

Utilização de padrões de desenho de software no desenvolvimento e de Sistemas de Informação Geográfica

O caso da Região Demarcada do Douro

Nelson Rodrigues¹, Lino Oliveira¹, Lúgia Silva¹, António Coelho^{1,2}.

1) Instituto de Engenharia de Sistemas e Computadores do Porto, Porto, Portugal

{nelson.b.rodrigues, lino.oliveira, ligia.silva, antonio.coelho}@inescporto.pt

2) Departamento de Engenharia Informática, Faculdade de Engenharia, Universidade do Porto, Porto, Portugal

Resumo

A Região Demarcada do Douro (RDD) possui características únicas a nível mundial e requer um sistema de classificação das parcelas com vinha que assegure os níveis de elevada qualidade dos vinhos do Douro e Porto. Com base neste pressuposto, foi desenvolvido um Sistema de Informação Geográfica para a gestão e classificação das parcelas com vinha da RDD. O sistema foi desenhado utilizando padrões de desenho de software com o objectivo de tornar a sua evolução sustentável, respondendo aos novos requisitos e preservando o legado inicial. Neste artigo é apresentada uma metodologia que permite a utilização de Rich Internet Application (RIA) sobre as aplicações Web tradicionais, proporcionando interfaces gráficas (GUI) mais ricas, dinâmicas e interactivas, tornando as aplicações mais atractivas para o utilizador.

É apresentado ainda um padrão de desenho de software, derivado do Model –View-Controller (MVC) designado por Model – View – LogicGateway (MVLG). É um padrão de arquitectura de software utilizado no desenvolvimento de User Interfaces (UI), que permite a separação lógica e física do código fonte, proporcionando a sua reutilização e facilitando a correcção de erros, tornando o código mais limpo e eficiente.

Dado a natureza da informação geográfica da parcela com vinha foi desenvolvido um componente modular, baseado em padrões de software, que permite a visualização de informação geográfica, utilizando serviços de *Web mapping* sem custos, provando ser uma alternativa às soluções comerciais.

Palavras-chave: Integração de Sistemas de Informação, Sistema de Gestão de Bases de Dados, Metodologia de Desenvolvimento de Software, Sistemas de Informação Geográfica

1. Introdução

A atribuição da denominação de origem na produção vitícola nacional é definida por um conjunto de leis regulamentadas por várias instituições. A Região Demarcada do Douro (RDD) responsável pela produção do Vinho do Porto possui características únicas, tornando-o num dos vinhos mais famosos do mundo. O Instituto dos Vinhos do Douro e Porto (IVDP) é a entidade responsável pela atribuição das Denominações de Origem Controlada (DOC) na RDD: "Porto" e "Douro".

O sistema de informação desenvolvido permite ao IVDP a classificação e cadastro das parcelas com vinha, tendo em conta as especificidades da RDD e o trabalho prévio efectuado na normalização de parcela com vinha [Coelho et al. 2011].

A aplicação foi desenvolvida assente numa arquitectura modular, baseada em componentes isolados, utilizando sempre que possível padrões de desenho de software. Toda a arquitectura foi projectada com os objectivos da reutilização de componentes e da escalabilidade do sistema. Inicialmente, foi implementado o módulo responsável pelo cadastro dos *Boletins de Campo* utilizando tecnologias Web, mais propriamente AJAX.

Neste artigo é proposta uma arquitectura modular que pode ser utilizada no desenvolvimento sistemas de informação com componente geográfica. É descrita ainda a evolução do sistema, demonstrando a importância dos padrões de desenho de software, acrescentado novas funcionalidades respeitando o legado do sistema inicial.

A título exploratório foi estudada a potencialidade das Rich Internet Applications (RIA) no desenvolvimento de interfaces gráficas, tendo como *focus* principal o dinamismo, a interactividade, a reutilização de componentes, a flexibilidade no desenho gráfico e a capacidade de ultrapassar as limitações presentes nas aplicações Web tradicionais.

O IVDP possui um conjunto de aplicações distintas utilizadas na gestão vitícola da RDD, complicando o fluxo de trabalho entre os vários actores envolvidos. Surgiu a necessidade de desenvolver um sistema de informação único com a missão de agregar todas as funcionalidades dessas aplicações. O sistema denomina-se por, Sistema de Informação Vitícola do Douro (SIVD) e é caracterizado por ser um sistema modular assente numa arquitectura baseada em padrões.

A utilização dos Sistema de Informação Geográfica (SIG) na viticultura de precisão possibilita um maior rigor no controlo de qualidade, uma maior disponibilidade de informação sobre as culturas e factores de produção e uma optimização do processo de cadastro e classificação das parcelas com vinha. As parcelas com vinha são porções de terreno, e como tal, possuem natureza geoespacial, tirando partido da informação georreferenciada é possível aplicar conceitos e tecnologias presentes nos SIG, proporcionando novas funcionalidades ao sistema, neste domínio. A aplicação informática permite visualizar a informação georreferenciada da parcela utilizando um serviço de *Web mapping*.

O artigo está organizado em várias secções distintas. Na secção 2 é descrito o trabalho nas áreas relacionadas com o projecto. A secção 3 apresenta o processo inicial de desenho e funcionalidades principais do Módulo de Classificação de Parcelas. A secção 4 ilustra a arquitectura física e lógica do sistema. A secção 5 descreve os principais padrões de desenho utilizados. A secção 6 relata a viabilidade das Rich Internet Applications no desenvolvimento de interfaces gráficas para aplicações Web. A secção 7 apresenta a evolução do MCP para um sistema integrado designado por Sistema de Informação Vitícola do Douro, detalhando o padrão utilizado. A secção 8 descreve o funcionamento do componente responsável pela visualização de informação georreferenciada com recursos a serviços de *Web mapping*. As conclusões e sugestões para trabalho futuro são apresentadas na secção 9.

2. Trabalho relacionado

Os Sistemas de Informação Geográfica (SIG) são utilizados para armazenar, analisar e apresentar informação que descreve as propriedades físicas e lógicas do mundo geográfico [Shekhar 1997]. A dificuldade em delimitar o domínio das aplicações SIG, a variedade do tipo de dados e a optimização da performance leva a que o processo de projecto e implementação deste género de sistemas seja complexo.

A crescente necessidade de utilizar funcionalidades SIG nas aplicações já existentes, levanta a problemática de estender e reutilizar aplicações abertas já existentes em vez de utilizar isoladamente um produto SIG específico [Balaguer 1997]. Dada a complexidade inerente dos SIG e a necessidade de integração nos mais diversos sistemas, os programadores podem beneficiar da utilização de padrões de desenho de software de forma a escrever código limpo,

modular e reutilizável. Os padrões de desenho podem ser definidos como soluções comuns para problemas recorrentes no desenvolvimento de software. Os padrões encapsulam o problema, o seu contexto e propõem uma solução. Um dos objectivos de desenvolver padrões reside na documentação e partilha de soluções, de forma a ajudar a comunidade de programadores no desenvolvimento de software [Câmara 2001].

Pretende-se desenvolver arquitecturas dinâmicas capazes responder aos novos requisitos impostos pela utilização frequente de componentes SIG. Tirar partido das tecnologias mais recentes no desenvolvimento de interfaces gráficas, integrando-as com serviços de *web mapping*. Os SIG tradicionais não possuem um método concreto de processar de uma forma continua, a informação proveniente de múltiplas fontes e por vezes, até de sistemas heterogéneos [Lou et al. 2011]. Com a crescente utilização da WebGIS pelos mais variados ramos da indústria, as interfaces gráficas baseadas no tradicional HyperText Markup Language (HTML), não satisfazem as necessidades dos utilizadores. Além da limitação na interactividade anteriormente referida, o HTML representa apenas a camada de apresentação, sendo necessário outras tecnologias e linguagens de programação para processar a informação, tornando o desenvolvimento lento e complexo [Yang et al. 2010].

As RIA representam uma nova geração no desenvolvimento Web, elevando os níveis de interactividade e optimização na transmissão de informação na rede. Estas tecnologias levantam novos desafios quebrando com os mecanismos tradicionais de pedido-resposta para mecanismos dinâmicos e padrões de desenho assíncronos [Guo et al. 2010]. Aliando as capacidades das RIA ao desenvolvimento WebGIS, possibilita-se o desenvolvimento de mapas mais ricos, interactivos e eficientes.

3. Módulo de Classificação de Parcelas

Após realizado o processo de normalização e modelação do conceito de “Parcela de Vinha da Região Demarcada do Douro” [Coelho et al. 2011], seguiu-se o desenvolvimento da aplicação informática correspondente.

O sistema desenvolvido é constituído por um módulo principal designado por Módulo de Classificação de Parcelas (MCP). Neste módulo são realizadas todas as operações de processamento do enquadramento legal, preenchimento dos boletins de campo e gestão dos direitos de plantação. Na Figura 1 está representado o digrama UML de caso de usos do “Módulo de Classificação de Parcelas”.

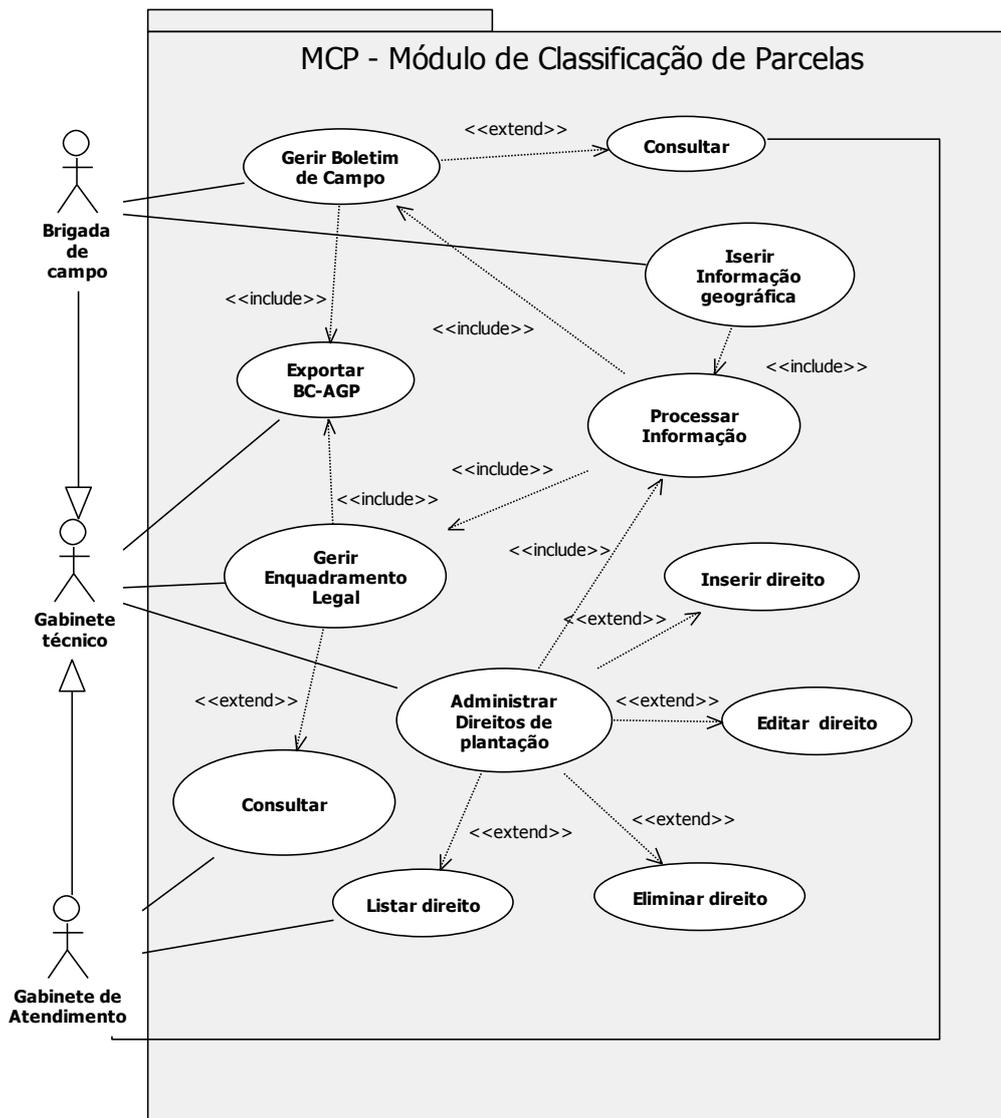


Figura 1 - Diagrama de casos de usos do Módulo de Classificação de Parcelas

Existem três grupos de actores que interagem directamente com a aplicação: o “Gabinete Técnico”, “Brigada de Campo” e o “Gabinete de atendimento”.

A “Brigada de Campo” descola-se ao terreno para efectuar a recolha dos dados. Tem ainda permissão para processar o boletim de campo, previamente preenchidos com informação recolhidos no terreno. Opcionalmente poderá realizar a exportação dos dados (BC-AGP) e consultar os boletins processados.

O “Gabinete de atendimento” representa o grupo de utilizadores que interage directamente com o público. O Gabinete de Atendimento tem permissões para consultar boletins de campo, listar direitos de plantação e consultar enquadramento legal.

O “Gabinete técnico” representa os utilizadores responsáveis pelo processamento do enquadramento legal. O processamento incluiu necessariamente dados com origem geográfica, como por exemplo a área da parcela, o tipo de castas, o tipo de terreno, o declive, a pedregosidade, a localização, a exposição e a aramação (altura e forma do arame que suporta a videira). Este grupo tem privilégios para processar os direitos de plantação, pode efectuar as

operações de inserir, editar, eliminar e listar direitos. Têm ainda permissões para realizar a exportação da informação dos boletins de campo (BC-AGP) para ficheiros estruturados, com vista a serem utilizados por outros sistemas e aplicações.

O MCP é constituído por três casos de uso principais:

- O caso de uso “Gerir Boletim de Campo” representa o processo de inserção de dados dos boletins de campo recolhidos no terreno pelas brigadas de campo.
- O caso de uso “Gerir Enquadramento Legal” representa o processo de atribuição do enquadramento legal de uma parcela, associando à parcela os direitos de plantação que cubram a área da parcela. Este processo resulta na exportação da informação para um ficheiro devidamente formatado.
- O caso de uso “Administrar Direitos de plantação” representa a gestão dos direitos de plantação. Neste caso de uso é possível preencher novo direito, editar informação de um direito existente, eliminar e listar direitos de plantação mediante determinados parâmetros de consulta.

4. Arquitectura do Módulo de Classificação de Parcelas

A arquitectura do sistema segue o paradigma *cliente-servidor* característico das aplicações Web. O servidor é onde está alojada a aplicação e pode também conter o sistema de gestão da base de dados. Caso se justifique, para uma melhor distribuição de carga, podem-se utilizar dois servidores como está ilustrado na Figura 2.

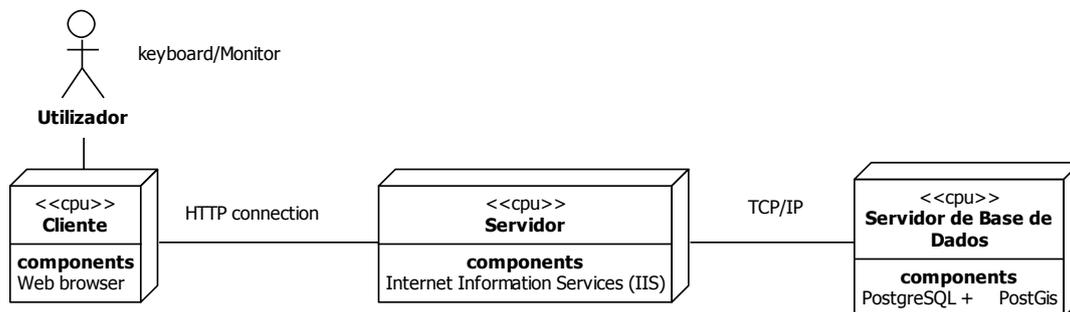


Figura 2- Arquitectura física

A aplicação é constituída pelos três componentes representados na Figura 3. Realizando uma análise *top-down* o primeiro componente corresponde à interface gráfica (GUI), no qual o utilizador interage com o sistema recorrendo a um *Web browser*. O componente intermédio contém todas as regras de negócio e estabelece a comunicação entre as camadas adjacentes de forma transparente para ambas. Toda a lógica de negócio está alojada num servidor onde os acessos são controlados pelo Internet Information Services (IIS). Por fim, o componente que representa o servidor de base de dados. O sistema de gestão de base de dados é o PostgreSQL com extensão espacial PostGIS.

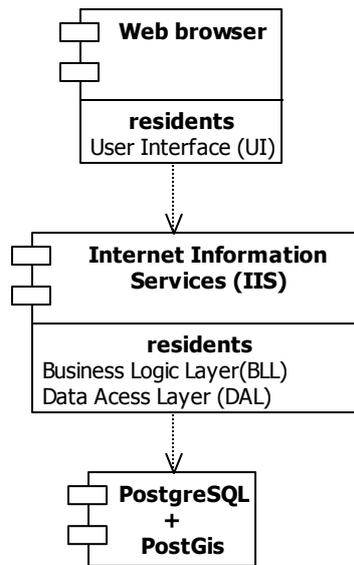


Figura 3 - Diagrama de componentes

Na Figura 4 estão representados diagramas de pacotes lógicos e as relações de composição e dependência entre pacotes. Cada pacote agrupa um conjunto de classes que partilham funcionalidades e responsabilidades comuns. A aplicação é constituída por três camadas: camada de interface, camada de negócio e camada de acesso a dados.

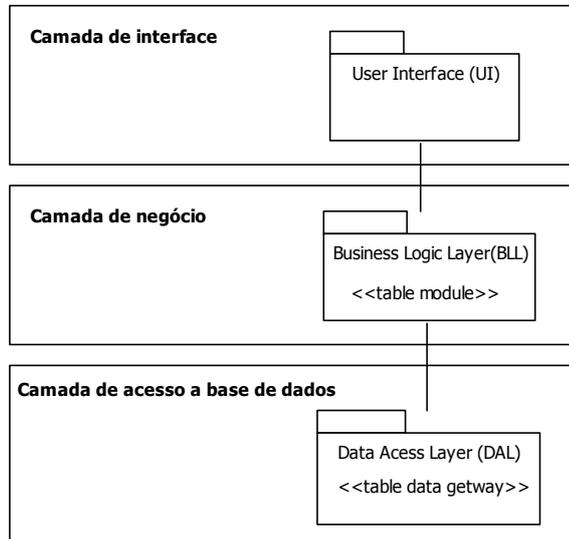


Figura 4 – Arquitectura lógica

Na camada de interface, *User Interface (UI)*, o utilizador interage directamente com a aplicação. Quando um método é invocado, esta camada comunica com camada de negócio, passando apenas os parâmetros que foram recolhidos nos formulários da interface. Após processamento da informação o utilizador visualiza os resultados através da camada de interface. A lógica da camada de interface foi implementada recorrendo a C#, a *Asynchronous JavaScript and XML (AJAX)* e a *Cascading Style Sheets (CSS)*.

A camada de negócio *Business Logic Layer* (BLL) tem como finalidade regularizar todos os processos de negócio. Na camada de negócio reside o núcleo da aplicação em termos de processamento. Recebe os dados da camada de interface realiza todas as validações e cálculos automáticos e enviando posteriormente, a informação para a camada de acesso a dados. A camada de negócio também é responsável por filtrar a informação proveniente da camada de acesso a dados e transferi-la para a camada de interface. A lógica desta camada foi igualmente desenvolvida utilizando C#.

A camada de acesso a dados *Data Access Layer* (DAL) é responsável por gerir a interacção com base de dados e comunicar as respostas à lógica de negócio. O código fonte desta camada foi implementado com o recurso a C#.

5. Padrões de Desenho

Esteve sempre presente, desde a fase embrionária do processo de implementação, o desenvolvimento de código baseado em padrões de desenho, sempre com o objectivo de isolar código fonte comum e criar componentes independentes, que possam ser reutilizados e escalados adicionando novas funcionalidades.

O padrão de desenho utilizado na construção das camadas BLL e DAL designa-se por “*Table Module + Table Data Gateway (DataSets)*”. Esta arquitectura baseia-se na junção de dois padrões, “*Table Module*” e “*Table Data Gateway*” [Fowler 2011]. Na Figura 5 está representado o diagrama de classes com a divisão lógica do código. A figura apresentar o caso particular do enquadramento legal, no entanto, o padrão repete-se para as restantes classes que implementam os outros casos de uso.

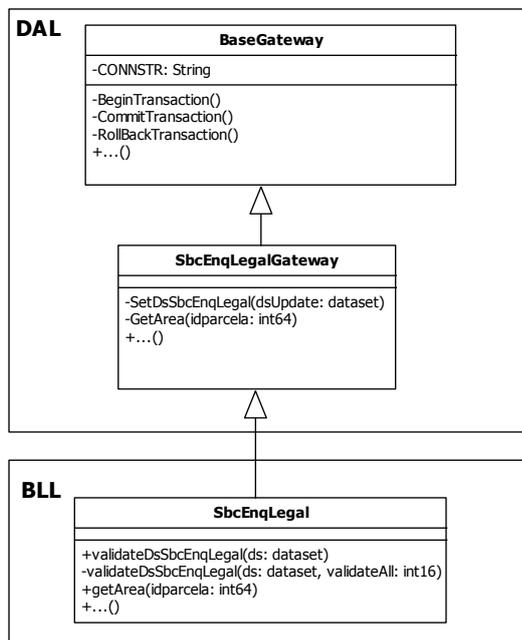


Figura 5 - Table Module + Table Data Gateway

“*Table Data Gateway*” - A classe *BaseGateway* é uma classe genérica que tem como função a gestão dos acessos à base de dados, verificação da ligação, a manutenção de *logs* e controlar as transacções. A classe *SbcEnqLegalGateway* agrupa todas as operações de acesso a base de dados relativas ao enquadramento legal, definindo os parâmetros e as declarações Structured Query Language (SQL). Desta forma, é possível isolar o código do SQL do restante, facilitando a implementação e a manutenção do mesmo. Este padrão repete-se por toda a DAL.

“*Table Module*” - Cada classe da lógica de domínio corresponde a uma tabela. A classe *SbcEnqlegal* representa toda a lógica de domínio relativa ao enquadramento legal. Ou seja, são definidas todas as regras de negócio, como por exemplo: as validações e cálculos automáticos relativos ao enquadramento ao enquadramento legal. Na classe *SbcEnqlegal* o tipo de estruturas é o mesmo da classe *SbcEnqLegalGateway*. Este é o padrão típico da BLL.

6. User Interface do Módulo de Classificação de Parcelas baseada numa Rich Internet Application

Numa fase exploratória do projecto investigou-se a utilização de Rich Internet Applications (RIA) no desenvolvimento da GUI. Com o aumento do grau de exigência imposto pelos utilizadores, as aplicações Web começam a demonstrar as suas limitações. As RIA proporcionam um novo leque de oportunidades no desenvolvimento Web terminado com as limitações apresentadas pelas aplicações Web tradicionais [Rodrigues 2009].

O conceito primário das RIA consiste na reprodução das funcionalidades e comportamento das aplicações *Desktop* na Web. As RIA procuram combinar o melhor de dois mundos, a arquitectura distribuída e os baixos custos de manutenção proporcionada pelas aplicações Web com as capacidades de interactividade e multimédia presentes nas aplicações *Desktop* [Rodrigues 2009].

As RIA seguem o paradigma cliente/servidor, mas contrariamente às aplicações Web tradicionais, as RIA são capazes de transferir parte do processamento da camada de negócio e a gestão da informação para o lado do cliente, com a possibilidade de utilizar a comunicação assíncrona com o servidor [Brambilla 2008]. Ou seja, nas aplicações Web tradicionais todo o processamento é realizado no lado do servidor, o cliente apenas apresenta um ou vários ficheiros HTML com conteúdo essencialmente estático. Este processo sobrecarrega os servidores pois toda a interacção com a aplicação e gestão da informação tem de passar pelo servidor. Segundo o modelo de comunicação das RIA os servidores ficam mais libertos pela diminuição das transacções entre cliente e servidor. Grande parte da informação é transferida para o cliente quando o *browser* carrega o conteúdo da página pela primeira vez. A actualização de conteúdos também difere das aplicações Web tradicionais, apenas actualiza partes da página e não a página por inteiro diminuindo os tempos de resposta na interacção com o utilizador. O motor da RIA encarrega-se por gerir e otimizar a actualização dos conteúdos.

No entanto existe um vazio no que toca a metodologias que permitam a transformação de aplicações Web 1.0 para aplicações Web 2.0. É crítico preservar todo o legado do MCP no que respeita à lógica de negócio, consultas e ligação ao SGBD, pois realiza um conjunto alargado de cálculos e validações de parâmetros já devidamente testados.

Foi proposta a seguinte metodologia baseada em dois conjuntos de operações distintos. Numa primeira fase procede-se a abstracção das camadas de negócio e de dados, separando-as da camada de interface gráfica, utilizando uma camada intermédia de serviços. Numa segunda fase, desenha-se a GUI de acordo com as especificidades da tecnologia envolvida na implementação da RIA [Rodrigues et al. 2009]. A Figura 6, esquematiza os diversos passos a adoptar na utilização da metodologia.

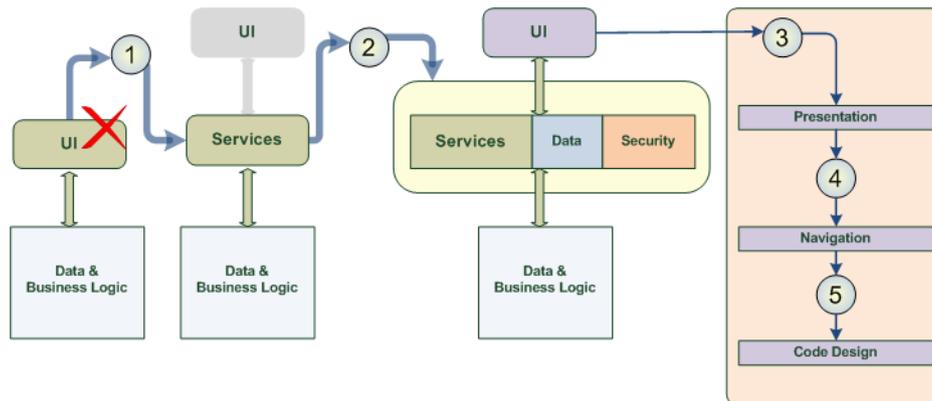


Figura 6 -Metodologia proposta [Rodrigues et al. 2009]

Detalhadamente, o processo de desenvolvimento utilizando a metodologia proposta é descrito nos seguintes tópicos:

- 1) “*Abstracção da User Interface*”: Caso o código não esteja devidamente estruturado, efectuar a separação do código da camada de apresentação das restantes camadas. Nomeadamente, da camada de negócio e de acesso a dados. Sempre que possível recorrer a padrões de desenho de software de forma a otimizar a reestruturação do código.
- 2) “*Camada de Serviços*”: A desacoplação é realizada com a criação de uma camada de serviços. A camada de serviços isola a tecnologia que foi utilizada para implementar as camadas de negócio e de dados. Desta forma, é possível utilizar para a mesma camada de negócio vários tipos de RIA, desde que estas suportem a utilização de *Web Services*. Ainda nesta camada, caso se justifique criar estruturas de dados que auxiliem o tratamento de dados entre a UI e a camada de negócio. Como se trata de uma camada de serviços pode levantar problemas no que diz respeito à integridade da informação e aos processos de autenticação, daí a necessidade de implementar um módulo específico que auxilie este processo.
- 3) “*Apresentação espacial*”: Definir a distribuição e apresentação dos vários elementos pela página. Para tal, utilizar métodos e software que auxiliam a prototipagem de interfaces gráficas.
- 4) “*Sistema de navegação*”: As RIA baseiam-se no paradigma “*single-page application*”. As aplicações Web tradicionais são constituídas por um conjunto de páginas onde é possível navegar entre elas. No caso das RIA o sistema de navegação e interacção é restringido a uma só página. A implementação de mecanismos que possibilitem as funcionalidades anteriormente referidas depende da tecnologia envolvida na construção da RIA.
- 5) “*Desenho do código*”: Realizar uma divisão de código quer física quer lógica de forma a facilitar a manutenção e escalabilidade do código. Podem ser utilizadas diversas técnicas, como por exemplo: a divisão utilizando *namespaces* e a implementação baseada em padrões de desenho.

Como referido anteriormente, a arquitectura do MCP foi projectada a pensar na reutilização e escalonamento da aplicação. As camadas de negócio e de acesso a dados foram implementadas utilizando os padrões “*Table Module*” e “*Table Data Gateway*”. Proporcionando a separação lógica da camada de interface das restantes, reaproveitando toda a camada de negócio e de acesso a dados.

Pondo em prática a metodologia proposta, aproveitou-se o código da BLL e DAL desenvolvido no MCP, implementou-se uma camada intermédia de serviços com o recurso a Windows Communication Foundation (WCF), ver Figura 7.

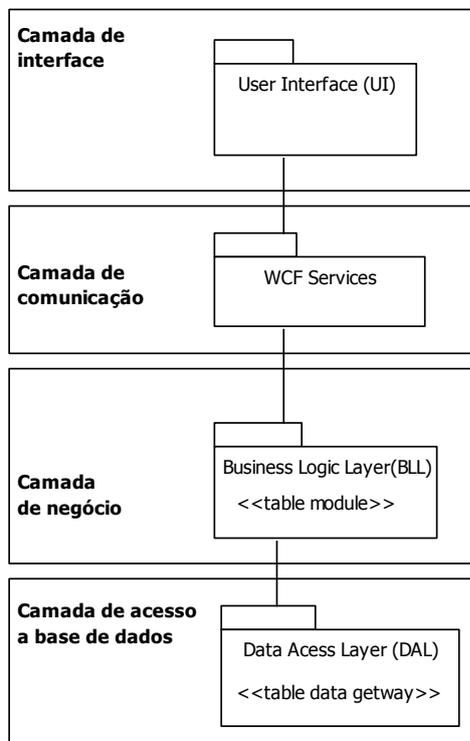


Figura 7 - Camada de serviços

O WCF é uma tecnologia presente na Microsoft .NET Framework desde a versão 3.0, que possibilita a construção de aplicações orientadas a serviços “*Service-Oriented Architecture*” (SOA). Com o objectivo de otimizar a transferência de informação, o WCF disponibiliza as seguintes funcionalidades: comunicação binária otimizada entre aplicações Windows, comunicação síncrona e assíncrona, comunicação com o recurso a transacções, mecanismos de segurança que garantem a integridade da informação e “*Microsoft Message Queuing*” que permite gerir filas de mensagens através da atribuição de prioridades. A interoperabilidade é garantida pela comunicação baseada no protocolo “*Simple Object Access Protocol*” (SOAP). Este foi o protocolo escolhido na implementação dos *Web Services* pois permite a troca de informação, de forma estruturada, baseando-se na associação de XML com uma linguagem descritiva “*Web Services Description Language*”. O SOAP é um dos protocolos *padrão* adoptados na indústria na construção de serviços Web. Desta forma, isola-se completamente a camada de negócio da tecnologia utilizada na implementação da RIA.

As RIA são uma excelente ferramenta que permite melhorar as interfaces gráficas e as arquitecturas SOA são um óptimo *backend* para a camada de negócio, combinadas proporcionam uma solução Web superior [Guo et al. 2010]. A tecnologia RIA escolhida para implementar a GUI foi o *Silverlight*, aposta da Microsoft no desenvolvimento Web, possibilitando a construção de aplicações dinâmicas, interactivas e com suporte multimédia. Na construção da RIA foram utilizadas bibliotecas auxiliares baseadas em padrões de desenho de software, designadas por Prism ou Composite Application Layer (CAL). O Prism é recomendado pela Microsoft *patterns & practices* para o desenvolvimento de aplicações *Silverlight* e WPF [Microsoft 2011].

O protótipo resultante baseia-se em módulos completamente independentes constituídos por componentes perfeitamente delineados. Os módulos são agrupados numa aplicação final interagindo entre si funcionando como um todo. A utilização das bibliotecas auxiliares CAL está intimamente ligada com a utilização de padrões de desenho de software. Consequentemente, demonstrou ser uma boa prática de desenvolvimento, pois possibilita realizar uma gestão e reutilização de código eficiente. Outra das mais-valias na utilização da CAL está na possibilidade de com o mesmo código produzir aplicações para Web (*Silverlight*) e Desktop (*Windows Presentation Foundation*) flexibilizando o desenvolvimento de aplicação, abrindo um novo leque de opções não estando limitado exclusivamente a aplicações Web.

7. Model-View-LogicGateway

O IVDP possui várias aplicações isoladas, com bases de dados próprias, que permitem gerir as demais tarefas relacionadas com a produção do vinho do Porto. Surgiu a necessidade de desenvolver uma aplicação que agregasse todas as aplicações isoladas numa só, com acesso a uma base de dados comum, colmatando com a heterogeneidade de sistemas. Essa aplicação denomina-se Sistema de Informação Vitícola do Porto (SIVD). O SIVD é um sistema modular, dinâmica baseada numa arquitectura padronizada. Mas esta não se limita a agregar as funcionalidades dos vários sistemas, possibilita também tarefas de administração do sistema e geração e exportação de relatórios. A Figura 8, ilustra os módulos principais do SIVD.

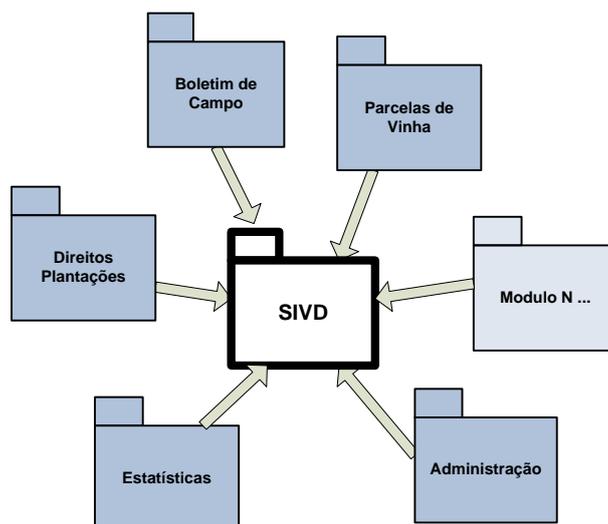


Figura 8 - Modularização SIVD

O SIVD é uma aplicação do tipo *Desktop* desenvolvida em *Windows Presentation Foundation* (WPF), utilizando a linguagem C#. O Sistema de Gestão de Bases de Dados (SGBD) escolhido foi *Microsoft SQL Server 2008*. A arquitectura lógica do SIVD é composta por dois pacotes de classes (*packages*) principais com ligação a uma base de dados. O *package* Middle Data Tier (MD) é responsável pela gestão das regras de negócio, pelas consultas e acessos à base de dados. O *package* User Interface (UI) define todo o *layout* da aplicação responsável pela interacção com o utilizador, ver Figura 9.

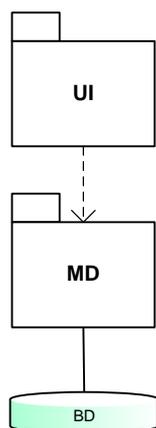


Figura 9 - Arquitectura lógica

Seguindo os mesmos padrões já implementados no MCP (*Table Module + Table Data Gateway*), o código da MD está dividido em duas secções: a *Business Logic Layer* (BLL) e a *Data Access Layer* (DAL), ver Figura 10. Na BLL estão contidas as regras de negócio e na DAL é realizada a gestão dos acessos à Base de Dados. Desta forma foi possível reutilizar a maioria do código desenvolvido no MCP alterando apenas os métodos de ligação à base de dados, adaptando-os para interagirem com uma base de dados do tipo *Microsoft SQL Server 2008*.

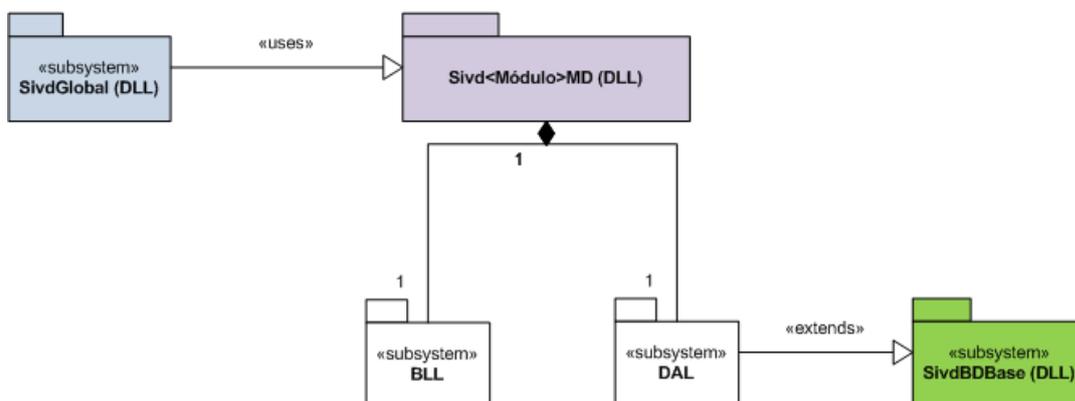


Figura 10 - Padrão interno utilizado na implementação da MD

No projecto exploratório de desenvolvimento de UI baseadas em RIA, foram utilizadas as bibliotecas auxiliares CAL. Esta permite o desenvolvimento em simultâneo de código para *Silverlight* e WPF. Desta forma foi possível reutilizar novamente parte desse código no desenvolvimento da UI no SIVD.

Existe um conjunto de soluções típicas no desenvolvimento de aplicações em WPF, como por exemplo: Model-View-Controller (MVC), e Model-View-Presenter (MVP) e Model-View-ViewModel (MVVM). No entanto, nenhum dos padrões possibilita uma reutilização considerável do código já desenvolvido de forma a manter o legado das aplicações anteriores. Sendo incompatíveis com os padrões já utilizados no desenvolvimento da MD. Para resolver esta limitação, estabeleceu-se um padrão denominado Model – View – LogicGateway (MVLG), com o objectivo de tornar o desenvolvimento da UI da aplicação SIVD mais robusto, modular, facilitando a reutilização de código e a realização de testes. O MVLG é responsável pela arquitectura de alto-nível da UI, agrupando código com funcionalidades comuns em pequenos módulos, criando elos de ligação entre eles. A Figura 11 representa a interacção entre a MD e UI utilizando o padrão MVLG.

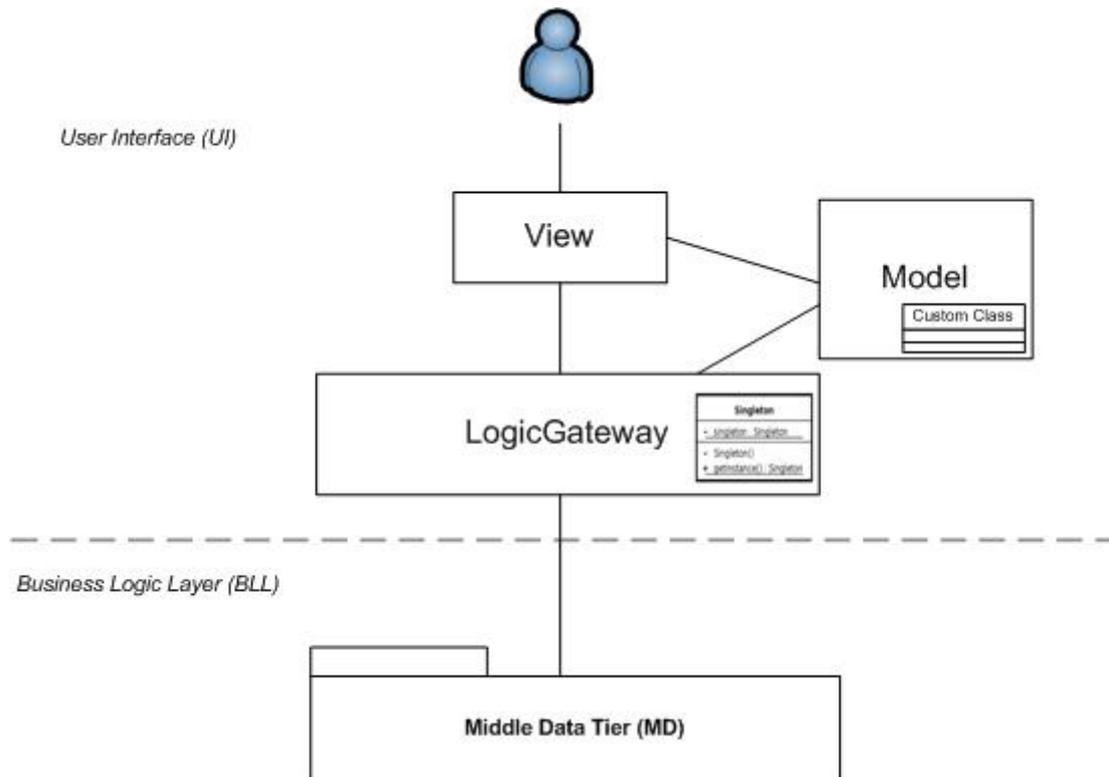


Figura 11 - Model-View-LogicGateway

A View (V) especifica todo o desenho da UI, essencialmente em código de apresentação gráfica), não são realizadas quaisquer validações ou cálculos. No protótipo desenvolvido a UI foi implementada em Extensible Application Markup Language (XAML). O Model (M) facilita os *castings* necessários na transferência de dados entre a View e Logic Gateway, definindo *custom classes* ou estruturas de dados responsáveis pela replicação lógica do modelo de dados. Por sua vez, a Logic Gateway (LG) é responsável por fazer a ponte entre os *packages* MD e a UI utilizando as *custom classes* definidas no Model. A Logic Gateway realiza algumas validações, como por exemplo a validações do tipo e formato dos dados inseridos da interacção na View com o utilizador. Caso haja alguma incoerência notifica a View.

O MVLG é constituído por padrões que auxiliam a composição e criação de objectos. Como por exemplo, o padrão de criação “Gang-of-Four” (Gof) Singleton, que é responsável por gerir as instâncias das classes da LogicGateway que comunicam com a MD.

8. Visualização da informação georreferenciada

Dada a natureza geoespacial da parcela com vinha, foi desenvolvido um módulo específico que possibilita a localização e a visualização da informação georreferenciada. Esta funcionalidade do sistema permite aos vicultores, ao gabinete de atendimento e ao gabinete técnico visualizar o posicionamento e a forma da área da parcela com vinha, facilitando a detecção de eventuais irregularidades.

Adicionando a extensão espacial PostGIS ao sistema de gestão de base de dados PostgreSQL, é possível exportar para o formato Keyhole Markup Language (KML) o resultado das consultas realizadas às parcelas com vinha. Estes resultados consistem num conjunto de coordenadas que representam um polígono equivalente á área da parcela com vinha.

O *Bing Maps* disponibiliza serviços grátis que podem ser utilizados na interacção com mapas, No entanto, uma das limitações da versão actual é a impossibilidade de desenhar directamente

multi-polígonos de acordo com os resultados provenientes da base de dados, originando por exemplo, figuras geométricas incorrectas. A Figura 12 apresenta um exemplo incorrecto, no lado esquerdo, da forma incorrecta de representação do polígono. Quando o pretendido é o desenho de figuras geométricas sem sobreposições, semelhantes ao lado direito da Figura 12.



Figura 12 - Multi-polígonos

Para resolver este problema realiza-se uma filtragem dos pontos provenientes do KML. Seguidamente, aproveitando a capacidade do serviço disponibilizado pelo *Bing Maps* possibilitar a representação de figuras geométricas em camadas distintas, inicia-se o processo de desenho da parcela com vinha. Inicialmente desenharam-se os pontos previamente filtrados e unem-se utilizando linhas. De seguida, desenha-se o polígono sem representar as bordas. Ao sobrepor as camadas obtêm-se o efeito desejado. Isto é visível na Figura 13.

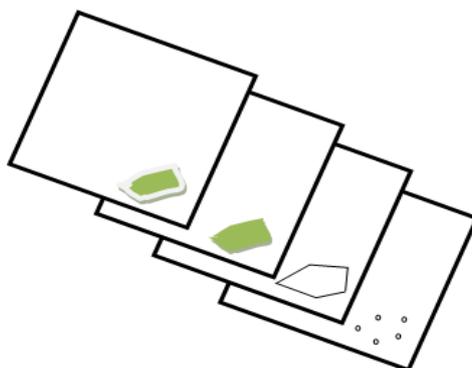


Figura 13 - Mapa multi-camadas

A Figura 14 ilustra o resultado da filtragem e desenho da parcela, que é apresentado ao utilizador na UI do MCP. Está representada uma parcela com vinha cujo formato se baseia num multi-polígono. Esta imagem prova que a filtragem de pontos e a utilização do sistema de camadas utilizado produzem os resultados pretendidos. O controlo utilizado tem as seguintes funcionalidades: diferentes níveis de *zoom*, controlos de navegação, visualização interactiva de mapas e de imagens de satélites. Isto é visível na Figura 14.



Figura 14 - Visualização de parcelas com vinha

Além da visualização gráfica da parcela é possível visualizar a informação de forma descritiva, como indica a Figura 15.

Parcela Processada	
Número Processo - Número Boletim:	[Redacted]
Geocódigo:	[Redacted]
Distrito:	04-Bragança
Concelho:	09-Torre de Moncorvo
Freguesia:	16-Torre de Moncorvo
Local:	Torre de Moncorvo
Nome Explorador:	[Redacted]
Nome Proprietário:	[Redacted]
Área Potencial (ha):	1,0132

Figura15 – Informação descritiva da parcela com vinha

No SIVD o *plugin* do *Bing Maps* disponibiliza funcionalidades adicionais, como a possibilidade de visualizar o mapa em 3D podendo escolher vários níveis de inclinação. Isto é visível na Figura 17.

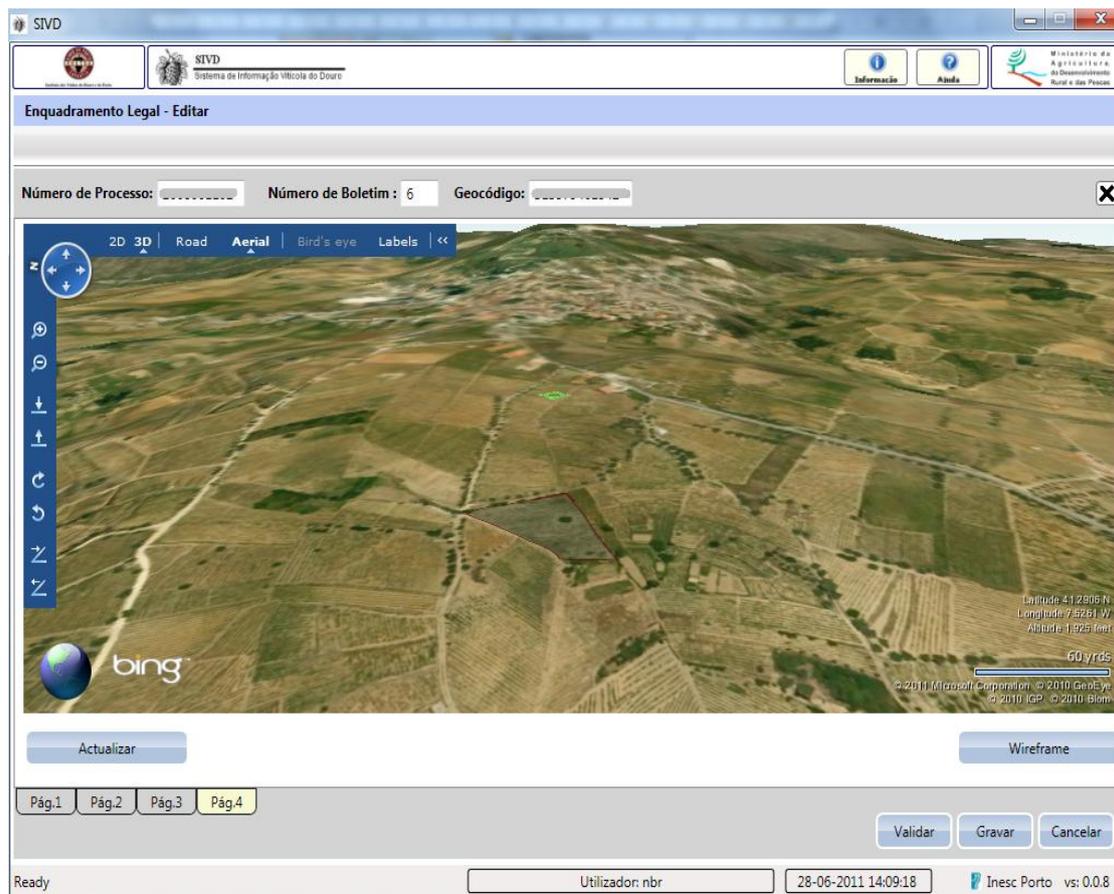


Figura 17- Visualização da parcela no SIVD

9. Conclusão

Com este projecto desenvolveu-se um SIG dinâmico, interativo e escalável adaptado às necessidades da Região Demarcada do Douro. Foram ainda exploradas as capacidades das tecnologias Web, nomeadamente as RIA, no desenvolvimento de UI tendo em conta o desempenho e a interactividade.

Ao longo de todo o processo de implementação, não foi esquecida a utilização de padrões de desenho de forma a manter o código reutilizável e escalável. Este facto demonstrou ser uma mais-valia, pois sempre que se optou por mudar de tecnologia e de plataforma no desenvolvimento da UI, grande parte do código foi mantido evitando esforço adicional da reescrita do mesmo.

Foi implementado uma aplicação que permite ao IVDP reunir as funcionalidades presentes em várias aplicações dispersas, diminuindo a entropia entre os vários sistemas aumentando o grau de transparência na troca de informação entre as várias entidades do IVDP.

Foi ainda possível desenvolver mecanismos e componentes que possibilitam a visualização de informação georreferenciada relativa às parcelas agrícolas com vinha, recorrendo serviços de *Web mapping* sem custos de utilização, demonstrando ser uma alternativa às aplicações comerciais.

Como trabalho futuro, pretende-se aplicar os padrões de desenho e a metodologia proposta, a RIA baseadas em motores distintos do Silverlight, como por exemplo, Flex e Javacript. Melhorar o componente responsável pela gestão da informação georreferenciada, dotando-o de

capacidade de edição directa no mapa por parte do utilizador. Por fim, focar em padrões de desenho direccionados para questões como usabilidade e interacção pessoa-computador.

10. Referências

- Balaguer, F., Gordillo, S., Das Neves, F., "Patterns for GIS Applications Design", Proceedings of PLoP'97 - Pattern Languages of Programming, La Plata, Buenos Aires, Argentina, 1997
- Câmara, G., Vinhas, L., Souza, R., Paiva, J., Monteito, A., Carvalho, M. e Raoult, B., "Design Patterns in GIS Development: The Terralib Experience", In III Workshop Brasileiro de Geoinformatica, GeoInfo, Image Processing Division, National Institute for Space Research (INPE), São José dos Campos, Brazil, 2001
- Coelho, A., Silva, L., Oliveira, L., *OIV2011: The concept of Vineyard Parcel for the Douro Delimited Region*, The Construction of Wine – A Conspiracy of Knowledge and Art, Porto, Portugal, 2011.
- Brambilla, M., Preciado, J. C., Linaje, M. & Sanchez-Figueroa, F. (2008) Business Process-Based Conceptual Design of Rich Internet Applications. *Web Engineering, 2008. ICWE '08. Eighth International Conference on*.
- Fowler M., *Catalog of Patterns of Enterprise Application Architecture*, <http://martinfowler.com/eaaCatalog/> (27 de Junho de 2011)
- Luo, J., Xiao, G., "Expression of GIS maps based on Flex," Geoinformatics, 2011 19th International Conference on , vol., no., pp.1-4, 24-26, Junho 2011
- Microsoft, *Developer's Guide to Microsoft Prism*, <http://msdn.microsoft.com/en-us/library/gg406140.aspx> (27 de Junho de 2011)
- Guo, L., Gong, J., Sun, J., Wei, X., "Study on GIS architecture based on SOA and RIA," Information Sciences and Interaction Sciences (ICIS), 2010 3rd International Conference on , vol., no., pp.620-625, 23-25 June 2010
- Rodrigues, N., *Aplicação Web para Classificação de Parcelas Agrícolas*, Faculdade de Engenharia da Universidade do Porto, 2009.
- Rodrigues, N., Oliveira, L. e Coelho, A., "Adapting traditional web applications to new user interfaces provided by rich internet applications", IADIS 2009 - IADIS International Conference www/Internet 2009, vol.2, pp.395-398, Roma, Itália, 2009.
- Shekhar, S., Coyle, M., Goyal, B., Liu, D. e Sarkar, S., "Data Models in Geographic Information Systems" Comm. of the ACM, April 1997, pp 103-111
- Yang, L., Xue-Feng, L., Jian-Hua, M., "Research on the Integration of Silverlight and WebGIS Based on REST," Multimedia Technology (ICMT), 2010 International Conference on , vol., no., pp.1-4, 29-31 Oct. 2010