UNIVERSIDADE FEDERAL DE CAMPINA GRANDE

CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA

DEPARTAMENTO DE SISTEMAS E COMPUTAÇÃO

# RELATÓRIO DE ESTÁGIO

# QT-BRISA

DANILO ARAÚJO DE FREITAS

Estagiário

ANGELO PERKUSICH

Orientador Acadêmico

THIAGO BRUNO MELO DE SALES

Supervisor Técnico

**Campina Grande – PB**

Campina Grande – PB

Dezembro de 2010

QT-BRISA

APROVADO EM _____

BANCA EXAMINADORA

Prof. Dr. Angelo Perkusich

ORIENTADOR ACADÊMICO

Joseana Macêdo Fechine

Prof Dr. Joseana Macêdo Fechine

MEMBRO DA BANCA

Hyggo Almeida de Almeida

Prof. Dr. Hyggo Almeida

MEMBRO DA BANCA


Biblioteca Setorial do CDSA. Maio de 2021.

Sumé - PB

# AGRADECIMENTOS

Agradeço, primeiramente, à empresa Spry Softwares e a Leandro Melo de Sales pela oportunidade de estágio. Agradeço também ao professor Angelo Perkusich, por ter me orientado durante o período de estágio, e a professora Joseana Macêdo Fechine, por ter dado as diretrizes para a realização do mesmo, e ao membro convidado da banca, professor Hyggo Almeida, e a todos os outros que, de alguma forma, puderam contribuir com o meu estágio.

# APRESENTAÇÃO

Segue-se o relatório de estágio do aluno Danilo Araújo de Freitas, como parte das exigências do curso de Ciência da Computação da Universidade Federal de Campina Grande.

O estágio foi realizado na empresa Spry Sofwares, com contrato de seis meses (de agosto de 2010 a janeiro de 2011).

Neste relatório serão detalhadas as atividades realizadas pelo estagiário, além da base técnica e teórica necessária para a realização dos mesmos. Serão descritos também os resultados obtidos em seu trabalho.

Este relatório foi dividido da seguinte forma:

Seção 1 – introdução.

Seção 2 – Ambiente de Estágio.

Seção 3 – Fundamentação Teórica e Tecnologias Utilizadas.

Seção 4 – Atividades do Estágio.

Seção 5 – Considerações Finais.

Referências Bibliográficas.

Apêndice.

# RESUMO

O objetivo do estágio, inicialmente, era contribuir para o projeto de automação que está sendo desenvolvido pela empresa utilizando o projeto Qt-BRisa (BRisa, 2010), um *framework* que implementa a especificação UPnP (UPnP Forum, 2010). Porém, durante o período de estágio, novas tarefas foram alocadas, com foco no próprio *framework*. Para o projeto de automação, foram criados alguns dispositivos comuns que provavelmente seriam automatizados (geladeira, ar condicionado, televisão, etc.). Para BRisa, foram realizadas algumas correções de *bugs* encontrados durante o período de estágio, implementação de novas funcionalidades especificadas pelo UPnP que ainda não estavam presentes no *framkework* (adaptação para UPnP 1.1), correção de *memory leaks* (vazamento de memória), refatoramento de código, desenvolvimento de exemplos utilizando o BRisa e manutenção de alguns exemplos já implementados.

# SUMÁRIO

# LISTA DE SIGLAS E ABREVIATURAS

- **UPnP** – *Universal Plug and Play*
- **UDP** – *User Datagram Protocol*
- **SAX** – *Simple API for XML*
- **DOM** – *Document Object Model*
- **INdT** – Instituto Nokia de Tecnologia

# LISTA DE FIGURAS

# LISTA DE QUADROS

# SEÇÃO I

# INTRODUÇÃO

# 1. INTRODUÇÃO

O estágio foi concedido pela empresa Spry Softwares, com contrato de agosto deste ano, até janeiro de 2011, e supervisionado por Thiago Bruno Melo de Sales, sócio da empresa.

Com o aumento da quantidade de pessoas que concluem um curso superior, o mercado de trabalho vem ficando cada vez mais disputado. Dessa forma, jovens que há pouco estavam eram graduandos tentam entrar nessa disputa, mas com ampla desvantagem. Com pouca ou nenhuma experiência, não são prioridades no preenchimento de vagas em empresas.

Tendo em mente esse problema, vários jovens buscam por oportunidades de estágio em empresas de todos os portes. O estágio pode ser considerado um benefício mútuo, pois, o jovem ganhará experiência que o ajudará a enfrentar os desafios do mercado de trabalho, e a empresa ganhará uma mão de obra relativamente barata. Como o período de estágio é curto (geralmente seis meses), não há grandes riscos para a empresa caso aconteça algo com o estagiário, nem para o próprio, caso o mesmo não se sinta bem com o trabalho. O estágio também pode ser utilizado pela empresa como um período probatório, podendo garantir a futura contratação do estagiário para um emprego formal.

P o estagiário em questão neste relatório, o estágio realizado está sendo de suma importância para a formação profissional. A partir dele, o mesmo está adquirindo experiência e responsabilidade, além do conhecimento de ferramentas e tecnologias, que poderá ser bastante útil futuramente.

O objetivo do estágio, inicialmente, era desenvolver uma aplicação para automação residencial, utilizando o *framework* BRisa, na versão Qt (Nokia, 2010). Porém, após a realização de algumas atividades para tal aplicação, foi feita uma nova alocação de atividades, dessa vez focadas no próprio *framework*. Dessa forma, o estágio foi focado mais na manutenção do Qt-BRisa.

# SEÇÃO II

# AMBIENTE DE ESTÁGIO

# 2. AMBIENTE DE ESTÁGIO

## *Sobre a empresa*

A empresa Sales Software Ltda., com nome de fantasia Spry Softwares, foi fundada no dia 20 de Fevereiro de 2008, por Leandro Melo de Sales e Mario Hozano Lucas de Souza, ambos bacharéis e mestres em Ciência da Computação pela Universidade Federal de Alagoas (UFAL) e pela Universidade Federal de Campina Grande (UFCG). Ambos foram bolsistas de projetos de pesquisa com cooperação entre UFCG e o INdT no Laboratório de Computação de Sistemas Embarcados e Computação Pervasiva (Embedded) da UFCG. A demanda crescente no INdT por soluções inovadoras aliado ao conhecimento adquirido com os projetos de pesquisa fez com que os sócios criassem a empresa.

Após aproximadamente um ano da fundação, o contrato social foi alterado com a entrada do novo sócio Felipe Barros Pontes, também bacharel em Ciência da Computação pela UFAL e mestre pela UFCG. Durante o mestrado, Felipe atuou em pesquisas relacionadas a sistemas distribuídos pelo Laboratório de Sistemas Distribuídos (LSD) na UFCG, participando como bolsista de um projeto de cooperação entre a UFCG e a empresa multinacional fabricante de computadores HP.

A última mudança no quadro societário da empresa aconteceu com a entrada de Thiago Bruno Melo de Sales no lugar de Mario Hozano Lucas de Souza. Assim como os demais, Thiago também é bacharel em Ciência da Computação pela UFAL e pela UFCG. Além disso, é aluno de doutorado do programa de pós-graduação da UFCG.

Os principais setores e atividade da empresa são o desenvolvimento de programas de computador sob encomenda, consultoria e assessoria em tecnologia da informação e venda de equipamentos e suprimentos de informática.

A Spry Softwares atualmente possui um empregado e dois estagiários.

## *Carga horária*

Foram disponibilizadas 20 horas semanais para a realização das atividades do estágio. Para não haver problemas com as atividades da graduação, o horário pode ser flexível, contanto que o trabalho seja compensado em outro momento futuro.

## *Infraestrutura*

As atividades foram realizadas no Embedded, que possui parceria com a empresa. Porém, algumas atividades puderam ser realizadas na casa do estagiário, dado que não havia total necessidade de presença no laboratório.

No laboratório foi utilizado um computador desktop com sistema operacional Linux (distribuição Ubuntu 10.04), com 2GB de memória principal RAM e disco rígido com capacidade 320GB. Em casa, era utilizado um notebook com sistemas operacionais Windows Vista Home Basic e Linux (distribuição Ubuntu 10.10), com memória RAM principal de 3GB e disco rígido com capacidade de 150GB. Processador Intel Core 2 Duo com *clock* de 2.13GHz.

Para a realização das atividades, foram utilizados basicamente a ferramenta Qt Creator (Nokia, 2010) e o terminal do Linux. Algumas outras ferramentas de apoio também foram utilizadas, como o Valgrind (Valgrind, 2010) e o *control point* do *framework* GUPnP (GUPnP, 2010).

## *Orientação e supervisão*

i) **Orientador Acadêmico**

**Nome:** Angelo Perkusich

**Endereço:** Universidade Federal de Campina Grande, Departamento de Engenharia Elétrica, Laboratório de Sistemas Embarcados e Computação Pervasiva. Bodocongó. 58109970 - Campina Grande, PB - Brasil - Caixa-Postal: 10105

**Email:** perkusic@embedded.ufcg.edu.br

ii) **Supervisor Técnico**

**Nome:** Thiago Bruno Melo de Sales

**Endereço:** Universidade Federal de Campina Grande, Departamento de Engenharia Elétrica, Laboratório de Sistemas Embarcados e Computação Pervasiva. Bodocongó. 58109970 - Campina Grande, PB - Brasil - Caixa-Postal: 10105

**Email:** thiagobrunoms@gmail.com

# SEÇÃO III

# FUNDAMENTAÇÃO TEÓRICA E TECNOLOGIAS UTILIZADAS

# 3. FUNDAMENTAÇÃO TEÓRICA E TECNOLOGIAS UTILIZADAS

É indispensável que o conteúdo inicial desta seção seja sobre UPnP. O *Universal Plug and Play* especifica várias regras para troca de mensagens entre aplicações que utilizam o mesmo. A especificação define algumas palavras chaves, como serviço, dispisitivo, *control point*, ações, etc. Cada definição será detalhada mais adiante.

O UPnP pode ser utilizado de diferentes formas, com diferentes aplicações. Uma delas, talvez a mais utilizada, é a criação de *Media Centres*, onde é possível acessar arquivos multimídia (áudio, vídeo e imagens) em um computador através de outro. Alguns exemplos de sistemas que utilizam UPnP para esse fim são o Windows Media Player e o console PlayStation 3.

Outra aplicação bastante promissora é na automação residencial. Com UPnP é possível criar programas que poderão ser embarcados em dispositivos, como eletrodomésticos, e serem controlados a partir de um computador central, onde deverá existir um *control point* capaz de se comunicar com tais dispositivos.

Para entender UPnP, é preciso conhecer as definições básicas apresentadas pela especificação (UPnP Device Architecture, 2008). Elas são:

- Dispositivo (*device*): Dispositivo em si. Pode ser embarcado em um dispositivo real, como um eletrodoméstico, um computador, etc.
- Serviço (*service*): Serviços providos por um dispositivo. Cada serviço possui um conjunto de ações e de varáveis de estado, tornando possível a comunicação com o dispositivo que possui tal serviço.
- Ação (*action*): Ação a ser utilizada por um ponto de controle, que irá utilizar ou alterar o estado de um dispositivo. Controlará diretamente as variáveis de estado.

- Variáveis de Estado (*State Variables*): Representam o estado de um dispositivo. Podem enviar eventos quando forem alteradas.

- Evento (*event*): Uma variável de estado pode ou não enviar eventos. Um evento é enviado quando o valor de uma variável é alterado. O evento pode ser *unicast* -envia diretamente para os *control points* que se "inscreveram" para receber esse evento (*subscribe*) - ou *multicast* – envia para um endereço fixo e quem estiver inscrito no grupo *multicast* receberá esse evento.

- Ponto de Controle (*Control Point*): Responsável por encontrar e controlar dispositivo, através dos seus serviços e ações.

O funcionamento de um sistema em UPnP é bastante simples de entender. Quando um dispositivo é criado, um *webserver* é criado na rede local para hospedar os arquivos do mesmo. Tais arquivos são do formato XML, que definirão as propriedades do dispositivo, os serviços, ações e variáveis de estado. Após a hospedagem dos arquivos pelo *webserver*, o dispositivo envia mensagem em *multicast* informando que entrou na rede. Os *control points* ativos, que estarão inscritos no grupo *multicast*, receberão a mensagem e irão acessar os arquivos hospedados pelo *webserver*, obtendo as informações necessárias para utilizar seus serviços.

O *control point*, ao ser criado, também envia mensagem em *multicast*, mas para realizar a busca por dispositivos. Essa busca é conhecida como *m-search*. Quando um dispositivo recebe a mensagem do *m-search*, o mesmo tem que responder diretamente ao *control point*. Após isso, o processo é o mesmo do anterior.

Existem várias implemetações do UPnP. O BRisa é um dos *frameworks* que o implementam. O objetivo de um *framework* é facilitar o desenvolvimento de aplicações específicas. No caso do BRisa, o objetivo é facilitar o desenvolvimento de aplicações UPnP através da criação de dispositivos, serviços, *control points*, etc.

O projeto BRisa foi criado e é mantido por brasileiros, principalmente estudantes e professores da UFAL (Universidade Federal de Alagoas) e da

UFCG (Universidade Federal de Campina Grande). O projeto é coordenado por Leandro Melo de Sales, professor da UFAL e doutorando da UFCG.

O BRisa possui duas implementações: uma em python e outra em Qt. A versão implementada em python está estável e não há muito desenvolvimento em cima dela. Já a versão em Qt, apesar de relativamente estável, está em constante desenvolvimento.

Qt é um *framework* de aplicação e UI (*User Interface*) multiplataforma, desenvolvido pela Trolltech, que hoje faz parte da Nokia. Qt facilita o desenvolvimento de aplicações, principalmente na interface gráfica. O *framework* possui várias bibliotecas que auxiliam o desenvolvedor na implementação de sua aplicação. Além disso, possui um ambiente de desenvolvimento, contendo uma IDE, para editar e escrever código, o *designer*, que agiliza o processo de criação de telas, e o *assitant*, contendo toda a documentação de Qt.

Esta ferramenta também possibilita o fácil desenvolvimento de aplicações orientadas a eventos, utilizando os conceitos de *signal*, *slot* e *connect*. *Signal* é o evento em si, *slot* é um método *callback* (é executado após a emissão de um *signal*) e o *connect* é o método responsável por fazer a ligação entre um *signal* e um *slot*. Um *signal* pode ser conectado a vários *slots* e vice-versa.

O *framework* possibilita o desenvolvimento de aplicações para diversas plataformas, como Linux, Windows, Mac OS e Symbian. Dessa forma, é possível desenvolver uma única aplicação e compilá-la para as diferentes plataformas desejáveis. A linguagem utilizada por Qt é C++, mas já existem outras versões, como pyQt (Python) e Jambi (Java).

A versão do BRisa implementada em Qt traz ao usuário uma API simples de ser utilizada, tornando a criação de dispositivos (incluindo a implementação de ações e servicos e manipulação das avariáveis de estado) e de *control points* bastante simples e intuitiva.

A idéia é que essa versão seja compatível com as plataformas mais utilizadas no mercado. Atualmente, ela é compatível com Linux, Maemo e Mac OS. Windows e Symbian poderão ser incluídos nessa lista futuramente.

A ferramenta mais utilizada para o desenvolvimento das atividades foi o Qt Creator, que engloba a IDE do Qt, o *designer* e o *helper*. Para o desenvolvimento de aplicações em Qt, essa talvez seja a melhor ferramenta disponível.

Também foi utilizada a ferramenta Valgrind, responsável por procurar falhas de memória no programa. O Valgrind foi utilizado pelo estagiário na tarefa de busca por *memory leaks* no Qt-BRisa. Porém, pouco sucesso foi obtido com seu uso, pois os *logs* gerados por ela eram muito grandes e pouco relevantes.

# SEÇÃO IV

# ATIVIDADES REALIZADAS

# 4. ATIVIDADES REALIZADAS

Como já foi dito neste relatório, o foco inicial do estágio era a contribuição no projeto de automação residencial, cujo proprietário é a empresa Spry Softwares, utilizando o Qt-BRisa. Porém, atividades relacionadas ao *framework* BRisa foram as prioridades do estagiário.

A partir do Plano de Estágio (em anexo), observa-se que as atividades realizadas foram um pouco diferente das planejadas.

Apesar de maior parte do tempo ter sido dedicado ao *framework*, o projeto de automação está perto de sua conclusão.

A seguir, serão detalhadas as atividades realizadas e os resultados obtidos.

## *Estudo da especificação UPnP e do Qt-BRisa*

É de extrema importância que, ao trabalhar na implementação de uma especificação, o desenvolvedor possua um bom conhecimento da mesma. No caso do Qt-BRisa, além dos conhecimentos de Qt, C++, dentre outros, é necessário um bom conhecimento sobre a especificação UPnP.

No início das atividades, o estagiário possuía pouco conhecimento sobre a especificação, sendo necessário um período de tempo dedicado ao estudo do mesmo. Foi utilizado o documento da especificação UPnP

Para iniciar as atividades de desenvolvimento, é necessário primeiramente conhecer projeto em si. Para isso, foi dedicado um tempo para estudo do código do Qt-BRisa e de sua utilização.

O estudo do UPnP e do Qt-BRisa tomaram demasiado tempo, pois, além de ser algo novo para o estagiário, tal conhecimento seria a base de todas as outras atividades realizadas.

### *Criação de dispositivos*

A tarefa mais relevante que contribuiu para o projeto foi a criação de *devices* para representar os eletrodomésticos que provavelmente serão automatizados.

Foram criados: fogão, luz binária (acende e apaga), luz *fuzzy* (vários níveis de iluminação), geladeira, ar condicionado, televisão, *smartphone*, Notebook, porta eletrônica e computador desktop. Tais dispositivos foram integrados à aplicação de automação e serão utilizados por um *control point* desenvolvido por outro estagiário da empresa.

Com algumas mudanças no Qt-BRisa, os dispositivos tiveram que ser adaptados (em nível de código) às novas versões, pois deixaram de funcionar.

### *Suporte a eventos* multicast

A especificação UPnP 1.1 torna obrigatória o envio de eventos *multicast*, porém, o BRisa ainda não dava esse suporte.

Para isso, foram utilizados *sockets* UDP para envio (pelos *devices*) e recepção (pelos *control points*) dos eventos. Com a forma que foi implementada, é possível filtrar apenas o nome da variável de estado alterada e seu novo valor, ou visualizar toda a mensagem recebida em forma de tabela (ou mapa). Isso foi facilitado pela API de eventos provida pelo Qt.

A Figura 1 mostra um dispositivo informando o envio de um evento *multicast*. Já a Figura 2 mostra um *control point* informando a recepção do evento.

### *Refatoramento de código*

Após a conclusão do suporte a eventos *multicast*, foi necessário um refatoramento, pois havia trechos de código bastante semelhantes em outras partes do projeto. Foi feita então a união desses códigos em uma única classe (BrisaUdpListener), e a mesma foi utilizada nos locais onde havia tal repetição (classes BrisaSSDPServer, BrisaSSDPClient e BrisaMulticastEventReceiver).

Outra parte do refatoramento foi aplicada no *parser* do XML dos dispositivos e dos serviços. Além da mudança de SAX para DOM, que são

duas formas de manipular arquivos XML, o código ficou consideravelmente mais simples e de fácil leitura. Apesar dessas mudanças, a principal motivação para esse trabalho foi o alto consumo de memória pelo *control point*, que será mais detalhado na próxima atividade.

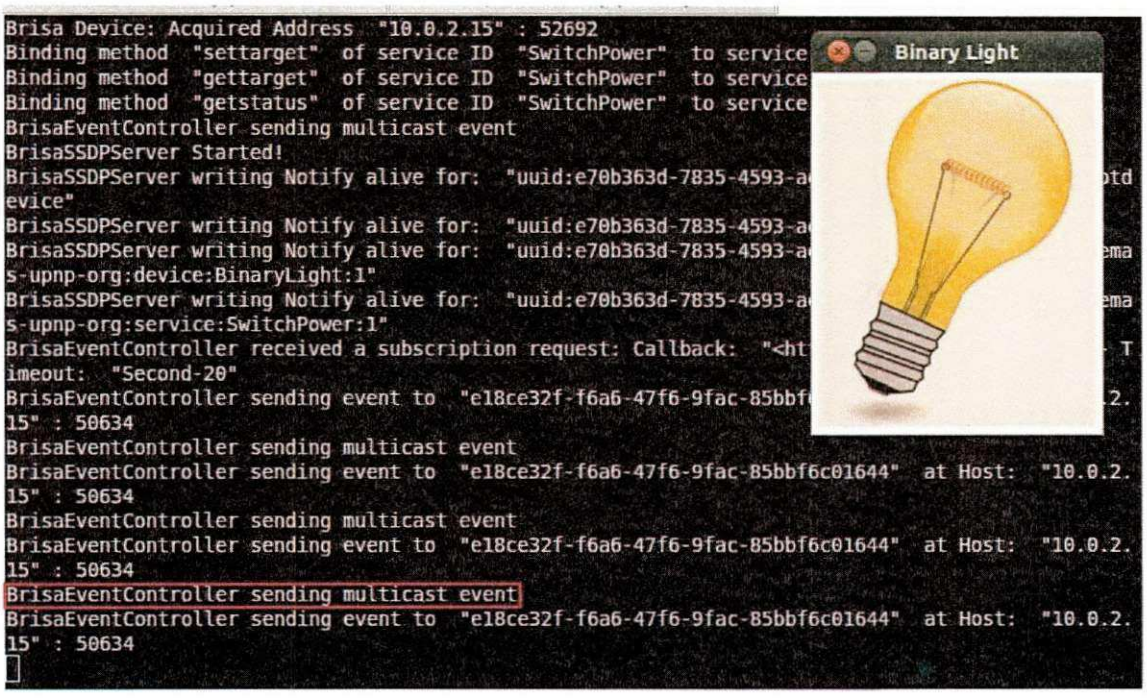


**Figura 1 - Dispositivo informando o envio de um evento *multicast*.**

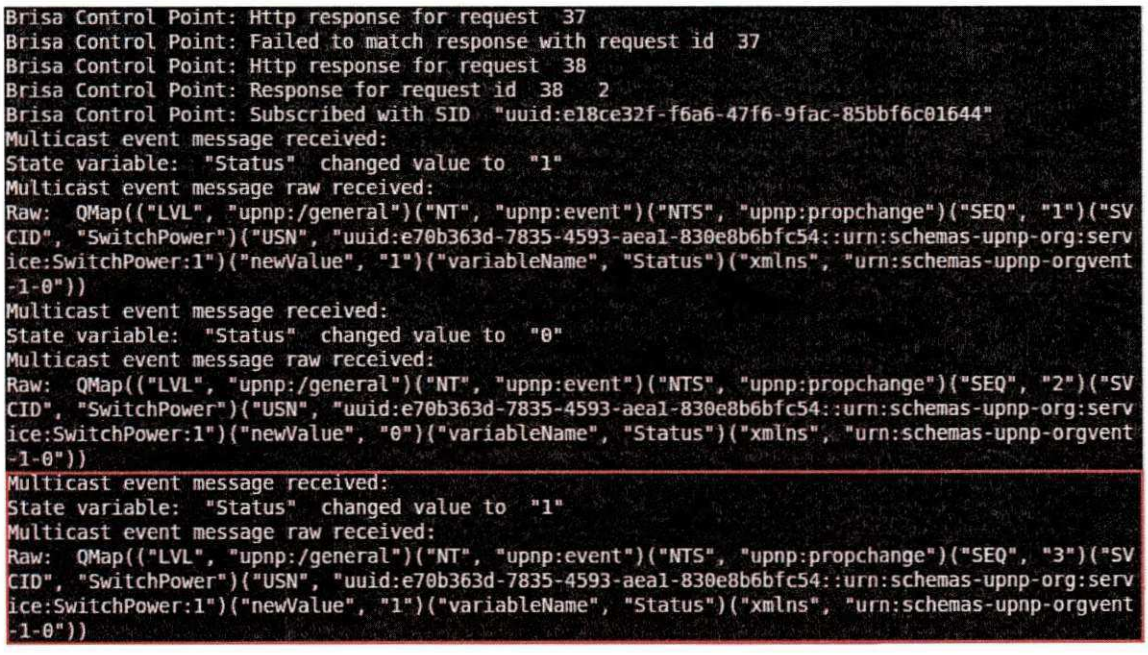


**Figura 2 - *Control point* informando a recepção de um evento *multicast*.**

### *Busca e correção de* memory leaks

Foi detectado um alto consumo de memória pelo *control point* do Qt-BRisa. Em poucas horas de execução, a aplicação já estava consumindo mais de 800MB de memória. Com as modificações realizadas, durante mais de um dia de execução, a aplicação consome pouco mais de 100 MB de memória.

A primeira atividade feita para tentar corrigir esse problema foi a procura por objetos criados que não eram destruídos posteriormente. Foram encontrados alguns, porém a destruição dos mesmos não reduziu o consumo de memória de forma satisfatória.

Logo após, foi feito o refatoramento do *parser* do XML dos dispositivos e serviços (atividade mencionada anteriormente). O resultado dessas mudanças foi notável, com uma grande redução do consumo de memória, mas ainda não totalmente satisfatório.

Outra tentativa de reduzir os *memory leaks* do *control point* foi destruir objetos armazenados em estrutura de dados (QVector, QMap, etc.) utilizando a função qDeleteAll(), provida por Qt, que destrói todos os elementos de um contêiner. Antes, a destruição desses objetos era feita através da iteração sobre os elementos desses contêineres. Não houve mudança notável no consumo de memória.

Foi feita também a mudança dos destrutores das classes do projeto para *virtual*. Essa mudança implica diretamente no uso de herança. Quando um objeto especializado é destruído, apenas o destrutor da classe filha é utilizado, podendo deixar alguns objetos ainda vivos na aplicação. Com o destrutor da classe mãe virtual, na destruição do objeto especializado, ambos os destrutores (da classe mãe e filha) são utilizados, evitando *memory leaks*. Poucas classes possuíam destrutores não virtuais, acarretando em pouca mudança no consumo de memória.

Foi utilizada a ferramenta Valgrind para encontrar locais na aplicação que poderiam gerar vazamento de memória. Porém, o *log* gerado pela ferramenta possuía mais de vinte mil linhas, onde a maioria reportava possíveis falhas nas bibliotecas C do Linux ou na biblioteca Qxt (LibQxt,2010)

(responsável pelo Web Server do Qt-BRisa). Por isso, nada muito útil pode ser colhido do Valgrind.

Foi desenvolvido um novo Web Server para o Qt-BRisa. O responsável por desenvolver o mesmo foi um aluno da Universidade Federal de Alagoas (UFAL), Vinícius dos Santos Oliveira, com o intuito de ficar livre de dependências externas além do Qt. Com o avanço do novo servidor web, novos testes de memória foram realizados. Os resultados do teste, porém, não foram muito satisfatórios em relação ao consumo de memória. Mas o desenvolvimento do Web Server ainda não foi encerrado e há expectativas de melhoria.

### *Funcionamento do Qt-BRisa com* threads

Foi detectado por Leandro que o *control point* não funcionava corretamente quando o mesmo era criado em uma *thread*. Além de dois warnings durante a execução, o mesmo não encontrava nenhum dispositivo na rede.

Foi pesquisada uma solução para o problema e foi descoberto que Qt não permite que um objeto seja inicializado em uma *thread* diferente da que foi criado. No caso do Qt-BRisa, esse problema ocorria com os *sockets* UDP (QUdpSocket), que iniciam uma *thread* para si.

Os *warnings* foram removidos, porém, os *devices* ainda não são encontrados pelo *control point*. A mensagem de *m-search* é entregue aos dispositivos. O que falta acontecer é a resposta dos mesmos chegarem até o *control point*. Esta atividade não foi concluída no período de estágio.

A Tabela 1 mostra o tempo gasto em cada atividade.

| Atividade | Descrição | Tempo Utilizado |
|---|---|---|
| A1 | Estudo da especificação UPnP e do Qt-BRisa | 40 horas |
| A2 | Criação de dispositivos | 70 horas |
| A3 | Suporte a eventos *multicast* | 30 horas |
| A4 | Refatoramento de código | 30 horas |
| A5 | Busca e correção de *memory leaks* | 100 horas |
| A6 | Funcionamento do Qt-BRisa com *threads* | 15 horas |
| A7 | Reuniões com orientador/supervisor e produção do relatório | 15 horas |
| TOTAL | - | 300 horas |

**Tabela 1 - Tempo médio gasto nas atividades.**

# SEÇÃO V

# CONSIDERAÇÕES FINAIS

# 5. CONSIDERAÇÕES FINAIS

Com as atividades realizadas, o estagiário obteve um acréscimo em seu conhecimento pela experiência obtida com o BRisa e com UPnP, que vem crescendo e sendo utilizado por várias empresas por todo o mundo, como Samsung e Nokia. O estagiário também adquiriu mais experiência como desenvolvedor, principalmente na linguagem C++ e em Qt, que vem sendo cada vez mais utilizado.

A realização das atividades atribuídas ao estagiário contribuiu para a manutenção e melhoria do Qt-BRisa. Depois do trabalho, o projeto agora está mais estável e atualizado, tendo a preferência de muitos usuários. O projeto de automação residencial da Spry Softwares também foi beneficiado com a criação dos dispositivos.

Apesar de os supervisores Thiago e Leandro nem sempre estarem presentes, a comunicação entre supervisor e estagiário ocorria de sem problemas via email e mensagens instantâneas (*gtalk*). Talvez pelo motivo de ambos os lados já terem experiência com trabalho a distância.

Houve alguns problemas durante o estágio, como algumas mudanças na equipe, que só ficou estável em meados de outubro. Porém, tais mudanças não afetaram as atividades do estagiário, pois essas eram geralmente independentes de outras atividades.

# REFERÊNCIAS BIBLIOGRÁFICAS

# REFERÊNCIAS BIBLIOGRÁFICAS


BRisa. Disponível em <http://brisa.garage.maemo.org/>. Último acesso em: 03 de dezembro de 2010.

GUPnP. Disponível em < http://gupnp.org/>. Último acesso em: 03 de dezembro de 2010.

LibQxt. Disponível em <http://dev.libqxt.org/libqxt/wiki/Home>. Último acesso em: 03 de dezembro de 2010.

NOKIA. Qt – A Cross-platform application and UI framework. Disponível em: <http://qt.nokia.com/>. Último acesso em: 03 de dezembro de 2010.

NOKIA. Qt Creator. Disponível em <http://qt.nokia.com/products/developer-tools/>. Último acesso em: 03 de dezembro de 2010.

UPnP Forum. Disponível em <http://upnp.org/>. Último acesso em: 12 de dezembro de 2010.

UPnP Forum. UPnP™ Device Architecture. Disponível em <http://www.upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.1.pdf>. Último acesso em: 03 de dezembro de 2010.

Valgrind. Disponível em <http://valgrind.org/>. Último acesso em: 03 de dezembro de 2010.

# APÊNDICES

# APÊNDICE A – PLANO DE ESTÁGIO


**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE**
**CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA**
**DEPARTAMENTO DE SISTEMAS E COMPUTAÇÃO**

## Plano de Estágio Integrado

**Danilo Araújo de Freitas**
dsurviver@gmail.com
Estagiário – UFCG

Agosto 2010

Informações Pessoais

**Nome:** Danilo Araújo de Freitas
**Matrícula:** 20811009
**Endereço residencial:** Rua Júlio Ferreira Tavares, 550 - Catolé
CEP: 58411-060, Campina Grande – PB.
**E-mail:** dsurviver@gmail.com
**Telefones:** (83) 88480131
(83) 99163152

## Ambiente de Estágio

O trabalho de estágio será realizado no Laboratório de Sistemas Embarcados e Computação Pervasiva - Embedded[0], no Laboratório 3 (Sala 04). O mesmo está localizado dentro das instalações da Universidade Federal de Campina Grande (UFCG).

O projeto será realizado em conjunto com mais oito pessoas, onde cinco são desenvolvedores, um gerente administrativo, um gerente de projeto e um gerente de desenvolvimento.

## Supervisão

*Supervisor Acadêmico*

**Nome:** Angelo Perkusich
**Endereço:** Universidade Federal de Campina Grande, Departamento de Engenharia Elétrica, Laboratório de Sistemas Embarcados e Computação Pervasiva. Bodocongó. 58109970 - Campina Grande, PB - Brasil - Caixa-Postal: 10105
**Email:** perkusic@embedded.ufcg.edu.br

*Supervisor Técnico*

**Nome:** Thiago Bruno Melo de Sales
**Endereço:** Universidade Federal de Campina Grande, Departamento de Engenharia Elétrica, Laboratório de Sistemas Embarcados e Computação Pervasiva. Bodocongó. 58109970 - Campina Grande, PB - Brasil - Caixa-Postal: 10105
**Email:** thiagobrunoms@gmail.com

## Resumo do problema

A automação residencial já é possível hoje em dia. Várias soluções para este problema são possíveis, porém, muitas são inviáveis por vários fatores, como custo e tecnologia.

O protocolo de rede UPnP[1] pode ser uma solução simples e barata. Através dele, é possível encontrar e controlar equipamentos que estejam conectados à rede. O projeto BRisa[2] é um *framework* que implementa tal protocolo, trazendo várias facilidades e vantagens para o desenvolvedor.

Combinando o BRisa e o uso de alguns equipamentos, torna-se possível e simples a automatização residencial de uma forma simples e barata. Dessa forma, será possívem controlar eletrodomésticos utilizando o computador, celular, PDA ou qualquer outro equipamento que possa se conectar à rede.

Para esse projeto, será utilizada a versão do BRisa em Qt[3].

## Objetivos

Primeiro, é preciso introduzir alguns conceitos utilizados no protocolo e no projeto BRisa:

- **Service:** É a menor unidade de controle. Possui ações e modelam o estado de um *device* com o uso de *state variables*.
- **Action:** Comandos de um *service*.
- **State Variables:** Variáveis que definem o estado de um *service*. Podem enviar eventos quando seus valores são alterados.
- **Device:** Dispositivo em si. Pode conter várias *services* que modelarão seu estado.
- **Control Point:** Responsável por buscar e controlar *devices*. Retém a descrição e *services* dos *devices* encontrados. Também envia ações aos *services* e recebe eventos dos mesmos.

Para uma automação residencial considerável, é preciso então implementar vários *devices*, que representaram os equipamentos e eletrodomésticos existentes em uma casa (computador, videogame, TV, geladeira, ar condicionado, etc.). Em paralelo, será desenvolvido um *control point*, que será responsável por encontrar os *devices* criados. O *control point* deverá ter uma interface gráfica intuitiva, de fácil uso, já que o perfil do usuário final poderá ser qualquer pessoa que queira automatizar sua residência utilizando nossa soulção. Através do mesmo, será possível encontrar e controlar todos os *devices* criados e que estejam conectados na rede local do usuário.

## Resultados Esperados

No final do estágio, espera-se ter alguns equipamentos total ou parcialmente automatizados. Para isso, teremos um *control point* totalmente funcional e alguns *devices* implementados.

Para uso futuro, torna-se necessário também a implementação de vários outros *devices*, que serão utilizados futuramente pela empresa em outros equipamentos.

## Metodologia

Serão disponibilizadas 4 horas por dia para o trabalho, 5 dias por semana, totalizando em 20 horas semanais. A cada semana, uma reunião de acompanhamento poderá ser realizada para acompanhamento, informando o status das atividades, modificações e definição de novas tarefas.

Atividades Planejadas

- A1: Estudo da especificação UPnP
- A2: Entendimento do código do BRisa Qt
- A3: Corrigir e adicionar documentação do BRisa Qt
- A4: Refatorar código do *parser* do xml das *services* e *devices*
- A5: Correção de bugs no BRisa Qt
- A6: Definição de alguns *devices* de eletrodomésticos
- A7: Implementação dos *devices*
- A8: Realização de testes utilizando o BRisa Qt e *devices* implementados
- A9: Correção dos devices implementados

Cronograma

| | S1 | S2 | S3 | S4 | S5 | S6 | S7 | S8 | S9 | S10 | S11 | S12 | S13 | S14 | S15 | S16 | S17 | S18 | S19 | S20 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A1 | X | X | | | | | | | | | | | | | | | | | | |
| A2 | | X | X | X | X | | | | | | | | | | | | | | | |
| A3 | | X | X | | | | | | | | | | | | | | | | | |
| A4 | | | | | | | X | | | | | | | | | | | | | |
| A5 | | | | | | X | X | X | X | X | | | | | | | | | | |
| A6 | | | | | | | | | | | X | | | | | | | | | |
| A7 | | | | | | | | | | | | X | X | X | X | X | | | | |
| A8 | | | | | | | | | | | | | | | | | X | X | X | |
| A9 | | | | | | | | | | | | | | | | | | | X | X |

# Bibliografia

[0] Laboratório de Sistemas Embarcados e Computação Pervasiva - Embedded: http://www.embedded.ufcg.edu.br/

[1] UPnP Forum: http://upnp.org/

[2] Projeto BRisa: http://brisa.garage.maemo.org/

[3] Qt: http://qt.nokia.com/

Aprovação

Declaro para os devidos fins que aprovo o planejamento das atividades descritas neste documento como plano de estágio do aluno Danilo Araújo de Freitas, matrícula 20811009.

**Angelo Perkusich**
Supervisor Acadêmico

**Thiago Bruno Melo de Sales**
Supervisor Técnico

**Joseana Macêdo Fechine**
Coordenador da Disciplina Estágio Integrado