



Sistema de Construção Visual de Interfaces

Instituto Politécnico de Beja

Escola Superior de Tecnologia e Gestão

Cláudia Oliveira, N.º 3459 - Cláudio Pedro, N.º 3805 - Nuno Coelho, N.º 3938



Introdução

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. Existem hoje em dia inúmeras ferramentas do tipo IDE, 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.

Objectivo do projecto

Este projecto teve 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.

Características da aplicação

- ✓ Construção de interfaces gráficas.
- ✓ Editor de propriedades de controlos visuais.
- ✓ Armazenamento de interfaces gráficas.
- ✓ Criação de templates de controlos visuais formatados.
- ✓ Geração de páginas HTML à imagem das interfaces.

Metodologia

No desenvolvimento desta ferramenta foi adoptado o modelo cascata.

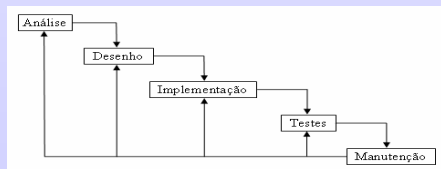


Figura 1 - Modelo cascata

➤ **Análise** – Nesta fase estudou-se o problema de forma a identificar as características que a ferramenta a desenvolver deveria ter.

➤ **Desenho** – Nesta fase desenhou-se o sistema de forma a identificar módulos de implementação e suas características.

➤ **Implementação** – Nesta fase implementaram-se os módulos identificados na fase de desenho.

➤ **Testes** – Nesta fase procurou-se verificar que os requisitos detectados durante a fase de análise foram complementados.

➤ **Manutenção** – Esta fase não foi tomada em conta durante a realização deste projecto.

Análise

- ✓ Estudaram-se ferramentas análogas de forma a reter as principais características a implementar.
- ✓ Definiram-se as tecnologias a utilizar.
- ✓ Elaborou-se o Diagrama de Casos de Uso da aplicação.
- ✓ Preencheram-se as templates dos mesmos.
- ✓ Descreveram-se três possíveis cenários de utilização para a aplicação.



Figura 2 – Diagrama de Casos de Uso

Desenho

- ✓ Elaborou-se o Diagrama de Classes.
- ✓ Elaboraram-se os Diagramas de Transição.
- ✓ Desenharam-se protótipos de baixa fidelidade da aplicação, dos quais fazem parte a janela da aplicação, que inclui as janelas de controlos e de propriedades, as janelas de abrir e guardar, e as janelas de diálogo para confirmação de acções.
- ✓ Realizou-se uma avaliação heurística dos protótipos de baixa fidelidade elaborados e consequentes refinações sobre os mesmos.



Figura 3 – Protótipos de baixa fidelidade

Instalações necessárias

Para o desenvolvimento da aplicação, tiveram que ser efectuadas as seguintes instalações e pela seguinte ordem:

- ✓ Tortoise SVN 1.4.1 7992
- ✓ Python 2.4.4
- ✓ Qt Open Source 4.1.4
- ✓ PyQt 4.0.1
- ✓ PyYaml 3.03

Repositório

Através do Google Code foi utilizado um serviço denominado de *Project Hosting*, que oferece uma plataforma de controlo de versões e de identificação de *bugs* sobre projectos de cariz *open-source*. Esta plataforma de alojamento disponibiliza várias facilidades:

- Espaço de armazenamento virtual para alojamento de ficheiros associados ao projecto;
- Controlo de acessos;
- Controlo de versões sobre os ficheiros existentes no repositório de dados;
- Envio de relatórios para a *mailing list* do projecto.

Conclusão

Este projecto permitiu acima de tudo aprofundar conhecimentos ao nível das aplicações que ajudam os utilizadores a desenvolver mais facilmente interfaces. Permitiu ainda, aplicar conhecimentos adquiridos em diversas disciplinas leccionadas durante o curso de engenharia de informática, como são os casos de Engenharia de Software, Pesquisa e Optimização, Multimédia e Sistemas Interactivos e Interação Pessoa-Computador. 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 e a criação e aplicação de templates. Relativamente ao desenvolvimento do sistema, a fase de análise, foi aquela com que mais nos preocupámos pois a mesma iria definir todo o restante trabalho. A fase de desenho, por sua vez foi a que definiu o aspecto e o funcionamento para o qual a nossa aplicação iria convergir. A fase mais prolongada foi claramente, a fase de implementação. A fase mais curta em termos de duração foi a fase de testes, onde a aplicação foi testada por três utilizadores completamente estranhos à aplicação. Em suma, este projecto foi extremamente aliciante e interessante devido à sua abrangência de conhecimentos diversificada, sendo que será igualmente interessante seguir o processo evolutivo da aplicação desenvolvida, bem como, o aumento das suas potencialidades e funcionalidades num futuro próximo.

Implementação

Após a especificação dos requisitos e a modelação do sistema, seguiu-se a fase de implementação, que consistiu na codificação dos vários módulos de implementação.

- ✓ **Recolha de informação sobre os controlos visuais a disponibilizar ao utilizador, para a construção das interfaces e escolha dos mesmos.** Foi também efectuado um estudo para definir quais os controlos semelhantes entre os controlos do qooxdoo e do Qt.

Controlos implementados					
Button	Check Box	Combo Box	Group Box	IFrame	Label
List	MenuBar	Password Field	Radio Button	Spinner	TabView
Table	Text Area	Text Field	ToolBar	Tree	

Figura 4 – Controlos implementados

- ✓ **Construção da interface gráfica da aplicação e implementação de mecanismos para interacção com os controlos visuais disponíveis.** Esta interface gráfica foi desenvolvida com base nos protótipos de baixa fidelidade desenhados durante a fase de modelação do sistema. Cada controlo visual é representado por uma classe específica, que permite a interacção com a aplicação.

- ✓ **Construção de um gerador de código HTML** que incluirá uma porção de código JavaScript, onde, através de comandos qooxdoo, se encontrará descrita a interface criada. O gerador de código HTML é o “motor” que proporciona ao utilizador a visualização da interface, num *browser*.

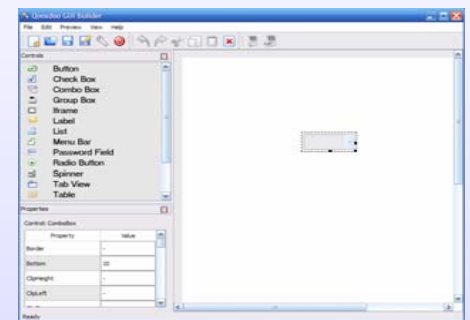


Figura 5 - Janela da aplicação

- ✓ **Construção de um interpretador e gerador de código YAML** que permite o armazenamento e carregamento das interfaces e templates criadas pelo utilizador. Na aplicação desenvolvida o utilizador terá hipótese de criar interfaces e templates que poderá depois armazenar numa unidade de armazenamento à sua escolha, para posteriormente visualizar ou alterar. A extensão dos ficheiros que guardam interfaces em disco é “.yaml” e a dos ficheiros que guardam templates é “.ymtl”.

Testes

Durante a fase de implementação do sistema foram sendo sistematicamente efectuados testes com o objectivo de cumprir com todos os requisitos inicialmente identificados. Ainda assim, com a implementação terminada, foram efectuados novamente testes de validação ao sistema, com três utilizadores estranhos à aplicação que testaram as funcionalidades e o desempenho da mesma.