

Universidade Federal da Bahia
Instituto de Matemática
Departamento de Ciência da Computação

Especificação da Projeto de Laboratório de Programação
Parte 1

Salvador, 2009

Universidade Federal da Bahia
Instituto de Matemática
Departamento de Ciência da Computação

Especificação da Projeto de Laboratório de Programação
Parte 1

Eduardo Ramos dos Santos Júnior
Maurício Souza Lage

Salvador, 2009

1. Qual foi a motivação dos desenvolvedores para construirem este produto?

O Qt é um framework bastante utilizado atualmente, sendo a principal ferramenta de construção de GUI's de projetos grandes, como o KDE. Além disso, o framework conta com excelente documentação e IDE's específicas, que agilizam bastante o desenvolvimento (como o QtCreator, Designer, etc). Além disso, a ferramenta é famosa pelo rápido aprendizado em seu uso.

2. Quais as principais características, capacidades e restrições deste framework?

O Qt foi criado pela empresa norueguesa Trolltech. O framework usa o paradigma de orientação a objetos, sendo principalmente usada para programação na linguagem C++, porém existem extensões (bindings) para outras linguagens, como Python, Ruby e Perl

O Qt é um framework de desenvolvimento multiplataforma, ou seja, o código precisa ser escrito apenas uma vez e compilado para a plataforma específica. O Qt consegue lidar com as API's para diversos sistemas.

Além disso, podemos citar a sua portabilidade entre os diversos tipos de Sistemas Operacionais e a possibilidade de ser integrado com ferramentas de desenvolvimento facilmente, tornando a programação mais rápida e eficiente

Também foi motivo da escolha deste Framework e que ficou implícito acima é a sua alta performance em tempo de execução, possibilitando que aplicações desenvolvidas com este Framework tenham um baixo tempo de resposta, implicando num alto desempenho.

Além das restrições de certas combinações de Sistemas Operacionais com compiladores específicos não suportarem Qt, temos as restrições geradas por versões específicas de compiladores, que levam a erros de compilação.

Segundo notas de compilação, na documentação oficial do Qt, o uso do compilador GCC 4.0.0 gera a diversos erros de compilação.

Além disso, existem diversos *bindings* para diversas linguagens de programação, levando a idiossincrasias da linguagem para o qual o Qt foi portado. São exemplos: QtRuby, QtAda, PyQt.

3. Sob qual licença ele se encontra? (Proprietária, GPL, QPL, Creative Commons, etc.)

O Qt está disponível para ser usada sob as licenças GPL v3.0, LGL v2.1, além de uma versão comercial. Estaremos utilizando a licença **GPL** durante o desenvolvimento do projeto.

3.1 Quais as principais restrições e exigências desta licença?

Como é uma versão disponível livremente para o desenvolvimento de aplicações, não existem restrições quanto para qual tipo de Sistema Operacional o software será desenvolvido ou para qual funcionalidade.

A única exigência é que caso deseje-se tirar vantagem comercial, é necessário manter os princípios que regem o GPL v. 3.0 pois caso contrário, será necessário adquirir um apropriado número de licenças comerciais

3.2 Quais as obrigações do desenvolvedor que utilizar este framework?

Já foi respondido no item anterior.

4. Quais plataformas e sistemas operacionais suportados?

Atualmente, o Qt suporta sistemas Linux/X11, Mac OS X, S60 (smartphones) e Windows. Para maiores detalhes sobre suporte do QT, acessar: <http://doc.trolltech.com/4.5/supported-platforms.html>

5. Quais os procedimentos necessários para se compilar um programa simples em C que utilize este framework?

Os procedimentos necessários para se compilar um programa em C que utilize Qt são (supondo o nome do programa ser main.c):

1. Criar um diretório chamado main
2. Colocar o código-fonte do programa em tal diretório
3. Executar:

```
qmake -project  
qmake main.pro  
make
```

Foi feito dessa forma porque, para facilitar o processo de compilação e testes, é útil ter um arquivo chamado Makefile que de forma automática faz a compilação do seu programa.

Para tal, movemos nosso código-fonte para uma pasta onde ficará armazenado nosso projeto. Em seguida, criamos um projeto Qt com a linha rotulada em 3.1

Continuando, um arquivo chamado main.pro será criado. E para gerar um Makefile específico de arquitetura, executamos a linha 3.2

Assim, temos nosso arquivo Makefile criado e podemos, de forma automática, sempre que atualizarmos nosso código-fonte, compilá-lo e testá-lo. Podemos fazer isso com diversas ferramentas, dependentes de SO, mas a mais conhecida é a *make*.

- 1. Qual será objetivo do software? Qual o problema que o software tenta resolver? (Se um jogo, qual o objetivo do jogo).**

O objetivo é construir um software de bate-papo. Os participantes podem estar em computadores diferentes e interagir através da troca de mensagens com outros participantes.

- 2. Qual será o funcionamento do programa? Pode-se incluir um fluxograma para ilustrar o funcionamento ou outro recurso gráfico desejado.**

Na verdade, o sistema vai ser composto de dois tipos de programa, servidor e cliente.

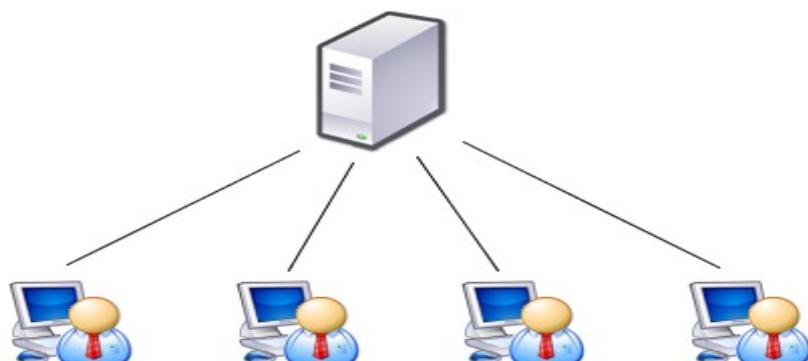
O servidor será um serviço em execução em outro host (dispositivo de redes) com endereçamento e porta bem conhecida entre os clientes. Ele funciona como um ponto central, tendo como principais tarefas:

- receber e tratar pedidos de conexões
- estabelecer conexões
- receber e propagar mensagens recebidas de clientes
- tratar encerramento de conexão.

Pode-se fazer uma analogia com a sala de bate papo da uol ou com as já consagradas sala de bate papo do IRC (Internet Relay Channel)

Cada mensagens recebida por um cliente é encaminhada em broadcast para os demais participantes.

O cliente é o programa que roda no host do usuário do chat. Ele provê a interface gráfica para o usuário. Através dessa interface, o usuário deve utilizar de forma intuitiva as tarefas básicas do bate-papo, como solicitar conexão, definir apelido, enviar mensagens e finalizar sessão.



3. Quantas pessoas são necessárias ao projeto e quais os seus papéis?

São necessários 2(duas) pessoas para desenvolver o projeto.

Na verdade, este valor é variável e depende de diversos outros fatores como: habilidade com programação, disponibilidade de tempo, características a serem implementadas, conhecimento do programador dentre outros.

Mas para uma atividade que tem por objetivo fixar conceitos vistos em sala de aula e expandir os mesmos, a programação em par é uma boa solução, pois a mesma já mostrou ser eficiente por diversos fatores:

- Qualidade do modelo
- Redução do custo de desenvolvimento
- Aprendizado e treinamento
- Diminui o risco de gerenciamento
- Fluxo mais rápido

Isso para citar alguns dos benefícios.

Já existem diversos artigos que ressaltam o ganho de produtividade e de compartilhamento do conhecimento quando se programa em par.

Alguns deles podem ser acessados através da seção Referências em:
http://en.wikipedia.org/wiki/Pair_programming

Logo, na programação em par não haverá definição formal do papel de cada um.

O par se reunirá para codificar, um dia alguém será o responsável por codificar de fato, fazer a interface entre o computador e as idéias, no próximo dia haverá essa troca e essa rotatividade será seguida até a conclusão do trabalho.

4. Qual o cronograma de desenvolvimento?

Atividades	Março	Abril	Maio	Junho
Definição do tema do projeto	X			
Definição do escopo do projeto		X		
Estudo sobre C++ e Qt		X	X	
Desenvolvimento do Projeto		X	X	X
Realização de Testes			X	X

5. Qual a forma e ferramentas que se pretende utilizar no desenvolvimento?

Inicialmente, todo o desenvolvimento será feito em cima de uma plataforma livre porque além de favorecer a liberdade no acesso a informação é a única plataforma disponível na Universidade. Além disso, será utilizada a versão mais atual do Qt e também será utilizado o compilador GCC também em sua versão mais atual. Sempre tomando-se o cuidado de verificar se existe algum tipo de restrição entre versão do Qt, do compilador naquele Sistema Operacional.

Além disso, temos os editores de texto, que não fugirá do já consagrado e eficiente Vi. O make, construtor automático de binários, a grosso modo. E se necessário, alguma ferramenta de debug, como gdb e o valgrind.