 1ª fase

Inicialmente o SIGEST necessitava de informação sobre os planos de estudo e sobre o serviço docente. Para poder fornecer essa informação, foram desenvolvidos quatro métodos:

- Método **docentes**: informação sobre um ou todos os docentes presentes no SGIP;
- Método **planos**: informação sobre os planos de estudos completos de uma ou de todas as unidades orgânicas e de um determinado ano letivo ou o mais recente;
- Método **agregadas**: informação sobre a agregação de disciplinas de uma ou de todas as unidades orgânicas e de um determinado ano letivo ou o mais recente;
- Método **sdocente**: informação sobre o serviço docente de uma ou de todas as unidades orgânicas e de um determinado ano letivo.

Claramente o primeiro método apresentado, *docentes*, não era necessário para a implementação em causa, tendo serviço como prova de conceito pela sua baixa complexidade de consulta à base de dados. Por outro lado, o método *agregadas* foi desenvolvido inicialmente como complemento da informação do método *planos* que não tinha qualquer informação sobre a agregação de disciplinas. Contudo, esta forma não era funcional para a alimentação do SAP pelo que se optou por incluir toda a informação no método *sdocente*.

Em relação à implementação dos métodos, foram impostas algumas características para os tornar mais seguros e garantir um acesso correto à informação. Entre essas características poderão destacar-se:

- Disponibilização do *web service* sobre SSL, em https;
- Autenticação integrada no domínio UALG com *username* e *password* para todos os métodos;
- Autorização explícita, acesso apenas aos utilizadores explicitamente autorizados;
- SOAP Server desenvolvido em PHP, tal como o SGIP.

Apresenta-se em ANEXO VIII o documento desenvolvido aquando do término da primeira fase de desenvolvimento dos *web services*.

4.3.2.3. *Desempenho e sistema de cache*

Alguns dos métodos desenvolvidos envolviam um elevado processamento de consulta e um enorme volume de dados. Tais características resultavam num tempo de resposta péssimo, usualmente acima dos 10 minutos, tendo em conta o hardware disponível na altura no *datacenter* da UAlg. Este valor tornava-se inaceitável uma vez que o SAP apenas dispunha de um *timeout* máximo de 5 minutos.

Apesar dos esforços na otimização das consultas, continuaram-se a obter tempos na ordem dos minutos. Assim, na impossibilidade de adquirir e implementar sistemas de maior disponibilidade, restava a construção de um mecanismo de cache que permitisse reduzir significativamente os tempos de resposta. Este sistema foi construído com as seguintes premissas:

- Tabelas distintas para cada método;
- Tabelas de cada método alimentadas com a consulta universal, sem filtros, respeitante ao método;
- Duas tabelas para cada método, uma para leitura e outra para alimentação, para evitar *locks* e diminuir a possibilidade de fornecimento de informação incoerente, com alternância da função nos ciclos de alimentação;
- Tabela de sistema para controlo das tabelas de leitura ativas para cada método;

- Alimentação das tabelas de cache controlada por um script CLI para permitir a sua invocação na linha de comandos;
- Temporização da alimentação das tabelas de cache controlada externamente no *cron* do sistema operativo.

Após a implementação do sistema, passou-se a ter tempos de resposta de apenas alguns segundos, que justificou plenamente a aposta.

4.3.2.4. Métodos desenvolvidos – 2ª fase

Quando se começou a preparar a atualização do Moodle, bastante desatualizado em versão 1.5, para a versão 2 surgiu a necessidade de realizar a alimentação do mesmo de outra forma que não a implementada. Esta recorria a consultas de ação diretas à base de dados que poderiam – e eventualmente deveriam – deixar de funcionar com quaisquer alterações de versão que pudessem modificar a lógica do modelo de dados. Além disso, não é de todo uma prática aconselhada pois a equipa não detinha um profundo conhecimento dos meandros do modelo de dados.

Nessa altura decidiu-se apostar na alimentação da informação de base da tutoria eletrónica através da utilização de *web services*. Este desenvolvimento tirou proveito do conhecimento entretanto obtido pelo desenvolvimento dos métodos especificados anteriormente, tendo sido desenvolvidos mais métodos do mesmo serviço. Tal simplificava o desenvolvimento e a colocação em produção dos mesmos e permitia ainda juntar num mesmo serviço um conjunto de métodos bastante completo de exploração da informação de serviço docente.

Como a informação que alimenta a tutoria tem como base o serviço docente, toda a lógica da publicação desta informação se baseia no serviço docente. Os métodos desenvolvidos foram:

- Método **dsdsHomologadas**: permite consultar as DSD homologadas de um determinado ano letivo, última versão por defeito e, opcionalmente, cuja homologação tenha sido realizada após uma data;

- Método **unidadesDsd**: permite consultar todas as unidades orgânicas que estejam envolvidas nas DSD especificadas;
- Método **cursosDsd**: permite consultar todos os cursos que estão envolvidos nas DSD especificadas;
- Método **disciplinasDsd**: permite consultar os registos de todas as disciplinas envolvidas nas DSD especificadas;
- Método **docentesDsd**: permite consultar os registos de todos os docentes que estão envolvidos nas DSD especificadas;
- Método **inscricaoDocentesDsd**: permite consultar os registos de todas as associações de docentes a disciplinas envolvidos nas DSD especificadas;
- Método **alunosDsd**: permite consultar os registos de todos os alunos abrangidos pelas DSD especificadas;
- Método **inscricaoAlunosDsd**: permite consultar os registos de todas as inscrições de alunos a disciplinas envolvidas nas DSD especificadas.

Deve esclarecer-se que ainda foram desenvolvidos alguns outros métodos mas que não foram da responsabilidade do autor. Como quase sempre acontece, estes outros métodos foram a resposta à necessidade de informação criadas por outras pessoas ou plataformas.

4.4. Trabalhos inacabados ou não iniciados

4.4.1. Gestão de espaços

Poder-se-á encarar a gestão de espaços como o segundo capítulo de um trabalho completo que deveria ser conhecido como SGIP. Este segundo capítulo pretendia dotar a UAlg da capacidade de gerir ativamente e de forma descentralizada os espaços que compõem os seus campi. De uma forma grosseira, poder-se-á afirmar que esses espaços podem ser de exploração privada por parte de cada unidade orgânica, por exemplo salas ou gabinetes dos edifícios de funcionamento das unidades orgânicas, ou espaços partilhados, como, por exemplo, os complexos pedagógicos.

A gestão de espaços deveria ter um conjunto de características das quais se destacam as principais:

- Capacidade de gestão dos espaços partilhados;
- Identificação dos vários tipos de espaço;
- Enquadramento geográfico do espaço;
- Definição das características essenciais do espaço, dependendo do tipo de espaço;
- Definição dos equipamentos integrantes dos vários espaços;
- Capacidade de gestão de atribuição dos espaços, através de marcações e requisições;
- Capacidade de gestão de manutenção das salas;
- Integração com o sistema de permissões.

Estas características deveriam ter uma integração plena com o sistema de permissões, conforme indicado. Dessa forma, seria possível a definição de vários perfis de utilizadores do módulo de espaços e a definição de âmbitos à entidade / unidade orgânica.

4.4.2. Agendas

A ideia inicial de agenda baseava-se no objetivo de poder dotar o docente da capacidade de gerir as suas obrigações pedagógicas. A marcação de aulas estaria sujeita ao definido por um horário, que será abordado mais à frente no presente documento, sendo que este define um padrão usualmente semanal de ocorrência de tempos de contacto com os alunos ao longo de um determinado período.

A disponibilização de uma agenda ao docente permitiria que este fizesse a gestão de alterações dos tempos de contacto, bem como a definição de horas de atendimento, marcação de eventos especiais como testes ou outros, definição de datas limite para entrega de trabalhos, gestão de alterações de aulas ou ainda outro tipo de operações usuais.

Intimamente ligada com esta ideia inicial de agenda do docente estava a certeza que a existência desta implicaria a necessidade de implementação de outras duas agendas: a agenda do aluno e a agenda do espaço. É a consequência natural do facto de as aulas e eventos pedagógicos dos turnos implicarem com os alunos inscritos nos turnos e ainda o facto de qualquer evento se desenrolar num espaço específico para tal.

A agenda do aluno seria baseada na definição das suas inscrições a disciplinas e respetivas inscrições aos turnos. Esta agenda permitiria que o aluno se inscrevesse em turnos de disciplinas onde ainda não o estava, sempre que existisse essa possibilidade, e também ter os seus eventos registados, caso o pretendesse.

A agenda do espaço refletiria a ocupação temporal do mesmo. Com esta agenda seria possível marcar ou fazer o pedido de marcação do espaço, bem como consultar o responsável por cada período de ocupação e qual o evento ao qual está destinado.

4.4.3. Horários e sumários

Os horários são, no fundo, a distribuição das horas de contacto das disciplinas numa frequência usualmente semanal e por um determinado período de tempo, como o ano letivo ou o semestre. Esta distribuição temporal despoleta a marcação em agenda dos turnos a que dizem respeito nos dias e horas definidos, nas semanas do período a que dizem respeito.

Uma outra característica dos horários é o facto de serem sujeitos a muitas mudanças e cada uma delas deverá ser aprovada pelos responsáveis da unidade orgânica a que dizem respeito as turmas dos cursos respetivos. Esta característica leva a que se deva considerar o horário como um objetivo versionável, com toda a gestão inerente a tal.

Por outro lado, o sumário é no fundo a formalização da ocorrência de uma aula, isto é, de um evento previsto pelo horário e concretizado nas respetivas agendas de docente, aluno e espaço. O sumário deveria ser introduzido pelo docente respetivo no evento correspondente da agenda e deverá ser constituído pela descrição dos conteúdos da aula e ainda pela respetiva assiduidade.

5. Conclusão

O meio universitário proporciona, ou deveria proporcionar, um ambiente favorável ao desenvolvimento de projetos de software entre portas. O nível de desenvolvimento poderá ser variável de acordo com o que se considere ideal para a organização tendo em conta a sua dimensão e a sua capacidade de alocar recursos, sejam eles humanos ou outros recursos físicos.

A necessidade de desenvolver aplicações poderá ter várias justificações que vão desde a incapacidade do mercado de oferecer um par aplicação-serviços com uma relação custo-benefício que a organização considere vantajosa ou simplesmente pela inexistência de uma peça de software no mercado que satisfaça do ponto de vista técnico ou de especificações. Seja como for, a aposta no desenvolvimento no interior da organização torna-se mais efetiva quanto maior for o período de aposta nessa filosofia.

O SGIP foi um projeto que já teve o seu término, tendo estado em produção desde o ano letivo de 2008/2009 até ao ano letivo de 2014/2015. Inicialmente encarado com alguma apreensão por ser algo intrusivo na vida das unidades orgânicas por interferir diretamente no seu trabalho, a verdade é que foi cimentando a sua posição e demonstrou a sua utilidade através do fornecimento de informação para alimentar a tutoria eletrónica, em primeira instância, fornecendo também informação de base para o SIGEST e para o PEAAD.

Na realidade, pode comprovar-se a sua importância na organização porque a dotou de um sistema e de conceitos universais que serviram de base ao início da implementação do atual sistema de gestão académica e pedagógica. É inegável que, no início da implementação do SiGES, os utilizadores já possuíam o léxico que permitiu a todos falar fluidamente e de forma compreensiva de questões de serviço docente.

A verdade é que das três principais vertentes planeadas inicialmente – a gestão do serviço docente, a gestão de espaços e a gestão de agendas e sumários – apenas uma delas entrou em produção, a gestão de serviço docente. Não importa dirimir as razões nem tecer quaisquer considerações, importa deixar claro que teve um grande número de aspetos positivos que importam relevar, apesar de ser importante esclarecer sobre as principais dificuldades encontradas pela equipa.

Considerando, então, apenas a componente que entrou em produção, podem apontar-se alguns pontos fracos da aplicação. Os principais pontos fracos, na perspetiva do autor, são:

- **Usabilidade:** Interface com muito trabalho de usabilidade por fazer, apesar da equipa ter bastantes planos e ideias para o realizar.
- **Erros:** Resolução dos poucos erros que persistiram, exigiam uma resolução com grande investimento por parte da equipa, dada a complexidade da aplicação.
- **Desempenho:** O desempenho não nunca foi um ponto forte da versão 1 da *framework* utilizada e tal refletia-se na aplicação. O *upgrade* para a versão 2 implicava a reescrita de uma grande parte da aplicação.
- **Dashboard:** Não foi construído nenhum *dashboard* que possibilitasse a consulta de indicadores de gestão.

Contudo, também se pode dizer que o SGIP teve pontos positivos importantes, que se passam a descrever:

- **Uniformização:** O SGIP permitiu a uniformização dos conceitos em torno do Serviço Docente, possibilitando que a organização falasse a mesma linguagem e operasse da mesma forma. O desenvolvimento do SGIP ainda teve em consideração alguns aspetos particulares de unidades orgânicas, transformando-os em opções disponíveis a toda a comunidade de utilizadores.
- **Automatização:** O SGIP permitiu automatizar um conjunto de procedimentos que chegaram a ser realizados de forma manual ou por meios desconexos ou *ad-hoc*, como, por exemplo, a criação dos cursos e disciplinas na tutoria.
- **Disponibilidade:** O SGIP permitiu a recolha de um conjunto de dados que importam à gestão da organização, cuja utilidade já foi comprovada.
- **Integração:** O SGIP permitiu iniciar uma cultura de integração de aplicações na Universidade. Pelo facto de possuir uma posição charneira na junção dos dados pedagógicos, foi possível integrar com outras aplicações, como por exemplo, a aplicação de recolha de questionários das perceções de aprendizagem para alunos e docentes (PEAAD).

Poderá dizer-se que, na perspetiva do autor e apesar de todas as dificuldades e contratempus e também depois de muito trabalho, o SGIP valeu a pena. A introdução de uma cultura nova, partilhada, no que diz respeito ao serviço docente permitiu que a nova aplicação de gestão académica e pedagógica tivesse um início mais fácil em relação a essa matéria. A existência de dados normalizados e confiáveis no que ao serviço docente diz respeito fornece uma fonte de informação de grande valor aos decisores, sendo que poderão tomar-se algumas decisões com base na mesma.

Por realizar num futuro, na perspetiva do autor, ainda fica:

- Terminar a gestão de espaços;
- Desenvolver toda a agenda e sumários;
- Melhorar a interface com o utilizador;
- Passar para a versão mais recente do *framework*;
- Realizar *refactoring* profundo ao código;
- Rever o modelo de dados em alguns pormenores.

Provavelmente estas ações nunca serão realizadas mas fica, na perspetiva do autor, o legado do SGIP, com todos os seus defeitos e virtudes.

Referências

- Agile (2016) Manifesto for Agile Software Development, <http://agilemanifesto.org/>, acessado em 2016/06/01.
- ASP (2016) Microsoft ASP.NET, <http://www.asp.net/>, acessado em 2016/05/19
- Atom (2016) ATOM a Hackable Text Editor, <https://atom.io/>, acessado em 2016/05/22
- Bazaar (2016) Bazaar, <http://bazaar.canonical.com/en/>, acessado em 2016/05/27
- Beck, K. (2000) Extreme programming explained: embrace change. Addison-Wesley Professional. ISBN-13: 978-0321278654
- Boehm, B. (1986) A spiral model of software development and enhancement. ACM SIGSOFT Software Engineering Notes, 11(4), 14-24.
- CakePHP (2016) CakePHP - The rapid development PHP framework, <http://cakephp.org/>, acessado em 2016/05/19
- CapGemini (2016) CapGemini, Consulting Technology Outsourcing, <https://www.pt.capgemini.com/>, acessado em 2016/05/26
- Chili (2016) Chili Project, <https://www.chiliproject.org/>, acessado em 2016/05/26
- Chrome (2016) Chrome, Use um navegador da Web gratuito e mais rápido, <https://www.google.com/chrome/>, acessado em 2016/06/05
- CVS (2016) CVS - Concurrent Versions System, <http://cvs.nongnu.org/>, acessado em 2016/05/27
- Desmond, B., Richards, J., Allen, R., & Lowe-Norris, A. G. (2013) Active Directory: Designing, Deploying, and Running Active Directory. O'Reilly Media (5th ed.)
- Eclipse (2016) The Eclipse Foundation, <https://eclipse.org/>, acessado em 2016/05/22
- FenixEdu (2016) FenixEdu, <http://fenixedu.org/>, acessado em 2016/06/13
- Fielding, R. T. (2000) Architectural Styles and the Design of Network-based Software Architectures, <https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>, acessado em 2016/05/13
- Firefox (2016) Mozilla Firefox, <https://www.mozilla.org/pt-PT/firefox/products/>, acessado em 2016/06/05
- GIT (2016) GIT - Fast Version Control, <https://git-scm.com/>, acessado em 2016/05/27
- Gosling, J. (2000) The Java language specification. Addison-Wesley Professional.
- HTTPD (2016) Apache HTTP Server Project, <https://httpd.apache.org/>, acessado em 2016/05/19

IIS (2016) Microsoft Internet Information Server, <http://www.iis.net/>, acessado em 2016/05/19

Infinite (2016) Infinite Solutions, <http://www.infinite.pt/>, acessado em 2016/05/19

Jablonski, S., Petrov, I., Meiler, C., & Mayer, U. (2004) Guide to web application and platform architectures, Guide to web application and platform architectures, pp. 121-148. Heidelberg, Springer.

King, E. (2011) Automated Database Refresh in Very Large and Highly Replicated Environments. Regis University.

Lacey, M. (2012) The Scrum Field Guide: practical advice for your first year. Addison-Wesley Professional (1st ed.). ISBN-13: 978-0321554154

Lacey, M. (2016) Mitch Lacey & Associates, Inc. – Scrum Framework Flow Diagram, <https://www.mitchlacey.com/resources/scrum-framework-flow-diagram>, acessado em 2016/06/17

Laravel (2016), Laravel – The PHP Framework For Web Artisans, <https://laravel.com/>, acessado em 2016/06/14

LIGHTTPD (2016) LIGHTTPD Fly Light, <https://www.lighttpd.net/>, acessado em 2016/05/19

MariaDB (2016) MariaDB Foundation, <https://mariadb.org/>, acessado em 2016/06/13

Mercurial (2016) Mercurial SCM, <https://www.mercurial-scm.org/>, acessado em 2016/05/27

McNurlin, B.C. & Sprague, R.H. (2005) Sprague, Information Systems Management in Practice (7 ed.), Pearson Prentice Hall

MSIE (2016) Microsoft Internet Explorer, <http://windows.microsoft.com/pt-pt/internet-explorer>, acessado em 2016/06/05

MySQL (2016) MySQL - The world's most popular open source database, <https://www.mysql.com/>, acessado em 2016/05/19

MySQLWB (2016) MySQL Workbench, <https://www.mysql.com/products/workbench/>, acessado em 2016/06/05

NetBeans (2016) NetBeans IDE, Fits the Pieces Together, <https://netbeans.org/>, acessado em 2016/05/22

NGINX (2016) NGINX – High Performance Load Balancer, Web Server & Reverse Proxy, <https://www.nginx.com/>, acessado em 2016/05/19

NoSQL (2016) NoSQL Databases, <http://nosql-database.org/>, acessado em 2016/06/19

OpenProject (2016) OpenProject – Open source project management software, <https://www.openproject.org/>, acessado em 2016/05/26

Opera (2016) Opera, <http://www.opera.com/>, acessado em 2016/06/05

Oracle (2016) Oracle Database, <https://www.oracle.com/database/index.html>, acessado em 2016/05/20

Otte, S. (2009) Version Control Systems. Computer Systems and Telematics, Institute of Computer Science, Freie Universität, Berlin, Germany

PHP (2016) Hypertext Preprocessor, <http://php.net/>, acessado em 2016/05/19

Planner (2016) Gnome Planner, <https://goo.gl/D3eG1Q>, acessado em 2016/05/27

PostgreSQL (2016) PostgreSQL - The world's most advanced open source database, <http://www.postgresql.org/>, acessado em 2016/05/19

Project (2016) Microsoft Project, <https://products.office.com/pt-pt/project/project-and-portfolio-management-software>, acessado em 2016/05/27

qubEDU (2016) qubEDU Plataforma de Gestão Académica, <http://www.qub-it.com/solutions/qubEdu>, acessado em 2016/06/13

Rails (2016) RAILS - Imagine what you could build if you learned Ruby on Rails..., <http://rubyonrails.org/>, acessado em 2016/05/19

Ramakrishnan, R., & Gehrke, J. (2000) Database management systems. McGraw-Hill.

Redmine (2016) Overview - Redmine, <http://www.redmine.org/>, acessado em 2016/05/26

Royce, W. W. (1970) Managing the development of large software systems. In Proc. of IEEE WESCON, 26(8), 1-9.

Ruby (2016) Linguagem de Programação Ruby, <http://www.ruby-lang.org/pt/>, acessado em 2016/05/19

Safari (2016) OS X - Safari - Apple, <http://www.apple.com/safari/>, acessado em 2016/06/05

SAP (2016) SAP Software Solutions, <http://go.sap.com/index.html>, acessado em 2016/06/17

Schwaber, K. (1997) Scrum development process. In Business Object Design and Implementation, pp. 117-134. Springer London.

Schwaber, K. (2014) What is Scrum, http://www.volaroint.com/wp-content/uploads/dlm_uploads/2014/03/DC-VOLARO-Training-Scrum-What_Is_Scrum.pdf, acessado em 2016/06/17

Schwaber, K., & Sutherland, J. (2011). The Scrum guide. Scrum Alliance.

Schwaber, K., Sutherland, J., & Beedle, M. (2013) The definitive guide to Scrum: The rules of the game. <http://www.scrumguides.org/docs/scrumguide/v1/scrum-guide-us.pdf>, acedido em 2016/06/17

SIGARRA (2016) SIGARRA - Gestão Académica Integrada, <http://www.sysnovare.pt/portal/sigarra>, acedido em 2016/05/13

SIGES (2016) Somos Digitalis, <http://www.digitalis.pt/>, acedido em 2016/05/14

SOPHIA (2016) Gestão Académica - Sophia, <http://www.normatica.pt/#!/gesto-acadmica---sophia/c5gt>, acedido em 2016/05/14

SQLServer (2016) SQL Server 2014, <https://www.microsoft.com/pt-pt/server-cloud/products/sql-server/overview.aspx>, acedido em 2016/05/20

Sublime (2016) Sublime Text, <https://www.sublimetext.com/>, acedido em 2016/05/22

SVN (2016) Apache Subversion, <http://subversion.apache.org/>, acedido em 2016/05/27

Symfony (2016) Symfony - High Performance PHP Framework for Web Development, <http://symfony.com/>, acedido em 2016/05/19

Takeuchi, H., & Nonaka, I. (1986) The new new product development game. Harvard business review, 64(1), 137-146.

Thai, T., & Lam, H. (2003) *.NET framework essentials*. O'Reilly Media, Inc.

Trac (2016) The Trac Project, <https://trac.edgewall.org/>, acedido em 2016/05/26

VBVCS (2016) Visual Basic and Visual C#, <https://goo.gl/IkoVGP>, acedido em 2016/05/19

VisualStudio (2016) Visual Studio – Microsoft, <https://www.visualstudio.com/>, acedido em 2016/05/22

VSCoDe (2016) Visual Studio Code – Text Editing. Redefined, <https://code.visualstudio.com/>, acedido em 2016/05/22

VSVC (2016) Version Control, Visual Studio Team Services, <https://www.visualstudio.com/version-control-vs>, acedido em 2016/05/27

WireMaze (2016) WireMaze, <http://www.wiremaze.com/>, acedido em 2016/05/27

YAML (2016) YAML Ain't Markup Language, <http://yaml.org>, acedido em 2016/06/11

Zaninotto, F., & Potencier, F. (2007) The definitive guide to Symfony. Apress

Zend (2016) Zend Framework, <http://framework.zend.com/>, acedido em 2016/06/16

Zervaas, Q. (2007) Practical Web 2.0 Applications with PHP. Apress

Anexos

Todos os anexos são disponibilizados em formato digital, uma vez que são ou despachos reitorais ou documentos com um número elevado de páginas.

ANEXO I. Despacho Reitoral RT.08/2007

ANEXO II. Documentos da Análise da Informação Pedagógica

ANEXO III. Despacho Reitoral RT.1/08

ANEXO IV. Product Backlog

ANEXO V. Despacho Reitoral RT.008/2012

ANEXO VI. Ficheiro configuração abertura ano letivo 2012/2013

ANEXO VII. Procedimento para abertura de um novo ano letivo

ANEXO VIII. SGIP Web Services

ANEXO IX. Curriculum Vitae do autor