



UFCG - Universidade Federal de Campina Grande
DSC - Departamento de Sistemas e Computação
CCC - Curso de Ciência da Computação
Disciplina de Estágio Integrado

Relatório de Estágio Integrado
Laboratório de Sistemas Embarcados e Computação Pervasiva.
Embedded

Hyggo Oliveira de Almeida
Orientador Acadêmico

Mário Hozano Lucas de Souza
Orientador Técnico

André Cavalcante Hora
Estagiário

Campina Grande, setembro de 2007.



Biblioteca Setorial do CDSA. Maio de 2021.

Sumé - PB

SUMÁRIO

1.	Introdução.....	4
2.	Ambiente de Estágio	6
2.1.	Descrição do Ambiente de Estágio	6
2.2.	Equipe.....	6
2.3.	Aspectos Positivos.....	7
2.4.	Aspectos Negativos	7
3.	Descrição do Problema.....	8
4.	Metodologia	9
5.	Proposta de Solução	11
5.1.	Principais Tecnologias Utilizadas	11
5.1.1.	PyMaemo	12
5.1.2.	GStreamer.....	12
5.2.	Revisão/Validação da Documentação da Python-Hildon	13
5.3.	Módulos Implementados	13
5.3.1.	Câmera	14
5.3.2.	E-mail.....	15
6.	Considerações Finais.....	16
7.	Referências	17
8.	Declaração de Aprovação de Estágio.....	18

LISTA DE ABREVIATURAS

API	Application Programming Interface
SVN	Subversion
Wi-Fi	Wireless Fidelity
GPS	Global Positioning System
PyMaemo	Python for Maemo
XP	Extreme Programming
Embedded	Lab. de Sistemas Embarcados e Computação Pervasiva
DSC	Departamento de Sistemas e Computação
UFCG	Universidade Federal de Campina Grande
CEEI	Centro de Engenharia Elétrica e Informática
INdT	Instituto Nokia de Tecnologia

1. Introdução

Nos últimos anos, tem-se observado uma tendência da indústria no desenvolvimento de tecnologias que tragam mobilidade e agilidade na realização de atividades dos usuários. Isto proporcionou um grande crescimento dos sistemas embarcados e, conseqüentemente, aplicações para dispositivos móveis estão se tornando cada vez mais populares. Recentemente tais dispositivos aumentaram seu poder computacional, tanto de processamento quanto de memória, concedendo ao usuário melhor desempenho.

Junto a este progresso estão os softwares que auxiliam a parte física da máquina tornando assim mais simples sua utilização. Neste tempo percebeu-se a criação de monopólios em todas as áreas da computação. A falta de opções levou a população ao software proprietário desde computadores pessoais até dispositivos portáteis. Mas nos últimos anos viu-se despontar o software livre como alternativa estável aos códigos proprietários. Nos dispositivos móveis o cenário não foi muito diferente, e mais uma vez, o software livre trouxe uma opção cabível nesta área. Por exemplo, o Maemo [1], que é uma plataforma de desenvolvimento utilizado em dispositivos móveis como os *Internet Tablets* Nokia 770 [2] e o Nokia N800 [3] que podem ser vistos na Figura 1.



Figura 1. Nokia 770 e Nokia N800 respectivamente.

O desenvolvimento de aplicações para tais dispositivos móveis não ocorre de forma direta quando comparado ao desenvolvimento para computadores pessoais. Aplicações para dispositivos móveis exigem um maior

conhecimento em tecnologias, o que normalmente torna as implementações mais difíceis.

Buscando solucionar esses problemas e outros relacionados ao desenvolvimento para a plataforma Maemo, a *API Easy* [4] foi desenvolvida para facilitar a implementação de aplicações para estes sistemas sendo composta pelos módulos de câmera, áudio, rádio, e-mail, *bluetooth* e GPS. Neste trabalho serão focados os módulos de câmera e e-mail.

Este relatório é o resultado de parte do trabalho de estágio no desenvolvimento da *API Easy*. Está dividido na seguinte forma: na Seção 2, descreve-se o ambiente de estágio; na Seção 3, apresenta-se a descrição do problema; na Seção 4, descreve-se a metodologia utilizada no desenvolvimento da *API*; na Seção 5, apresenta-se a solução para o problema descrito na Seção 4; na Seção 6, detalham-se as atividades desenvolvidas; e finalmente na Seção 7, são apresentadas algumas considerações finais.

2. Ambiente de Estágio

2.1. Descrição do Ambiente de Estágio

O estágio foi realizado no Laboratório de Sistemas Embarcados e Computação Pervasiva - Embedded, vinculado à Universidade Federal de Campina Grande - UFCG, situada à Avenida Aprígio Veloso, 882 no Bairro de Bodocongó.

O Embedded está situado no campus 2 da UFCG. O laboratório faz parte do Centro de Engenharia Elétrica e Informática - CEEI e possui laboratórios de desenvolvimento e pesquisa, salas de aula, salas de professores e sala de reuniões.

A rede local do laboratório está baseada em Ethernet de um gigabite conectada à rede do campus da UFCG e à Internet. Além disso, todo o laboratório está equipado com acesso à Internet via rede sem fio (Wi-Fi). O laboratório possui uma biblioteca que está em constante atualização, contendo livros recentes adquiridos através de recursos de projetos.

A execução do projeto ocorreu no Laboratório 2 de pesquisa e desenvolvimento do Embedded utilizando 3 computadores e 3 *Internet Tablets* Nokia N800.

2.2. Equipe

O Embedded possui alunos de graduação, mestrado e doutorado, nos cursos de Ciência da Computação e Engenharia Elétrica. Esses alunos são responsáveis pelo desenvolvimento de projetos específicos e possuem atividades de pesquisa e desenvolvimento para sistemas embarcados e computação pervasiva que estão diretamente ligados ao Instituto Nokia de Tecnologia - INdT.

O desenvolvimento da *API* foi gerenciado pelo mestrando Mário Hozano Lucas de Souza. O autor deste documento, Emanuel Dantas (Computação) e Miguel Falcão (Engenharia Elétrica) foram os responsáveis pelo desenvolvimento da *API*. Alguns colaboradores também foram de fundamental

importância como: Osvaldo Santana, Marcio Macedo, Lauro Moura e Luciano Wolf, todos do Instituto Nokia de Tecnologia.

2.3. Aspectos Positivos

Parte do conhecimento adquirido durante o curso de Bacharelado em Ciência da Computação pôde ser posto em prática durante o desenvolvimento das atividades no laboratório. A integração com alunos do curso de Engenharia Elétrica, mestrandos e com professores com um vasto conhecimento em suas áreas nos propiciou um aprendizado extracurricular importante para a nossa vida acadêmica.

A flexibilidade de horários de trabalho e o fácil acesso ao laboratório foram importantes, quando necessário para cumprir as tarefas determinadas. Outro grande fator de importância foi a disponibilidade total dos computadores e dos *Internet Tablets N800* aos desenvolvedores.

2.4. Aspectos Negativos

No ambiente de trabalho, não foram identificados aspectos negativos significativos, que merecessem ser relatados aqui. Mas a falta destes aspectos faz com que este ambiente não seja tão semelhante à maioria dos ambientes de trabalho comercial. Sendo assim, o estagiário não adquiriu uma noção bem realista de como poderia ser sua vida profissional no mercado de trabalho.

3. Descrição do Problema

O desenvolvimento de aplicações para a plataforma Maemo não ocorre de forma simples. Embora Maemo seja uma plataforma aberta, acessar seus recursos exige um bom nível de conhecimento nesta plataforma. Para ter acesso aos seus recursos como câmera, áudio, rádio, e-mail, *bluetooth* e GPS é necessário um bom nível de conhecimento em várias bibliotecas e *frameworks*. Por exemplo, para reproduzir ou gravar um arquivo de áudio e vídeo ou acessar a câmera do dispositivo o desenvolvedor precisa ter conhecimento no *framework* multimídia Gstreamer [5] e GTK [6], sendo forçado a aprender conceitos de *pipelines* e *codecs*, dificultando a implementação da aplicação.

4. Metodologia

O processo de desenvolvimento de software utilizado foi o XP [7] que segue uma metodologia de desenvolvimento *lightweight*. XP enfatiza o uso extremo de práticas que se mostram funcionais, como revisão de código, teste, simplicidade e ciclo curto e iterativo [8]. XP visa alcançar duas metas desejadas pelas indústrias de tecnologia da informação: desenvolvimento rápido e consistente com as reais necessidades do cliente e uma fácil manutenibilidade permitindo que o software seja modificado à medida que as necessidades se alterem.

A grande preocupação inicial foi em relação aos conhecimentos da equipe em relação às tecnologias utilizadas no desenvolvimento do *Easy*. Foi necessária uma fase de estudos para a aprendizagem destas tecnologias e nivelamento da equipe de desenvolvimento.

Existiram três papéis desempenhados pelos membros da equipe. O gerente de projeto foi o mestrando Mario Hozano. O papel do cliente foi desempenhado pelo INdT e o papel do desenvolvedor foi desempenhado pelo estagiário, Emanuel Dantas e Miguel Falcão. As *User Stories* foram definidas pelo cliente.

As reuniões entre desenvolvedores e gerente ocorriam semanalmente. Entretanto, o contato entre cliente, gerente e desenvolvedores ocorria diariamente através da *web*, visto que o cliente se encontrava em outra cidade.

O projeto do *Easy* utiliza o ambiente GFORGE [9] como forma de interação entre a equipe de desenvolvimento e cliente. Tal ambiente é uma ferramenta corporativa para desenvolvimento de software de código aberto. Baseia-se no *Sourceforge* [10], uma ferramenta semelhante amplamente utilizada pela comunidade de software livre, que abriga hoje mais de 10.000 projetos e 1 milhão de usuários. O projeto *Easy* utiliza de forma intensiva vários recursos disponíveis no ambiente, como os listados abaixo:

- **Comunicação**

Fóruns - fóruns de discussão gerenciáveis

Listas - listas de e-mail para a comunidade

Enquetes - sistema de votação sobre assuntos diversos

Notícias - ferramenta para a divulgação de novidades

Estatísticas - apontam estatística que indicam números de arquivos inseridos por cada membro da equipe

- **Suporte**

Tracker - gerenciador do projeto

Tarefas - sistema para a criação e alocação de tarefas (*User Stories*)

Docs - mecanismo para armazenar documentação de software

- **Controle de versão**

SCM - sistema de controle de versão de código-fonte (SVN)

Arquivos - distribuição de arquivos e pacotes de software

Tais recursos são de fundamental importância para o desenvolvimento do projeto, pois o cliente obtém *feedbacks* diariamente de tudo que está sendo implementado e interage de forma mais efetiva com os membros da equipe.

5. Proposta de Solução

Como resultado do estágio foi revisada e validada a documentação de uma biblioteca para a criação de interfaces gráficas na plataforma Maemo, denominada Python-Hildon [11]. Também foi implementada a API *Easy* que encapsula a complexidade de acesso a componentes multimídia e outros serviços desta mesma plataforma. Tal implementação foi feita utilizando a linguagem de programação Python [12] e uma camada de abstração para acessar componentes de mais baixo nível oferecidos pelo dispositivo como os recursos multimídia, rádio, bluetooth e GPS. *Easy* é a solução para grande parte dos problemas descritos na seção 4. Neste trabalho serão descritos os módulos de câmera e e-mail, como proposto para o estágio. Sua arquitetura pode ser vista na Figura 2.

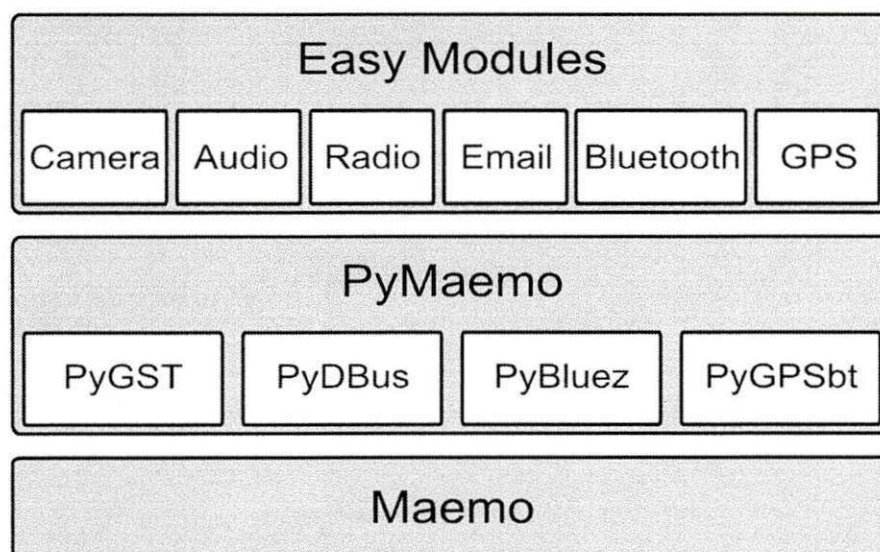


Figura 2. Arquitetura do *Easy*.

5.1. Principais Tecnologias Utilizadas

Dentre as principais tecnologias utilizadas na implementação do *Easy* destacam-se *PyMaemo* [13] e o *framework* multimídia GStreamer.

5.1.1. PyMaemo

PyMaemo (Python for Maemo) é um trabalho para adequar a linguagem Python à plataforma Maemo permitindo que os dispositivos que utilizam esta plataforma possam utilizar aplicações desenvolvidas em Python, não se restringindo à linguagem C. PyMaemo foi projetado para ter menor tamanho e algumas otimizações visto que sua utilização se dá em dispositivos móveis. O grande número de componentes PyMaemo são a base para o desenvolvimento de vários módulos do *Easy*. São exemplos destes componentes: PyGST [14], PyBlueZ [15], PyDBus[16], PyGPSbt [17] que são inslustrados na Figura 2.

5.1.2. GStreamer

Gstreamer é um *framework* para criação de aplicações multimídia. Depois da versão 0.10 muitas aplicações multimídias começaram a usá-lo, devido à correção de bugs em seu código. A versão mais atual é a 0.10.14 que é liberada sob a licença LGPL.

Aplicações em multimídia necessitavam de uma padronização visto que implementavam as mesmas funcionalidades de maneiras diferentes onde os esforços eram duplicados e não havia cooperação, ou seja, as aplicações só poderiam ser implementadas quando toda infra-estrutura estivesse funcionando. GStreamer provê uma padronização através de uma camada de abstração sobre componentes de manipulação para qualquer formato para o qual haja um *plugin* instalado. As aplicações não precisam mais se preocupar em implementar decodificadores, *mixers*, sincronia nem codecs o que significa maior produtividade [18]. Na Figura 3 pode-se observar onde o *framework* é posicionado.

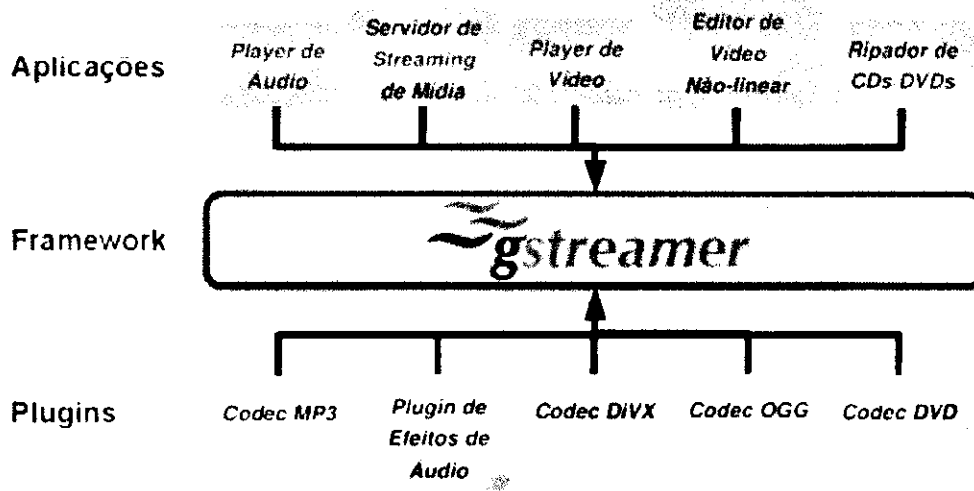


Figura 3. Framework GStreamer.

5.2. Revisão/Validação da Documentação da Python-Hildon

Python-Hildon é uma biblioteca que oferece aos desenvolvedores *bindings* para utilizar componentes de interface gráfica em Python. Tal biblioteca é composta de 23 módulos que facilita a criação de interfaces gráficas em Python para Maemo. Alguns exemplos dos componentes oferecidos pela biblioteca são: `hildon.CalendarPopup`, `hildon.ColorSelector`, `hildon.FontSelectionDialog`, `hildon.GetPasswordDialog`, `hildon.Note`, `hildon.NumberEditor`, `hildon.Seekbar`, `hildon.TimeEditor`, `hildon.TimePicker` e `hildon.WeekdayPicker` que oferecem funções básicas para manipulação de interface.

A documentação foi revisada para que a mesma refletisse o que a biblioteca provia, pois ambos se encontravam em versões diferentes sendo necessária uma sincronização. Tal documentação também era composta por exemplos que foram revisados e adequados à nova versão da biblioteca.

5.3. Módulos Implementados

Nas seções seguintes são descritos os dois módulos implementados para este trabalho que são: o módulo para manipular a câmera do dispositivo e o módulo para manipulação de e-mail.

5.3.1. Câmera

O módulo câmera é composto por funções para manipular a câmera do dispositivo, incluindo gravar e reproduzir arquivos de vídeo em formato AVI [19], assim como tirar fotos em formato JPG [20]. Outras funções também estão incorporadas ao módulo, como: mudança de propriedades do vídeo (brilho, contraste e saturação), seek, duração do vídeo e posição corrente.

Este módulo utiliza o *framework* GStreamer para efetuar as operações multimídia e é composto por dois *pipelines* (fluxos de *stream*): um responsável por iniciar a câmera, gravar e capturar imagens e outro para reproduzir arquivos de vídeo. Tais funções também podem ser realizadas de forma síncrona ou assíncrona. Abaixo segue um exemplo de uso do módulo para iniciar a câmera, gravar um vídeo, exibi-lo e capturar uma foto.

```
1 from easy import camera
2 camera.start()
3 camera.record("myvideo.avi", 30)
4 camera.play("myvideo.avi")
5 camera.click("myphoto.jpg")
```

Na primeira linha importamos o módulo câmera da *API* e assim temos acesso a todas as funções disponibilizadas por este módulo. Na linha 2 a câmera do dispositivo é iniciada. Na linha 3 gravamos um vídeo de trinta segundos e em seguida na linha 4 exibimos esse vídeo. Na última linha ocorre a captura de uma imagem. Pode-se observar a simplicidade de se fazer quatro operações em apenas cinco linhas de código.

Para efetuar as mesmas operações utilizando o GStreamer seria necessário conhecimento em *plugins* o que dificultaria a implementação de aplicações multimídia. Na Figura 4 observa-se como os *plugins* GStreamer estão organizados dentro do *pipeline* responsável por iniciar câmera, gravar e capturar imagens.

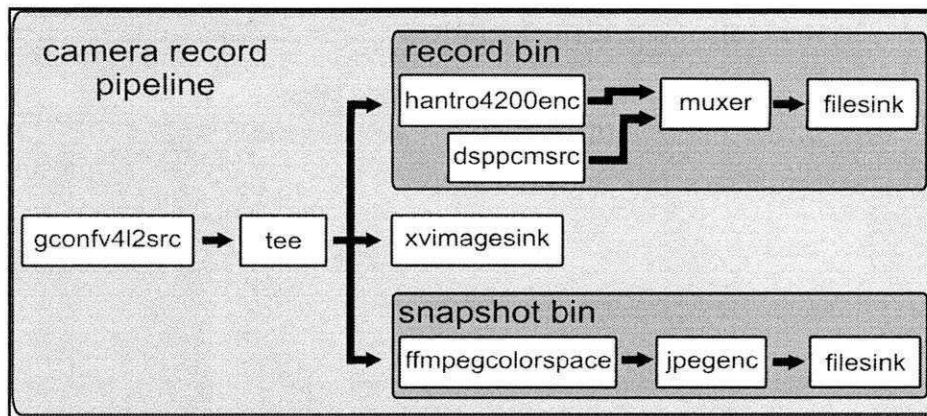


Figura 4. Pipeline responsável por iniciar/gravar/tirar fotos.

5.3.2. E-mail

O módulo e-mail é composto de funções para enviar e-mail. Possui fácil manipulação, exigindo apenas o servidor SMTP, o *login* e a senha para se conectar. Após a autenticação o desenvolvedor necessita informar o destinatário e a existência ou não de anexo na mensagem. Este módulo encapsula a biblioteca SMTPLIB [21] disponível para Python.

Um exemplo da utilização dos módulos descritos neste trabalho podem ser visto abaixo:

```

1 from easy import camera, e_mail
2 camera.start()
3 camera.click("myphoto.jpg")
4 e_mail.connect("login", "password", "smtpserver")
5 e_mail.attach("myphoto.jpg")
6 e_mail.send_mail("destination", "Subject", "Message")

```

Na primeira linha importamos o módulo câmera e e_mail. As linhas 2 e 3 são para a captura de uma imagem, como descritas no exemplo anterior. Na linha 4 ocorre a conexão com o servidor de e-mail. Na linha 5 a foto é anexada e na última linha o e-mail é enviado para o destinatário.

6. Considerações Finais

Vimos que a *API Easy* é uma solução para grande parte dos problemas descritos. Sua fácil manipulação e simplicidade de uso é o diferencial desta *API* que permite a utilização dos recursos disponíveis na plataforma Maemo com poucas linhas de código. Outra característica importante é que seu código está disponível para comunidade interessada na área e conseqüentemente está sempre em evolução. A revisão da biblioteca foi importante para o aprimoramento de se manipular com interfaces gráficas no Maemo.

O estágio foi de grande importância para o aprendizado do estagiário e sua experiência no que diz respeito ao desenvolvimento de software livre, realizando atividades de plano de projeto, escolhas de tecnologias e elaboração de projeto arquitetural. Portanto, houve um aprimoramento na habilidade do estagiário em realizar estas tarefas que são de suma importância à sua formação acadêmica e ingresso no mercado de trabalho.

Além destas contribuições ao estagiário, pode-se destacar também o aprendizado nas tecnologias descritas nas Seções 5.1.1 e 5.1.2 além do aprimoramento dos seus conhecimentos na linguagem de programação Python.

O cliente se mostrou satisfeito com o trabalho desenvolvido, pois a *API* atende às suas necessidades e correspondeu às suas expectativas do início de projeto. Através das ferramentas de colaboração e gerência o cliente acompanhou e interagiu durante todo o desenvolvimento.

7. Referências

- [1] <http://maemo.org> (Acessado em 17/07/2007)
- [2] <http://www.nokiausa.com/770> (Acessado em 17/09/2007)
- [3] <http://www.nseries.com/n800> (Acessado em 17/09/2007)
- [4] <https://garage.maemo.org/projects/easy> (Acessado em 18/09/2007)
- [5] <http://gstreamer.freedesktop.org> (Acessado em 18/09/2007)
- [6] <http://www.gtk.org> (Acessado em 18/09/2007)
- [7] <http://www.extremeprogramming.org> (Acessado em 18/09/2007)
- [8] Kent Beck. "Extreme Programming Explained: Embrace Change" - Addison - Wesley, 2ª edição, 2000.
- [9] <http://gforge.org> (Acessado em 18/09/2007)
- [10] <http://sourceforge.net> (Acessado em 18/09/2007)
- [11] http://maemo.org/development/documentation/apis/3-x/python-maemo-3.x/python_hildon_widgets_api.html (Acessado em 20/09/2007)
- [12] <http://www.python.org> (Acessado em 20/09/2007)
- [13] <http://pymaemo.garage.maemo.org> (Acessado em 20/09/2007)
- [14] <http://gstreamer.freedesktop.org/modules/gst-python.html> (Acessado em 20/09/2007)
- [15] <http://org.csail.mit.edu/pybluez> (Acessado em 22/09/2007)
- [16] <http://www.freedesktop.org/wiki/Software/dbus> (Acessado em 21/09/2007)
- [17] http://pymaemo.garage.maemo.org/documentation/python_gps_examples/index.html (Acessado em 20/09/2007)
- [18] http://www.cin.ufpe.br/~cinlug/wiki/index.php/Apresentando_o_GStreamer (Acessado em 22/09/2007)
- [19] <http://www.jpeg.org> (Acessado em 22/09/2007)
- [20] <http://www.jmcgowan.com/avi.html> (Acessado em 22/09/2007)
- [21] <http://docs.python.org/lib/module-smtplib.html> (Acessado em 22/09/2007)

8. Declaração de Aprovação de Estágio

Declaro, para os devidos fins, que André Cavalcante Hora, estudante do curso de Ciência da Computação da Universidade Federal de Campina Grande, matrícula 20411019, cumpriu corretamente suas atividades de desenvolvimento na API *Easy*, que foi alvo de seu estágio no Laboratório de Sistemas Embarcados e Computação Pervasiva, realizado no período de junho de 2007 a setembro de 2007.

Mário Hozano Lucas de Souza
Orientador Técnico