

Instituto Politécnico de Beja
Escola Superior de Tecnologia e Gestão
Curso de Engenharia Informática



1º Relatório Intercalar de Projecto

“Sistema de Construção Visual de Interfaces”

Cláudia Isabel Hermozilha Oliveira, N.º 3459

Cláudio Samuel Monteiro Pedro, N.º 3805

Nuno Filipe Alves Coelho, N.º 3938

2007

Instituto Politécnico de Beja
Escola Superior de Tecnologia e Gestão
Curso de Engenharia Informática

1º Relatório Intercalar de Projecto
Sistema de Construção Visual de Interfaces

Primeiro relatório intercalar de projecto de fim de curso, apresentado na Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Beja, como parte dos requisitos necessários à obtenção do grau de licenciado no curso de Engenharia Informática.

Alunos

Cláudia Isabel Hermozilha Oliveira, N.º 3459

Cláudio Samuel Monteiro Pedro, N.º 3805

Nuno Filipe Alves Coelho, N.º 3938

Orientador

Eng.º José Jasnau Caeiro

Beja, Janeiro de 2007

Índice

1	INTRODUÇÃO.....	3
1.1	DESCRIÇÃO DO PROJECTO	3
1.2	O QUE FOI FEITO NESTE RELATÓRIO INTERCALAR	4
1.3	ESTRUTURA DESTE DOCUMENTO.....	4
2	FERRAMENTAS ANÁLOGAS.....	5
2.1	ESTUDO DAS FERRAMENTAS	5
2.2	CARACTERÍSTICAS A IMPLEMENTAR	7
3	METODOLOGIA.....	9
3.1	DESCRIÇÃO DAS FASES DE DESENVOLVIMENTO	9
4	ESPECIFICAÇÃO	11
4.1	ANÁLISE DE REQUISITOS	11
5	CONCLUSÃO.....	14
6	BIBLIOGRAFIA	15
 APÊNDICES		
APÊNDICE 1 – GRÁFICO DE GANTT DO PROJECTO		1
APÊNDICE 2 – TEMPLATES DOS CASOS DE USO.....		2

1 Introdução

1.1 Descrição do projecto

A construção de aplicações *Web* mais interactivas é possível através das tecnologias de desenvolvimento *Web*, AJAX¹ [1]. Um dos factores fundamentais para o aumento de interactividade é a troca constante de pequenas quantidades de dados com o servidor, em *background*. Desta forma, as páginas *Web* não precisam de ser actualizadas na sua totalidade, cada vez que é feita uma alteração.

Qooxdoo [2] é uma das mais poderosas *frameworks* *AJAX open-source* porque disponibiliza uma vasta quantidade de classes e um conjunto de *widgets* que dificilmente se distinguem dos elementos gráficos que estamos habituados a ver em aplicações nativas de sistemas operativos. É uma *framework* que permite desenvolvimento através da linguagem JavaScript, suporta comunicação cliente-servidor de alto nível e inclui um *toolkit* gráfico que apresenta características semelhantes às de outras bibliotecas gráficas existentes, como a Qt (Dalheimer, 2002) [3].

As interfaces qooxdoo são desenvolvidas através de *scripts* escritos na linguagem JavaScript, que permitem maximizar o desempenho de execução das mesmas. Integradas num ambiente *Web* do tipo cliente-servidor, estas interfaces são processadas no lado do cliente através de uma *framework* JavaScript instalada no *browser*, que implementa as propriedades gráficas da biblioteca qooxdoo. Ou seja, qualquer *browser* que suporte *JavaScript* como o Internet Explorer, Mozilla Firefox ou Opera, pode processar estas interfaces.

O desenvolvimento de interfaces gráficas em modo de texto é moroso e de difícil acerto dadas as claras dificuldades com que um programador se depara, como por exemplo, o caso da frequente compilação ou execução da codificação para visualização da interface criada. Com o objectivo de ajudar no desenvolvimento de interfaces gráficas, existem hoje em dia inúmeras ferramentas do tipo IDE², que permitem de forma interactiva e rápida, construir interfaces bem estruturadas e apelativas. Ao contrário do que acontece com outros *toolkits* gráficos, para o qooxdoo não existe actualmente nenhum ambiente do género, desenvolvido.

Este projecto tem como objectivo o desenvolvimento de uma ferramenta gráfica que sirva de suporte à construção de interfaces constituídas por elementos do *toolkit* gráfico qooxdoo. Pretende-se com esta ferramenta diminuir o tempo gasto pelos programadores na construção das interfaces, facilitando-lhes ainda a construção das mesmas. No desenvolvimento desta ferramenta ir-se-á recorrer ao *toolkit* gráfico Qt que apresenta duas vantagens de extrema relevância. Permite efectuar *drag & drop* e os seus controlos gráficos possuem *layouts* semelhantes aos de controlos com a mesma função, existentes no *toolkit* qooxdoo.

¹ *Asynchronous JavaScript and XML*

² *Integrated Development Environment*

A ferramenta a desenvolver destina-se à *framework* AJAX qooxdoo, mas também poderia ser desenvolvida para outras *frameworks*, bibliotecas ou *toolkits* AJAX. De seguida são apresentados aspectos a ter em conta sobre alguns dos maiores concorrentes da qooxdoo. A *framework* Bindows [4], por exemplo, oferece um excelente GUI³ e não requer conhecimentos de HTML⁴ nem de CSS⁵, mas tem a grande desvantagem de não ser *open-source*. Já o *toolkit* Dojo [5], é *open-source*, ajuda a otimizar o código JavaScript e permite a criação de *widgets*, no entanto, problemas com o DHTML⁶, têm-no impedido de ser adoptado em massa pelos programadores. Pode-se ainda referir a biblioteca *open-source* Prototype [6], que possui um poderoso sistema de classes e é orientada a objectos, mas não apresenta *widgets*.

1.2 O que foi feito neste relatório intercalar

Neste relatório intercalar, começou-se por desenhar o gráfico de Gantt relativo ao projecto. Depois, realizou-se um estudo sobre ferramentas análogas que permitiu efectuar um levantamento de características que deverão ser implementadas na ferramenta a desenvolver. Descreveram-se as fases de desenvolvimento do projecto, elaborou-se o diagrama de casos de uso, preencheram-se as templates dos mesmos, e por fim, descreveram-se três possíveis cenários de utilização para a aplicação.

1.3 Estrutura deste documento

Este documento divide-se por seis secções. Na secção 1 denominada de *Introdução*, encontra-se uma descrição do projecto, informação sobre o que foi feito no mesmo e a estrutura deste documento. Na secção 2 denominada de *Ferramentas análogas*, pode-se ler o estudo realizado sobre ferramentas semelhantes e ainda, as características a implementar na ferramenta a desenvolver. Na secção 3 denominada de *Metodologia*, descrevem-se as fases de desenvolvimento. Na secção 4 denominada de *Especificação*, encontra-se o diagrama de casos de uso da ferramenta, bem como, três possíveis cenários de utilização para a mesma. Na secção 5 denominada de *Conclusão*, descrevem-se as conclusões retiradas durante a realização deste relatório intercalar, e na secção 6 denominada de *Bibliografia*, estão enunciadas as referências bibliográficas adoptadas para a realização deste documento. Na secção *Apêndices*, encontram-se o *Apêndice 1* que corresponde ao gráfico de Gantt do projecto, e o *Apêndice 2*, que contém as templates dos casos de uso.

³ Graphical User Interface

⁴ HyperText Markup Language

⁵ Cascading Style Sheets

⁶ Dynamic HyperText Markup Language

2 Ferramentas análogas

2.1 Estudo das ferramentas

Com o objectivo de salientar as características mais importantes numa ferramenta deste género, foi realizado um estudo que incidu sobre várias ferramentas de desenvolvimento de interfaces gráficas. No âmbito deste estudo foram tomadas em conta as seguintes ferramentas:

Aptana [7]

Características:

- Assistência na construção do código JavaScript, HTML e CSS;
- Comunicações através dos protocolos FTP⁷ ou SFTP⁸;
- *Debugger* JavaScript;
- Avisos e notificações de erros;
- Permite personalizar a própria interface;
- Suporte para cruzar plataformas.

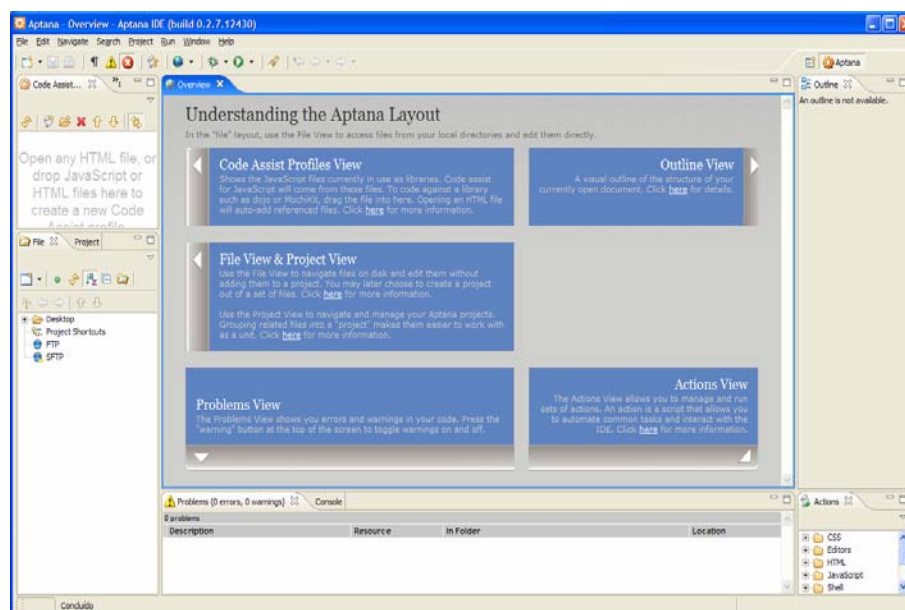


Figura 2.1 – Interface do Aptana

Macromedia Dreamweaver [8]

Características:

- Construção de sítios Web estáticos, constituídos por páginas Web HTML;

⁷ File Transfer Protocol

⁸ Secure File Transfer Protocol

- Criação de ficheiros de definições de apresentação para páginas HTML, através da linguagem de estilos CSS;
- Criação de páginas HTML a partir de modelos existentes;
- Armazenamento remoto de ficheiros;
- Possibilidade de adicionar à área de construção das páginas grelhas e régua, de forma a facilitar e otimizar a construção de interfaces visuais;
- Funcionalidade *drag & drop* na colocação de controlos visuais, na área de construção das interfaces;
- Visualização estruturada de todos os ficheiros que representam o projecto em que o utilizador está a trabalhar;
- Para cada controlo visual apresentado na área de construção de interfaces, são indicadas as suas dimensões relativamente ao ecrã, o espaço ocupado em percentagem e o tamanho em *pixels*;
- Apresentação simultânea da área de desenho e do código da página *Web*;
- Criação de *records* sobre as acções do utilizador, gravando acções do mesmo, que mais tarde poderão ser repetidas sobre a área de desenho, bastando pressionar sobre a opção de repetir acção.

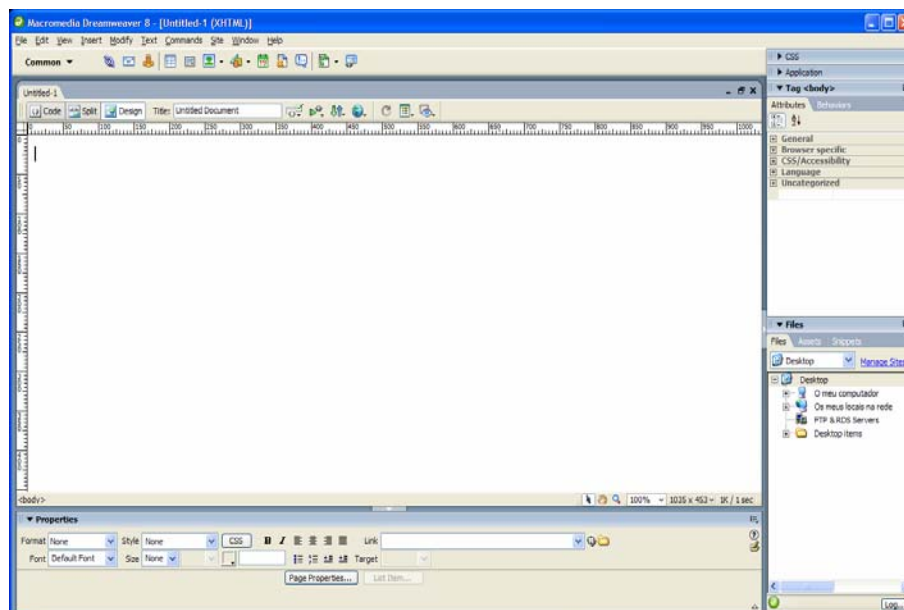


Figura 2.2 – Interface do Macromedia Dreamweaver

Qt Designer [9]

Características:

- *Drag & drop* na construção de interfaces gráficas;
- Modo de pré-visualização;
- Divisão dos controlos por tipo: *Input Widgets, Buttons, Spacers, Containers*, etc.;
- Criação de super-controlos⁹;
- Criação de templates;

⁹ Controlo visual composto por um ou mais controlos simples.

- Criação de janelas a partir de opções predefinidas:
 - *Dialog*;
 - *Widget*;
 - *Main Window*;

Na opção por esta última, é disponibilizado um conjunto de opções que permitem personalizar a janela a ser criada. É possível escolher, por exemplo, um *File Menu*, um *Toolbar Menu*, um *Help*, etc.

- Listagem das propriedades de um controlo;
- Listagem dos eventos associados a um controlo;
- Funcionalidades de anular, refazer, copiar, cortar e colar para executar sobre os controlos;
- Possibilidade de formatar o posicionamento dinâmico ou estático dos controlos, através do conceito de ancoragem.

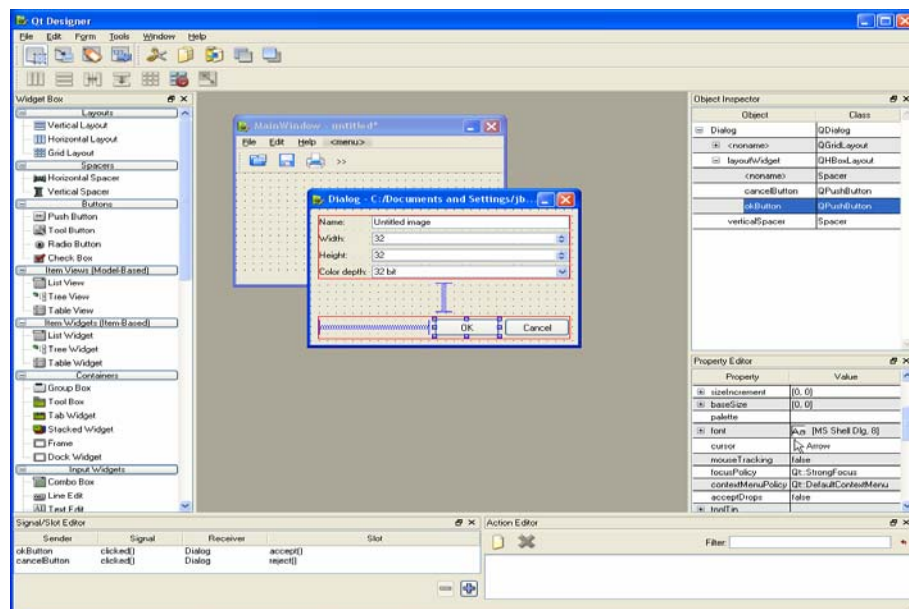


Figura 2.3 – Interface do Qt Designer

2.2 Características a implementar

Do estudo realizado, resultou um conjunto de características que se pretendem ver implementadas na ferramenta a desenvolver:

- Funcionalidade de *drag & drop* para a manipulação dos controlos visuais na construção das interfaces;
- Pré-visualização da interface criada, num *browser*;
- Pré-visualização da interface criada, na aplicação;
- Editor de propriedades que permitirá formatar os controlos visuais implementados na área de desenho;
- Armazenamento de templates com as características visuais de controlos formatados;

- Aplicação de templates de controlos visuais sobre controlos do mesmo tipo;
- Funcionalidades de anular, refazer, copiar, cortar, colar e apagar para executar sobre os controlos visuais existentes na área de desenho.

3 Metodologia

3.1 Descrição das fases de desenvolvimento

Para desenvolver software de qualidade é necessário ter em consideração um conjunto de factores como:

- O que fazer antes de começar a desenvolver?
- Quais as estruturas necessárias para um desenvolvimento adequado?
- O que deve ser feito para o software responder às necessidades estabelecidas?
- Que técnicas se devem utilizar para controlar a complexidade, prever as falhas, documentar o software e controlar as versões?

No desenvolvimento desta ferramenta ir-se-á adoptar o modelo de desenvolvimento conhecido por modelo cascata [10], por este ser recomendado para o desenvolvimento de software onde existe um conjunto de requisitos estáveis e de elevada qualidade, e em que a duração prevista para a elaboração do desenho do mesmo não é extensa.

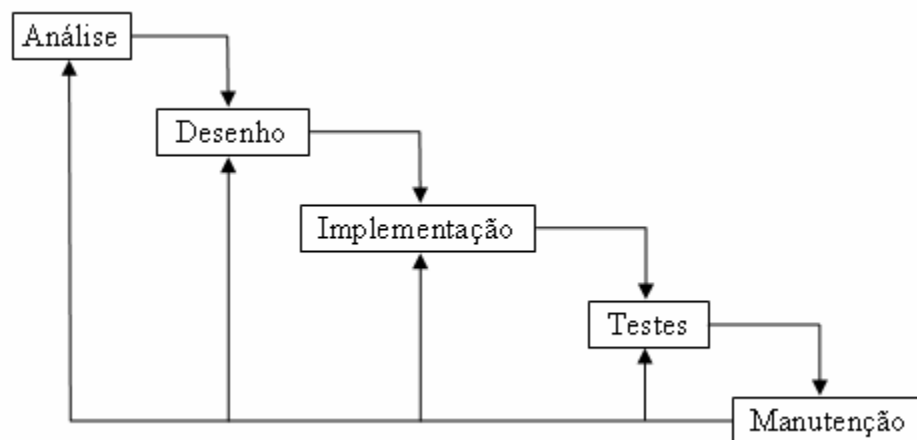


Figura 3.1 – Modelo Cascata

O desenvolvimento deste projecto será dividido em várias fases, de acordo com o modelo apresentado e por forma de endereçar cada um dos factores anteriormente listados, no devido momento.

Numa fase inicial, que corresponde à fase da análise no modelo acima apresentado, será estudado o problema de forma a identificar as características que a ferramenta a desenvolver deverá ter. Serão estudadas ferramentas análogas de forma a reter as principais características que oferecem, assim como para perceber as dificuldades que poderão surgir no desenvolvimento de um projecto deste tipo.

De seguida, numa fase que corresponderá à fase de desenho do modelo cascata, será elaborado o desenho do sistema onde serão identificados os módulos da implementação do projecto, as suas características e os tempos de desenvolvimento previstos. Serão desenhados alguns modelos visuais onde serão projectadas interfaces gráficas da aplicação. Estas vão ser objecto de constantes refinações até adquirirem um aspecto óptimo na óptica do utilizador final. A partir das interfaces criadas será possível reproduzir *storyboards*¹⁰ [11].

Durante a fase de implementação existente no modelo cascata serão implementados os módulos identificados na fase de desenho. A implementação do sistema irá ser efectuada pelos vários elementos do grupo de uma forma partilhada, e controlada por ferramentas de controlo de versões.

Passar-se-á então à fase de testes, durante a qual se procurará verificar que os requisitos detectados durante a fase de análise foram complementados.

Relativamente à última fase do modelo cascata, a da manutenção, há que referir que esta não será tomada em conta durante a realização deste projecto. No entanto, é uma fase que se estende a partir do momento em que o software é disponibilizado, e que consiste na correcção de erros previamente não detectados e no melhoramento da funcionalidade do sistema, sempre que possível.

Resta afirmar que a documentação do código será realizada através de uma ferramenta de geração automática de código e posteriormente publicada no sítio *Web* do projecto. Também serão criados manuais de instalação e de utilização, com o intuito de fornecer algum apoio aos utilizadores.

¹⁰ Cenários de utilização do sistema.

4 Especificação

4.1 Análise de requisitos

Casos de Uso



Figura 4.1 – Diagrama de Casos de Uso da aplicação a desenvolver

Cenários de utilização

Cenário n.º 1 – Utilização de templates na formatação de controlos visuais

A Eng. Sofia, como programadora *Web* tem de projectar um sítio *Web* para um projecto sobre a marcação de férias de funcionários de uma empresa. Este sítio *Web* será constituído por várias páginas *Web*, sendo que em cada uma delas existirão vários tipos de controlos visuais.

De forma a normalizar a estética do sítio *Web*, todos os controlos visuais terão de ter uma formatação semelhante. A Eng. Sofia antes de construir a sua interface irá proceder à construção de templates de formatação para cada um dos controlos visuais que as páginas *Web* irão utilizar. Abriu a aplicação e para a área de desenho arrastou um botão e uma caixa de texto. O botão terá a função de aceitar alterações e a caixa de texto servirá para a introdução de dados. Relativamente ao botão, foi alterado o seu tamanho em largura para 60 *pixels* e em altura para 30 *pixels*. A cor de fundo escolhida para o botão foi a cinzenta e foi-lhe ainda aplicada uma imagem de fundo que reflecte metaforicamente aceitação. Relativamente à caixa de texto, foi alterado o seu tamanho em largura para 150 *pixels* e em altura para 30 *pixels*. A cor de fundo escolhida para a caixa de texto foi a amarela clara. Em ambos os controlos, o tipo de letra escolhido foi “Arial” com cor azul e tamanho 12. Para uma posterior aplicação destas formatações sobre controlos do mesmo tipo em outras interfaces, a Eng. Sofia armazenou em disco uma template para cada um destes tipos de controlos.

No dia seguinte decide então começar a construção das interfaces gráficas das páginas *Web* pela colocação dos controlos visuais necessários. A criação prévia de templates irá facilitar a formatação dos botões e das caixas de texto. A Eng. Sofia foi seleccionando um a um os botões existentes na interface e cada vez que seleccionava um, com o botão direito do rato escolhia a opção “Apply template...”. Surgia-lhe então uma janela na qual seleccionava a template que queria aplicar sobre o botão, sendo então aplicada a formatação que a template armazenava. Relativamente às caixas de texto, ela seleccionou todas de uma só vez e com o botão direito do rato sobre uma delas, seleccionou a opção “Apply template...”. Neste caso, escolheu a template que armazenava a formatação que a Eng. Sofia havia criado para as caixas de texto. Por fim, guardou as alterações efectuadas sobre o projecto.

Cenário n.º 2 – Pré-visualização da interface gráfica criada na aplicação e num *browser*

O Cristiano, um aluno do curso de engenharia informática da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Beja, estava encarregue do desenvolvimento de uma página *Web* pessoal para a professora da disciplina de Matemática Discreta.

Para a construção da interface da página *Web*, o Cristiano optou pela ferramenta de desenvolvimento de interfaces gráficas *Web*, Qooxdoo GUI Builder. Iniciou a aplicação, dispôs os controlos visuais que achava necessários sobre a área de desenho e formatou os mesmos a seu gosto. Recorreu à opção “Preview in application” existente no menu “Preview” da aplicação e após esta pré-visualização efectuou o derradeiro teste que passava pela visualização da interface criada, no *browser* por defeito existente no sistema. Para conseguir visualizar a interface num *browser*, recorreu à opção “Preview in browser” existente no menu “Preview” da aplicação. Satisfeito com a aparência conseguida, guardou a interface gráfica e enviou a mesma por correio electrónico para a professora.

Cenário n.º 3 – Modificação de interfaces gráficas previamente criadas

O Danilo, um rapaz empregado numa loja de informática do comércio local, decidiu nos seus tempos livres construir uma página *Web* para a loja.

Idealizou a interface gráfica que pretendia construir e iniciou a sua construção recorrendo ao menu da aplicação, onde dentro do menu “File” escolheu a opção “New interface”. Entretanto entraram clientes na loja e ele viu-se obrigado a interromper a interface na qual estava a trabalhar. Decidiu então guardá-la recorrendo à opção “Save interface...” existente no menu “File” da aplicação, para retomar mais tarde quando tivesse algum tempo livre.

Mais tarde já em casa, sem a interferência dos clientes a entrar e sair, ele decidiu retomar o seu trabalho. No menu “File” da aplicação escolheu a opção “Open interface...”. Depois seleccionou a interface que queria modificar e deu continuidade à sua construção acrescentando controlos visuais e realizando formatações sobre os mesmos.

Por fim, deu por terminada a interface em que esteve trabalhando e guardou-a recorrendo à opção “Save interface...” existente no menu “File” da aplicação. No entanto, o Danilo sabe que poderá voltar a modificá-la em qualquer momento.

5 Conclusão

Durante a realização deste relatório intercalar foi-se chegando a várias conclusões. A primeira conclusão a que se chegou foi que a *framework* qooxdoo merece de facto o desenvolvimento de um IDE a ela destinado, sobretudo devido aos excelentes componentes gráficos que oferece para a construção de interfaces.

Através do estudo realizado sobre ferramentas análogas, tiraram-se conclusões quanto às características cuja implementação na ferramenta a desenvolver, é fundamental. De entre as várias características destacam-se o *drag & drop* na manipulação de controlos visuais, a criação e aplicação de templates e os modos de pré-visualização que permitem pré-visualizar a interface existente na área de desenho.

6 Bibliografia

- [1] *Ajax (programação)*. (2007, Janeiro 14). [Online]. Disponível em http://pt.wikipedia.org/wiki/AJAX_%28programa%C3%A7%C3%A3o%29
- [2] *qooxdoo*. (2006, Dezembro 22). [Online]. Disponível em: <http://qooxdoo.org>
- [3] Matthias Kalle Dalheimer, *Programming with Qt, 2nd Edition*, O'Reilly, Janeiro 2002, capítulo 1.
- [4] *Bindows*. (2007, Janeiro 29). [Online]. Disponível em <http://www.bindows.net>
- [5] *Dojo*. (2006). [Online]. Disponível em <http://dojotoolkit.org>
- [6] *Prototype*. (2007). [Online]. Disponível em <http://www.prototypejs.org>
- [7] *Aptana*. (2006, Dezembro 20). [Online]. Disponível em <http://www.aptana.com>
- [8] *Macromedia Dreamweaver*. (2006, Junho 21). [Online]. Disponível em <http://www.adobe.com/products/dreamweaver>
- [9] *Qt Designer*. (2007). [Online]. Disponível em <http://www.trolltech.com/products/qt/features/designer>
- [10] *Waterfall model*. (2007, Janeiro 28). [Online]. Disponível em http://en.wikipedia.org/wiki/Waterfall_development
- [11] *Storyboards*. (2007, Janeiro 23). [Online]. Disponível em <http://en.wikipedia.org/wiki/Storyboards>

APÊNDICES

Apêndice 1 – Gráfico de Gantt do projecto

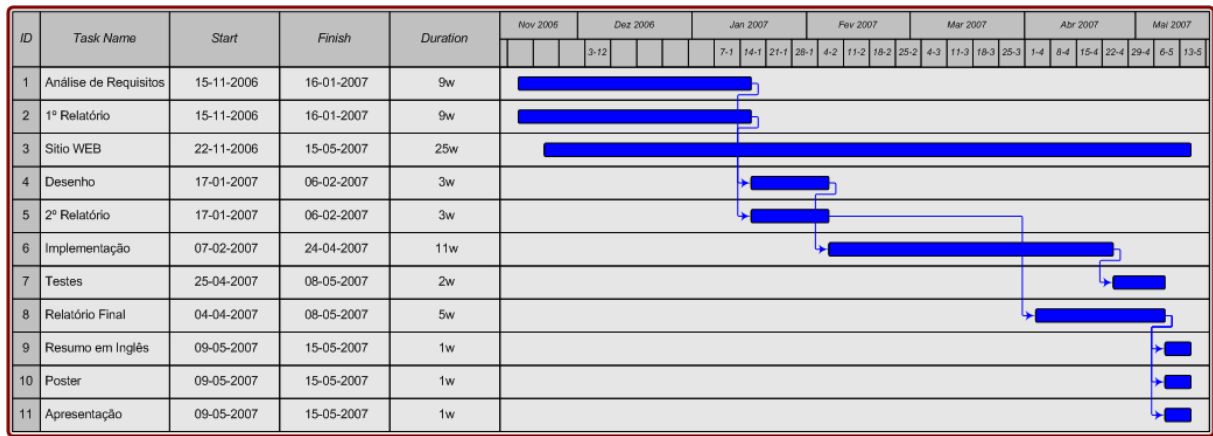


Figura 1.1 – Gráfico de Gantt do projecto

Apêndice 2 – Templates dos Casos de Uso

Tabela 1 – Template para o Caso de Uso “Criar nova interface”

Descrição do Caso de Uso	
Nome	Criar nova interface
Versão	1.0
Autor	Cláudia Oliveira / Cláudio Pedro / Nuno Coelho
Última actualização	08/01/2007
Actores	Utilizador
Outros Casos de Uso relacionados	<ul style="list-style-type: none"> • Guardar interface.
Pré-condições	<ul style="list-style-type: none"> • Ter a aplicação em execução.
Cenário para o Caso de Uso	
Inicialização do Caso de Uso	<ol style="list-style-type: none"> 1. Optar por “New interface” no menu da aplicação; 2. Caso já exista uma interface na área de desenho, permite guardar a mesma.
Descrição da execução do Caso de Uso	<ol style="list-style-type: none"> 1. Eliminação de todos os dados relacionados com controlos visuais, que existiam na memória volátil.
Conclusão do Caso de Uso	<ol style="list-style-type: none"> 1. Inicialização da área de desenho sem controlos visuais.
Pós-condições	<ul style="list-style-type: none"> • Constatar visualmente a área de desenho inicializada.
Ambiguidades	---
Possíveis melhorias	---

Tabela 2 – Template para o Caso de Uso “Guardar interface”

Descrição do Caso de Uso	
Nome	Guardar interface
Versão	1.0
Autor	Cláudia Oliveira / Cláudio Pedro / Nuno Coelho
Última actualização	08/01/2007
Actores	Utilizador
Outros Casos de Uso relacionados	<ul style="list-style-type: none"> • Criar nova interface; • Abrir interface.
Pré-condições	<ul style="list-style-type: none"> • Existir pelo menos um controlo visual na área de desenho.
Cenário para o Caso de Uso	
Inicialização do Caso de Uso	<ol style="list-style-type: none"> 1. Optar por “Save interface...” no menu da aplicação.
Descrição da execução do Caso de Uso	<ol style="list-style-type: none"> 1. Geração do <i>output</i> YAML que descreve a interface, a partir dos dados existentes na memória volátil; 2. Geração do <i>output</i> HTML com código JavaScript, que permite apresentar a interface num <i>browser</i>, a partir dos dados existentes na memória volátil.
Conclusão do Caso de Uso	<ol style="list-style-type: none"> 1. Guardar <i>output</i> YAML gerado, num ficheiro formatado segundo a linguagem YAML; 2. Guardar <i>output</i> HTML com código JavaScript, num ficheiro formatado segundo a linguagem HTML.
Pós-condições	<ul style="list-style-type: none"> • Existência do ficheiro YAML no dispositivo escolhido para o seu armazenamento; • Existência do ficheiro HTML no dispositivo escolhido para o seu

	armazenamento.
Ambiguidades	<ul style="list-style-type: none"> • Memória volátil disponível ser insuficiente para a geração do <i>output</i> YAML; • Memória volátil disponível ser insuficiente para a geração do <i>output</i> HTML com código JavaScript; • Espaço disponível no dispositivo escolhido para o armazenamento do ficheiro YAML ser insuficiente; • Espaço disponível no dispositivo escolhido para o armazenamento do ficheiro HTML ser insuficiente.
Possíveis melhorias	---

Tabela 3 – Template para o Caso de Uso “Abrir interface”

Descrição do Caso de Uso	
Nome	Abrir interface
Versão	1.0
Autor	Cláudia Oliveira / Cláudio Pedro / Nuno Coelho
Última actualização	08/01/2007
Actores	Utilizador
Outros Casos de Uso relacionados	<ul style="list-style-type: none"> • Guardar interface. • Seleccionar interface.
Pré-condições	<ul style="list-style-type: none"> • Ter a aplicação em execução.
Cenário para o Caso de Uso	
Inicialização do Caso de Uso	<ol style="list-style-type: none"> 1. Optar por “Open interface...” no menu da aplicação; 2. Caso já exista uma interface na área de desenho, permite guardar a mesma.
Descrição da execução do Caso de Uso	<ol style="list-style-type: none"> 1. Eliminação de todos os dados relacionados com controlos visuais que existam na memória volátil; 2. Seleccionar ficheiro de interface; 3. Verificação de que o ficheiro escolhido contém dados que descrevem uma interface.
Conclusão do Caso de Uso	<ol style="list-style-type: none"> 1. Carregamento dos dados existentes no ficheiro, para a memória volátil; 2. Geração dos controlos visuais que constituem a interface na área de desenho, a partir dos dados carregados.
Pós-condições	<ul style="list-style-type: none"> • Existência dos dados carregados na memória volátil; • Constatar visualmente a interface gerada.
Ambiguidades	<ul style="list-style-type: none"> • Adulteração dos conteúdos do ficheiro de interface escolhido; • Memória volátil disponível ser insuficiente para o carregamento dos dados sobre controlos visuais existentes no ficheiro.
Possíveis melhorias	---

Tabela 4 – Template para o Caso de Uso “Seleccionar interface”

Descrição do Caso de Uso	
Nome	Seleccionar interface
Versão	1.0
Autor	Cláudia Oliveira / Cláudio Pedro / Nuno Coelho
Última actualização	08/01/2007
Actores	Utilizador
Outros Casos de Uso relacionados	<ul style="list-style-type: none"> • Abrir interface.
Pré-condições	<ul style="list-style-type: none"> • Optar por “Open interface...” no menu da aplicação.

Cenário para o Caso de Uso	
Inicialização do Caso de Uso	1. Explorar o sistema de ficheiros até encontrar o ficheiro de interface pretendido.
Descrição da execução do Caso de Uso	1. Seleccionar o ficheiro pretendido; 2. Confirmar a selecção.
Conclusão do Caso de Uso	1. Carregamento do caminho pelo sistema de ficheiros até ao ficheiro seleccionado, para a memória volátil.
Pós-condições	<ul style="list-style-type: none"> Existência dos dados que indicam o caminho até ao ficheiro seleccionado, na memória volátil.
Ambiguidades	<ul style="list-style-type: none"> Não encontrar o ficheiro de interface pretendido; Memória volátil disponível ser insuficiente para o carregamento dos dados que indicam o caminho até ao ficheiro seleccionado.
Possíveis melhorias	<ul style="list-style-type: none"> Filtragem que providenciasse para selecção apenas ficheiros cujos conteúdos não foram adulterados e que contém dados que definem uma interface.

Tabela 5 – Template para o Caso de Uso “Inserir controlo visual”

Descrição do Caso de Uso	
Nome	Inserir controlo visual
Versão	1.0
Autor	Cláudia Oliveira / Cláudio Pedro / Nuno Coelho
Última actualização	08/01/2007
Actores	Utilizador
Outros Casos de Uso relacionados	---
Pré-condições	<ul style="list-style-type: none"> Ter a aplicação em execução.
Cenário para o Caso de Uso	
Inicialização do Caso de Uso	1. Escolher o controlo visual a inserir na área desenho.
Descrição da execução do Caso de Uso	1. Arrastar o controlo visual escolhido para a área de desenho; 2. Largar o controlo visual na posição pretendida.
Conclusão do Caso de Uso	1. Carregamento dos dados relacionados com o controlo visual inserido, para a memória volátil.
Pós-condições	<ul style="list-style-type: none"> Constatar visualmente a existência do controlo visual na área de desenho; Existência dos dados relacionados com o controlo visual inserido, na memória volátil.
Ambiguidades	<ul style="list-style-type: none"> Largar o controlo visual fora da área de desenho; Largar o controlo visual sobre outro já existente; Memória volátil disponível ser insuficiente para o carregamento dos dados relacionados com o controlo visual inserido.
Possíveis melhorias	---

Tabela 6 – Template para o Caso de Uso “Mover controlo visual”

Descrição do Caso de Uso	
Nome	Mover controlo visual
Versão	1.0
Autor	Cláudia Oliveira / Cláudio Pedro / Nuno Coelho
Última actualização	08/01/2007
Actores	Utilizador

Outros Casos de Uso relacionados	---
Pré-condições	<ul style="list-style-type: none"> Existir pelo menos um controlo visual na área de desenho.
Cenário para o Caso de Uso	
Inicialização do Caso de Uso	1. Seleccionar o(s) controlo(s) a mover.
Descrição da execução do Caso de Uso	<ol style="list-style-type: none"> Arrastar o(s) controlo(s) escolhido(s) para a posição pretendida; Largar o(s) controlo(s).
Conclusão do Caso de Uso	1. Alteração dos dados associados à posição do(s) controlo(s), existentes na memória volátil.
Pós-condições	<ul style="list-style-type: none"> Constatar visualmente a mudança de posição do(s) controlo(s) na área de desenho.
Ambiguidades	<ul style="list-style-type: none"> Largar o(s) controlo(s) fora da área de desenho; Largar o(s) controlo(s) sobre um outro controlo.
Possíveis melhorias	---

Tabela 7 – Template para o Caso de Uso “Apagar controlo visual”

Descrição do Caso de Uso	
Nome	Apagar controlo visual
Versão	1.0
Autor	Cláudia Oliveira / Cláudio Pedro / Nuno Coelho
Última actualização	08/01/2007
Actores	Utilizador
Outros Casos de Uso relacionados	---
Pré-condições	<ul style="list-style-type: none"> Existir pelo menos um controlo visual na área de desenho.
Cenário para o Caso de Uso	
Inicialização do Caso de Uso	1. Seleccionar o(s) controlo(s) a apagar.
Descrição da execução do Caso de Uso	<ol style="list-style-type: none"> Optar por “Delete” no menu acessível através do botão direito do rato ou premir a tecla “Del” existente no teclado; Confirmar que pretende apagar o(s) controlo(s).
Conclusão do Caso de Uso	1. Eliminação de todos os dados relacionados com o(s) controlo(s), que existiam na memória volátil.
Pós-condições	<ul style="list-style-type: none"> Constatar visualmente a remoção do(s) controlo(s) da área de desenho.
Ambiguidades	---
Possíveis melhorias	---

Tabela 8 – Template para o Caso de Uso “Formatar controlo visual”

Descrição do Caso de Uso	
Nome	Formatar controlo visual
Versão	1.0
Autor	Cláudia Oliveira / Cláudio Pedro / Nuno Coelho
Última actualização	08/01/2007
Actores	Utilizador
Outros Casos de Uso relacionados	---
Pré-condições	<ul style="list-style-type: none"> Existir pelo menos um controlo visual na área de desenho.
Cenário para o Caso de Uso	

Inicialização do Caso de Uso	1. Seleccionar um controlo visual ou mais do que um mas do mesmo tipo.
Descrição da execução do Caso de Uso	1. Alterar características visuais; 2. Confirmar alterações nas características.
Conclusão do Caso de Uso	1. Alteração dos dados associados às características visuais alteradas, existentes na memória volátil; 2. Realização das alterações visuais no(s) controlo(s).
Pós-condições	• Constatar visualmente as alterações efectuadas.
Ambiguidades	• Memória volátil disponível ser insuficiente para o armazenamento das alterações efectuadas.
Possíveis melhorias	---

Tabela 9 – Template para o Caso de Uso “Guardar template de controlo visual”

Descrição do Caso de Uso	
Nome	Guardar template de controlo visual
Versão	1.0
Autor	Cláudia Oliveira / Cláudio Pedro / Nuno Coelho
Última actualização	08/01/2007
Actores	Utilizador
Outros Casos de Uso relacionados	---
Pré-condições	• Existir pelo menos um controlo visual na área de desenho.
Cenário para o Caso de Uso	
Inicialização do Caso de Uso	1. Seleccionar apenas um controlo visual; 2. Optar por “Save template...” no menu acessível através do botão direito do rato.
Descrição da execução do Caso de Uso	1. Geração do <i>output</i> YAML que caracteriza o controlo visual seleccionado, de acordo com a formatação que apresenta, a partir dos dados existentes na memória volátil sobre a caracterização do mesmo.
Conclusão do Caso de Uso	1. Guardar <i>output</i> gerado, num ficheiro formatado segundo a linguagem YAML.
Pós-condições	• Armazenamento do ficheiro no dispositivo escolhido.
Ambiguidades	• Memória volátil disponível ser insuficiente para a geração do <i>output</i> YAML; • Espaço disponível no dispositivo escolhido ser insuficiente para o armazenamento do ficheiro.
Possíveis melhorias	---

Tabela 10 – Template para o Caso de Uso “Aplicar template sobre controlo visual”

Descrição do Caso de Uso	
Nome	Aplicar template sobre controlo visual
Versão	1.0
Autor	Cláudia Oliveira / Cláudio Pedro / Nuno Coelho
Última actualização	08/01/2007
Actores	Utilizador
Outros Casos de Uso relacionados	• Seleccionar template.
Pré-condições	• Existir pelo menos um controlo visual na área de desenho.
Cenário para o Caso de Uso	

Inicialização do Caso de Uso	<ol style="list-style-type: none"> 1. Seleccionar um controlo visual ou mais do que um, mas do mesmo tipo; 2. Optar por “Apply template...” no menu acessível através do botão direito do rato.
Descrição da execução do Caso de Uso	<ol style="list-style-type: none"> 1. Seleccionar ficheiro de template; 2. Verificação de que o ficheiro escolhido contém dados que descrevem uma template; 3. Verificação da compatibilidade da template existente no ficheiro escolhido com o(s) controlo(s) seleccionados.
Conclusão do Caso de Uso	<ol style="list-style-type: none"> 1. Carregamento dos dados existentes no ficheiro, para a memória volátil; 2. Formatação do(s) controlo(s) seleccionado(s), de acordo com as caracterizações existentes nos dados carregados.
Pós-condições	<ul style="list-style-type: none"> • Existência dos dados carregados na memória volátil; • Constatar visualmente a formatação do(s) controlo(s) de acordo com a template seleccionada.
Ambiguidades	<ul style="list-style-type: none"> • Ficheiro de template escolhido encontrar-se adulterado; • Incompatibilidade da template descrita no ficheiro com o(s) controlo(s) seleccionado(s); • Memória volátil disponível ser insuficiente para o carregamento dos dados existentes no ficheiro escolhido.
Possíveis melhorias	---

Tabela 11 – Template para o Caso de Uso “Abrir template de controlo visual”

Descrição do Caso de Uso	
Nome	Abrir template de controlo visual
Versão	1.0
Autor	Cláudia Oliveira / Cláudio Pedro / Nuno Coelho
Última actualização	08/01/2007
Actores	Utilizador
Outros Casos de Uso relacionados	<ul style="list-style-type: none"> • Seleccionar template.
Pré-condições	<ul style="list-style-type: none"> • Ter a aplicação em execução.
Cenário para o Caso de Uso	
Inicialização do Caso de Uso	<ol style="list-style-type: none"> 1. Seleccionar um controlo visual ou mais do que um, mas do mesmo tipo; 2. Optar por “Open template...” no menu da aplicação.
Descrição da execução do Caso de Uso	<ol style="list-style-type: none"> 1. Seleccionar ficheiro de template; 2. Verificação de que o ficheiro escolhido contém dados que descrevem uma template.
Conclusão do Caso de Uso	<ol style="list-style-type: none"> 1. Carregamento dos dados existentes no ficheiro, para a memória volátil; 2. Geração na área de desenho, do controlo visual ao qual se adequa a template carregada; 3. Formatação do controlo gerado, de acordo com as caracterizações existentes nos dados carregados.
Pós-condições	<ul style="list-style-type: none"> • Existência dos dados carregados na memória volátil; • Constatar visualmente o controlo visual existente na área de desenho.
Ambiguidades	<ul style="list-style-type: none"> • Ficheiro de template escolhido encontrar-se adulterado; • Memória volátil disponível ser insuficiente para o carregamento dos dados existentes no ficheiro escolhido.
Possíveis melhorias	---

Tabela 12 – Template para o Caso de Uso “Seleccionar template”

Descrição do Caso de Uso	
Nome	Seleccionar template
Versão	1.0
Autor	Cláudia Oliveira / Cláudio Pedro / Nuno Coelho
Última actualização	08/01/2007
Actores	Utilizador
Outros Casos de Uso relacionados	<ul style="list-style-type: none"> • Aplicar template sobre controlo visual; • Abrir template de controlo visual.
Pré-condições	<ul style="list-style-type: none"> • Existir um controlo visual seleccionado ou mais do que um mas do mesmo tipo; • Optar por “Apply template...” no menu acessível através do botão direito do rato, ou “Open template...” no menu da aplicação.
Cenário para o Caso de Uso	
Inicialização do Caso de Uso	1. Explorar o sistema de ficheiros até encontrar o ficheiro de template pretendido.
Descrição da execução do Caso de Uso	<ol style="list-style-type: none"> 1. Seleccionar o ficheiro pretendido; 2. Confirmar a selecção.
Conclusão do Caso de Uso	2. Carregamento do caminho pelo sistema de ficheiros até ao ficheiro escolhido, para a memória volátil.
Pós-condições	<ul style="list-style-type: none"> • Existência dos dados que indicam o caminho até ao ficheiro seleccionado, na memória volátil.
Ambiguidades	<ul style="list-style-type: none"> • Não encontrar o ficheiro de template pretendido; • Memória volátil disponível ser insuficiente para o carregamento dos dados que indicam o caminho até ao ficheiro seleccionado.
Possíveis melhorias	<ul style="list-style-type: none"> • Filtragem que providenciasse para selecção apenas ficheiros cujos conteúdos não foram adulterados e que contém dados que definem uma template.

Tabela 13 – Template para o Caso de Uso “Pré-visualizar interface na aplicação”

Descrição do Caso de Uso	
Nome	Pré-visualizar interface na aplicação
Versão	1.0
Autor	Cláudia Oliveira / Cláudio Pedro / Nuno Coelho
Última actualização	08/01/2007
Actores	Utilizador
Outros Casos de Uso relacionados	---
Pré-condições	<ul style="list-style-type: none"> • Existir uma interface na área de desenho.
Cenário para o Caso de Uso	
Inicialização do Caso de Uso	1. Optar por “Preview in application” no menu da aplicação.
Descrição da execução do Caso de Uso	1. Geração de uma interface aplicacional a partir dos dados presentes na memória volátil sobre a interface existente na área de desenho.
Conclusão do Caso de Uso	1. Apresentação da interface gerada numa janela de pré-visualização.
Pós-condições	<ul style="list-style-type: none"> • Observar a interface na janela de pré-visualização.
Ambiguidades	<ul style="list-style-type: none"> • Memória volátil disponível ser insuficiente para a geração da interface aplicacional.

Possíveis melhorias	---
---------------------	-----

Tabela 14 – Template para o Caso de Uso “Pré-visualizar interface num *browser*”

Descrição do Caso de Uso	
Nome	Pré-visualizar interface num <i>browser</i>
Versão	1.0
Autor	Cláudia Oliveira / Cláudio Pedro / Nuno Coelho
Última actualização	08/01/2007
Actores	Utilizador
Outros Casos de Uso relacionados	---
Pré-condições	<ul style="list-style-type: none"> Existir uma interface na área de desenho; Existir um <i>browser</i> instalado no sistema em que a aplicação está a ser executada.
Cenário para o Caso de Uso	
Inicialização do Caso de Uso	1. Optar por “Preview in browser” no menu da aplicação.
Descrição da execução do Caso de Uso	1. Geração do <i>output</i> HTML com código JavaScript, a partir dos dados existentes na memória volátil sobre a interface existente na área de desenho.
Conclusão do Caso de Uso	1. Apresentação do <i>output</i> gerado, no <i>browser</i> por defeito existente no sistema em que a aplicação está sendo executada.
Pós-condições	<ul style="list-style-type: none"> Observar a interface no <i>browser</i>.
Ambiguidades	<ul style="list-style-type: none"> Não existir nenhum <i>browser</i> instalado no sistema; Memória volátil disponível ser insuficiente para a geração do <i>output</i> HTML com código JavaScript.
Possíveis melhorias	<ul style="list-style-type: none"> Permitir, caso exista mais do que um <i>browser</i> instalado, seleccionar em qual a interface vai ser apresentada.