

# Sistema de Construção Visual de Interfaces

## Escola Superior de Tecnologia e Gestão de Beja

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

Junho de 2007

### *Resumo*

Este documento representa uma abordagem ao que foi feito pelos alunos durante o desenvolvimento de um sistema de construção visual de interfaces.

### I – INTRODUÇÃO

Este projecto teve como objectivo o desenvolvimento de uma ferramenta gráfica que sirva de suporte à construção de interfaces compostas por elementos do *toolkit* gráfico qooxdoo [1]. Pretende-se com esta ferramenta diminuir o tempo gasto pelos programadores na construção das interfaces, facilitando-lhes ainda a construção das mesmas. 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. No desenvolvimento desta ferramenta recorreu-se ao *toolkit* gráfico Qt (Dalheimer, 2002) [2] 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 no *toolkit* qooxdoo.

### II – METODOLOGIA

No desenvolvimento desta ferramenta foi adoptado o modelo cascata [3] que é constituído por cinco fases distintas. *Análise*, na qual ferramentas análogas foram estudadas e as características a implementar foram identificadas. *Desenho*, onde foram identificados

os módulos da implementação, as suas características e os tempos de desenvolvimento previstos. Implementação durante a qual foram implementados os módulos identificados na fase anterior. *Testes*, que serviu para verificar que os requisitos detectados durante a fase de análise foram implementados, e *Manutenção*, que não foi considerada durante a realização deste projecto.

### III – TECNOLOGIAS UTILIZADAS

No desenvolvimento da aplicação utilizou-se a linguagem de programação Python, o *toolkit* gráfico Qt, as linguagens de codificação YAML e HTML, a linguagem de programação *JavaScript*, as tecnologias AJAX [4] e a *framework* qooxdoo.

### IV – TRABALHO DESENVOLVIDO

O desenvolvimento da aplicação foi iniciado procurando na *Web* por ferramentas análogas e estudando-as. Depois fez-se o Diagrama de Casos de Uso, preencheram-se as templates do mesmo, e descreveram-se três possíveis

cenários de utilização para a aplicação.

Numa segunda fase o Diagrama de Classes e os Diagramas de Transição foram elaboradas, e os protótipos de baixa fidelidade de interfaces gráficas da aplicação foram desenhados. Esses protótipos incluíam a janela da aplicação, bem como, janelas de diálogo para confirmação de acções. Durante o desenvolvimento destes protótipos, foi feita uma avaliação heurística aos mesmos, baseada nas 10 heurísticas de Nielsen [5].

Com a especificação dos requisitos e a modelação do sistema terminadas, chegou a fase de implementação. Esta fase consistiu na codificação dos seguintes quatro 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

- controles do qooxdoo e do Qt;
- **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*;
  - **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 “.ymli” e a dos ficheiros que guardam templates é “.ymlt”.

## V – CONCLUSÃO

Este projecto permitiu acima de tudo aprofundar conhecimentos ao nível das aplicações que facilitam o desenvolvimento de interfaces gráficas. Permitiu ainda, aplicar conhecimentos adquiridos em diversas disciplinas leccionadas durante o curso de engenharia de informática. 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.

## VI – BIBLIOGRAFIA

- [1] *qooxdoo*. (2006, Dezembro 22). [Online]. Disponível em <http://qooxdoo.org>
- [2] Matthias Kalle Dalheimer, *Programming with Qt, 2nd Edition*, O'Reilly, Janeiro 2002, capítulo 1
- [3] *Waterfall model*. (2007, Janeiro 28). [Online]. Disponível em [http://en.wikipedia.org/wiki/Waterfall\\_development](http://en.wikipedia.org/wiki/Waterfall_development)
- [4] *Ajax (programação)*. (2007, Janeiro 14). [Online]. Disponível em [http://pt.wikipedia.org/wiki/AJAX\\_%28programa%C3%A7%C3%A3o%29](http://pt.wikipedia.org/wiki/AJAX_%28programa%C3%A7%C3%A3o%29)
- [5] *Heuristic Evaluation*. (2005). [Online]. Disponível em <http://www.useit.com/papers/heuristic>