

UNIVERSIDADE FEDERAL DE VIÇOSA
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

NARAIEL PEREIRA FERRARI

Controle De Processos Através Do Smartphone

VIÇOSA

2011

NARAIEL PEREIRA FERRARI

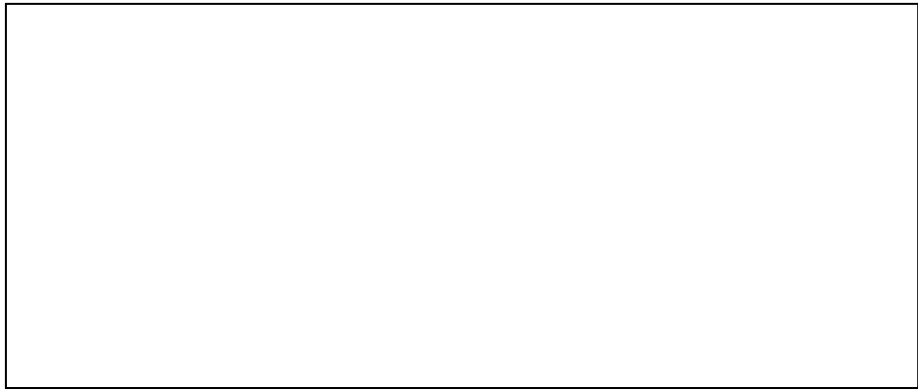
Controle De Processos Através Do Smartphone

Monografia apresentada ao Departamento de Engenharia Elétrica do Centro de Ciências Exatas e Tecnológicas da Universidade Federal de Viçosa, para a obtenção dos créditos da disciplina ELT 490 – Monografia e Seminário e cumprimento do requisito parcial para obtenção do grau de Bacharel em Engenharia Elétrica.

Orientador: Dr. André Gomes Torres –
DEL/UFV

VIÇOSA

2011



NARAIEL PEREIRA FERRARI

Controle De Processos Através Do Smartphone

Monografia apresentada ao Departamento de Engenharia Elétrica do Centro de Ciências Exatas e Tecnológicas da Universidade Federal de Viçosa, para a obtenção dos créditos da disciplina ELT 490 – Monografia e Seminário e cumprimento do requisito parcial para obtenção do grau de Bacharel em Engenharia Elétrica.

Aprovada em 02 de Dezembro de 2011.

COMISSÃO EXAMINADORA

Prof. Dr. André Gomes Torres
Universidade Federal de Viçosa

Prof. Dra. Kétia Soares Moreira
Universidade Federal de Viçosa

Prof. Ms. Heverton Augusto Pereira
Universidade Federal de Viçosa

“Conseguir é melhor do que ter”.

(Bill Gates.)

A minha amada mãe, Denise!

Agradecimentos

Gostaria de agradecer ao meu orientador e amigo, professor André Gomes Torres, que sempre me ajudou a solucionar os problemas quando solicitado.

Agradeço ao amigo, Rafael Cruz que me ajudou com todo o projeto em si, principalmente, com a ideia geral do trabalho realizado.

Agradeço aos meus pais, que sempre estiveram ao meu lado, me apoiando e dando força; agradeço também aos meus irmãos, Caio Ferrari, Lívia Ferrari e Flávia Ferrari, sem eles não seria quem sou; enfim agradeço a minha família por cada momento de minha vida.

Agradeço aos meus amigos da Universidade, Rodrigo Moreira, Fabiano Marcarini, Afrânio Vilela, Gabriel Akira, Bruno Marin, Diego César e André Sasaki que estiveram ao meu lado durante toda a graduação.

A equipe Richards do Porto Velho Shopping que mesmo distantes sempre contribuíram para o meu sucesso. Assim como, meus antigo amigos, que há muito não os vejo.

Enfim, obrigado por vocês terem me encorajado. Se, hoje, estou formando, agradeço a vocês.

Resumo

A crescente busca pela comodidade impulsiona a espécie humana a desvendar a natureza e as suas leis, desenvolver mecanismos, criar métodos, equipamentos, tudo para trazer-lhe o maior conforto. O uso dos smartphones vem atendendo muito bem o mundo neste quesito. Centralizando várias funções em um único dispositivo. Neste trabalho será criado um aplicativo para o iPhone, que fará o controle ON/OFF de duas bombas, de uma planta industrial. A interface, ainda, contará com a aquisição do histórico das bombas assim como o seu estado em tempo real. Os resultados mostram como o aplicativo funcionou e como este pode facilitar a interação humana com as plantas industriais.

Palavras-Chave: Controle De Processos, Smartphone, IHM.

Sumário

1. Introdução.....	13
1.1. Smartphones	13
1.1.1. Introdução.....	13
1.1.2. Smartphones No Brasil E No Mundo.....	13
1.1.3. Apple	15
1.1.4. xCode	16
1.2. Desenvolvimento Web	17
1.2.1. Introdução.....	17
1.2.2. Páginas Estáticas e Páginas Dinâmicas.....	17
1.2.3. Personal Home Page (PHP)	18
1.2.4. WAMP Server	20
1.3. Orientação a Objetos	20
1.3.1. Introdução.....	20
1.3.2. Objetos	21
1.3.3. Encapsulamento.....	21
1.3.4. Classe	22
1.3.5. Herança	22
1.3.6. Polimorfismo	22
1.4. Controle De Processos Industriais	23
1.4.1. Introdução.....	23
1.4.2. Tecnologia OLE/COM	26
1.4.3. Tecnologia OPC	27
1.4.4. Matlab e OPC	28
2. Metodologia.....	29
2.1. Desenvolvimento Geral	29
2.2. Desenvolvimento do Aplicativo.....	30
2.3. Desenvolvimento Web	35
2.4. Desenvolvimento da comunicação com a planta.....	36

3.	<i>Resultados</i>	40
4.	<i>Conclusão</i>	47
5.	<i>Referências Bibliografia</i>	48

Lista de Figuras

FIGURA 1 - VENDA DE PCS E SMARTPHONES NO MUNDO.....	14
FIGURA 2 - <i>MARKET SHARE</i> DE SMARTPHONES NO 3º TRIMESTRE DE 2011.....	15
FIGURA 3 - PRODUTOS DA APPLE.....	16
FIGURA 4 - ESQUEMA DE FUNCIONAMENTO DO PHP.....	19
FIGURA 5 - ARQUITETURA BÁSICA DO SISTEMA SCADA.....	25
FIGURA 6 - ARQUITETURA OPC.....	28
FIGURA 7 - ESQUEMA GERAL.....	29
FIGURA 8 - WINDOWS E MAC OS X TRABALHANDO JUNTOS NO VMWARE WORKSTATION.....	30
FIGURA 9 - AMBIENTE MAC OS X E ITUNES.....	31
FIGURA 10 - MODELOS PARA A CRIAÇÃO DE APLICATIVOS.....	32
FIGURA 11 - ARQUIVOS <i>DO VIEW-BASED APPLICATION</i>	33
FIGURA 12- LÓGICA <i>LADDER</i> DA BOMBA 1.....	37
FIGURA 13 - <i>IOS SIMULATOR</i>	40
FIGURA 14 - <i>VIEW</i> PRINCIPAL (ESQUERDA) E SECUNDÁRIA (DIREITA).....	41
FIGURA 15 - RESULTADO AO LIGAR BOMBA 1.....	42
FIGURA 16 - RESULTADO AO LIGAR A BOMBA 2.....	43
FIGURA 17 - RESULTADO AO LIGAR A BOMBA 2 QUE JÁ ESTÁ LIGADA.....	44
FIGURA 18 - HISTÓRICO SALVO NO SERVIDOR.....	45
FIGURA 19 - HISTÓRICO NO SMARTPHONE.....	45

LISTA DE TABELAS

TABELA 1 - OBJETOS DA <i>VIEW</i> PRINCIPAL	34
TABELA 2 - OBJETOS DA <i>VIEW</i> SECUNDÁRIA	34
TABELA 3 - AÇÃO E URL	36
TABELA 4 - NOME E REPRESENTAÇÃO DOS ITENS DA LÓGICA <i>LADDER</i>	37
TABELA 5 – EXEMPLO DE CICLO DE FUNCIONAMENTO	38

1. Introdução

1.1. Smartphones

1.1.1. Introdução

Smartphone (“telefone inteligente”) nada mais é do que um celular, com muitas outras funcionalidades [1]. Executados por meio de programas no seu Sistema Operacional (OS), este aparelho conta com as principais tecnologias de comunicação e serviços que temos em um computador, como acesso a e-mails, mensagens instantâneas, internet, GPS e ainda possibilita a criação de aplicativos (programas, em geral, direcionados a este dispositivo).

Não existe um consenso na indústria sobre quais quesitos um celular deve preencher para ser considerado um smartphone [2]. Entretanto, os telefones inteligentes, possuem como características relevantes: um teclado completo, tela touch-screen, câmera integrada de alta resolução, canais de entrada tipo USB, players de vídeos e áudio. Além disso, possuem meios de se comunicar através de Bluetooth, infravermelho e internet banda larga, possibilitando a transferência de vídeos, fotos, músicas, planilhas e até mesmo sincronizar informações com outros programas no computador.

Neste mesmo quesito, entra uma questão bem complicada, o custo dos Smartphones. Quanto mais completo é o dispositivo e quanto maior as opções de serviço contratado pelas operadoras, maior será o custo. Sendo assim, os fabricantes investem em tecnologias com intuito de baixar seus preço e conquistarem o maior *Market-Share* possível. Dentre os fabricantes, destacam-se: Apple, Motorola, Nokia e Samsung.

1.1.2. Smartphones No Brasil E No Mundo

“*In three years time, desktops will be irrelevant*” – John Herlihy, chefe da Google Europa, afirma que em três anos, computadores de mesa serão irrelevantes. Estas previsões podem ser confirmadas através do histórico das vendas de PCs no mundo em relação com as vendas de smartphones, Figura 1 [3] [4] [5].

Apesar de haver um crescimento nas vendas de computadores, o crescimento dos smartphones já ultrapassam os desktops. A Gartner [4], empresa de consultoria, estima que em 2011 sejam vendidos 356,9 milhões de PCs, contra 404,1 milhões de Smartphones.

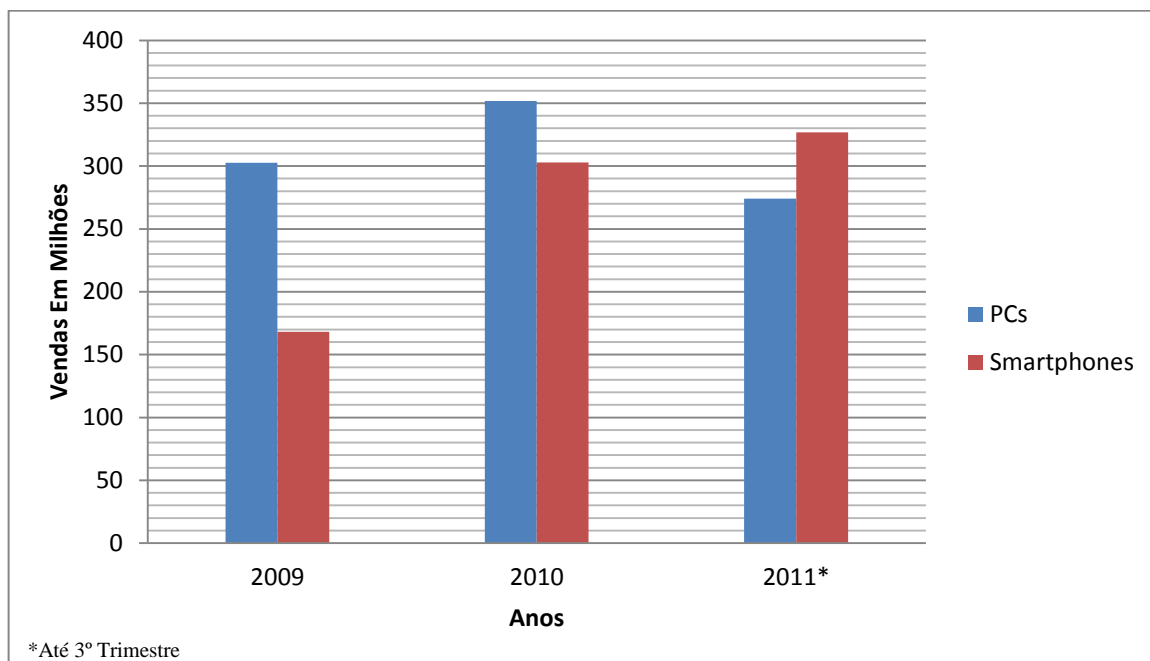


Figura 1 - Venda de PCs e smartphones no mundo

Devido a essa grande expansão neste segmento, os grandes fabricantes atendem cada vez mais a necessidade dos seus clientes, e toda vez que há o lançamento de um novo produto, os números do *market share* variam muito. No ano de 2010, a Nokia liderava o mercado com mais de 36 pontos percentuais, seguido do RIM (17,6%), da Apple (16,2%) e por fim da Samsung (6,9%) [5]. Com o lançamento do Samsung Galaxy e a nova geração dos iPhones (Apple), este panorama mudou e no terceiro trimestre do ano de 2011 o *market share* das empresas são mostradas na Figura 2.

Estima-se que para o ano de 2015, serão vendidos mais de 1 bilhão de aparelhos móveis inteligentes – IDC [3]. Este número mais do que dobra em relação a 2011. Todo esse crescimento se dará devido a preços mais baixos, design e melhores recursos. Boa parte da popularização de smartphone será nos países da América Latina e Ásia, onde a adoção desses produtos ainda é pequena comparada com outros mercados.

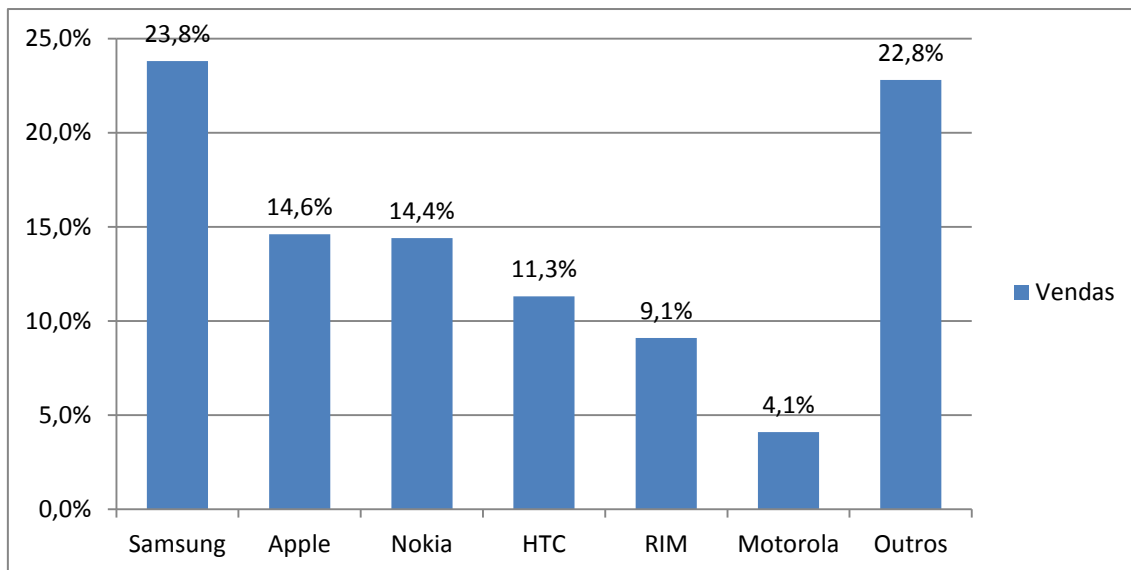


Figura 2 - *Market share* de smartphones no 3º trimestre de 2011

No Brasil, números divulgados pela W/McCann [6], mostram que, existem pouco mais de 19 milhões de celulares inteligentes, isso representa aproximadamente 9% do total de celulares usado no país. Apesar do pequeno número de aparelhos, nos últimos 6 meses houve um crescimento expressivo de 165%. A Nielsen [7], indica que 39% dos smartphones vendidos são Androids (sistema operacional desenvolvido, pela Google, que domina uma ampla gama de aparelhos, inclusive os smartphones, de diversos fabricantes) e a iOS (sistema operacional da Apple) representa 10% do mercado.

1.1.3. Apple

Fundada em 1976 por Steven Jobs e Stephen Gary Wozniak, a Apple, tinha como objetivo principal a criação de uma interface amigável para computadores de uso doméstico [8]. A Apple I era montada em um caixote e foi o primeiro computador a sair de fábrica com monitor, teclado e leitor de disquetes para armazenar programas. Este projeto foi um sucesso e então foi lançada a linha Apple II, que era baseada na navegação através de ícones, pastas, janelas e uso do mouse.

Com a saturação no mercado, a Apple demitiu vários funcionários e Wozniak saiu da empresa. Por motivos de má gestão e brigas internas, Jobs, é demitido em 1985. Com a empresa sem seus principais fundadores, a empresa perde o foco e a penetração no mercado e o seu sistema se torna obsoleto, trazendo prejuízo à empresa.

Em 1996, para ter o empresário novamente à frente da sua empresa original, a Apple compra, estrategicamente, a Next, que tinha como fundador o próprio Steven Jobs. Este saneou a linha de produtos e reajustou os sistemas operacionais da Apple. Tornou-se mais uma vez CEO (chefe executivo) da empresa e revolucionou o seus produtos.

Os produtos de hardware mais conhecidos da empresa incluem a linha de computadores Macintosh, iPod, iPhone e o iPad, Figura 3. Com sede em Cupertino, California, em 2010 a empresa operava 301 lojas de varejo em dez países e uma loja online onde seus produtos são vendidos [8]. Em maio de 2011, a Apple atinge o patamar de a empresa mais valiosa planeta em seu ramo, tendo ultrapassado, até mesmo, a Microsoft.



Figura 3 - Produtos da Apple

Steven Jobs veio a falecer no dia 05 de Outubro de 2011 vítima de câncer de pâncreas. Wozniak fundou a empresa Wheels Of Zeus, em 2001, uma companhia que produz soluções sem fio, onde trabalha atualmente.

1.1.4. xCode

Criado pela própria Apple, o xCode é um software livre que fornece todas as ferramentas necessárias para criar aplicativos para os produtos da Apple (ver Figura 3), [9].

Este software é totalmente integrado ao Cocoa e as estruturas do Cocoa Touch, criando um ambiente de desenvolvimento produtivo e de fácil manipulação. O conjunto de

ferramentas inclui o xCode IDE, um ambiente de desenvolvimento com interface gráfica do usuário (GUI). E, ainda, um compilador totalmente integrado.

No xCode, então usado todos os recursos que vão definir o aplicativo: códigos, imagens, sons e outros. O código então é editado e o xCode compila o aplicativo. No GUI é criado todo o design do aplicativo. Este permite criar objetos e editar os seus atributos.

Quando compilado, o iOS Simulator entra em ação. Este nada mais é do que um simulador do seu dispositivo (iPhone; iPod; iPad), para o aplicativo criado. É extremamente útil, para testar se o aplicativo opera do jeito que foi desejado.

A linguagem usada no xCode é o Objective-C [8]. Uma linguagem herdada da linguagem C, de forma mais simples e elegante. Pode ser desenvolvido em Ruby ou Python.

1.2. Desenvolvimento Web

1.2.1. Introdução

Segundo a ONU [10], o número de usuários de internet no mundo, em 2011, alcançou a marca de dois bilhões de pessoas. Sendo um conjunto de redes de computadores interligado pelo mundo, a internet permite a esses usuários ter acesso a informações e se comunicar remotamente. Em termos de conectividade de redes de tecnologia distintas é mais bem sucedido empreendimento do mundo.

Conquistando usuários do mundo inteiro e infiltrando-se nos mais diversos segmentos, a internet faz com que pessoas queiram aparecer cada vez mais na Web e não só acessá-las como meros espectadores.

1.2.2. Páginas Estáticas e Páginas Dinâmicas

World Wide Web (www), é um sistema de documentos de hipertexto que se encontram ligando entre si e que são acessíveis através da internet [11]. Através do protocolo de comunicação HTTP, que realiza a comunicação cliente-servidor, os usuários podem visualizar as páginas eletrônicas (que contenham texto, imagens, vídeos e outros) e navegar nas mesmas por intermédio de hiper ligações.

No nível de serviços, o funcionamento é da seguinte forma: o cliente solicita os dados ao servidor. Este, por sua vez, busca em seu banco de dados às informações que deverão ser formatadas e devolvidas ao cliente. A desvantagem é que, neste módulo, é proporcionado o conteúdo estático, ou seja, uma vez programadas, apresentam sempre a mesmas informações.

Para contornar este problema, foram criadas as páginas dinâmicas. O usuário entra com as informações utilizando formulários HTML, o browser repassa as informações ao servidor que executa um programa, e, enfim, as informações são devolvidas ao cliente. Este programa trata as informações e retornam uma página no formato HTML, ou seja, as páginas são alteradas conforme a implementação. Por isso levam o nome de páginas dinâmicas.

Para deixar a Web mais dinâmica e interativa, criou-se o CGI (Common Gateway Interface). Apesar de dar mais “vida” a Web, programas CGI possuem uma série de desvantagens técnicas, sendo a principal delas o fato de tais programas serem executados de forma diferenciada na Web Server [12]. Um servidor Web que recebe várias requisições simultâneas, facilmente se sobrecarrega e para. Por este motivo, surgiram, e ainda surgem a cada dia, tecnologias alternativas ao uso do CGI: ISAPI, NISAPI, IDC/HTX, Cold Fusion, Java Server Pages (JSP), Personal Home Page (PHP), Active Server Pages (ASP) e outros.

1.2.3. Personal Home Page (PHP)

Visando uma plataforma potente e estável para a programação de páginas do lado do servidor, foram criadas o PHP 3 e o PHP 4 que atenderam de forma satisfatória a comunidade de desenvolvedores. Entretanto, existiam muitos pontos a serem melhorados. Surge, então, a versão 5. O seu principal objetivo foi melhorar os mecanismos de programação orientada a objeto [12].

A partir disso, tudo que pode ser feito em qualquer programa CGI, pode ser feito em PHP. A característica mais importante do PHP é que este trabalha com um grande número de bancos de dados (dBase, Oracle, Sybase, mySQL e outros), e para construção de páginas que exigem este tipo de serviço, o PHP é a melhor opção. Outros pontos são: os custos zeros, além da sua licença é grátis, o código-fonte é aberto e, ainda, dá suporte para os mais importantes protocolos de comunicação, inclusive o HTTP.

Com esses conceitos em mente, pode-se descrever o funcionamento do PHP, de acordo com a Figura 4. O cliente solicita a página da Web através de um Navegador. Entra em ação o servidor, buscando as páginas gravadas, mas com dados que irão ser buscados pelo PHP em seu banco de dados, mostrando ao cliente um arquivo final em HTML.

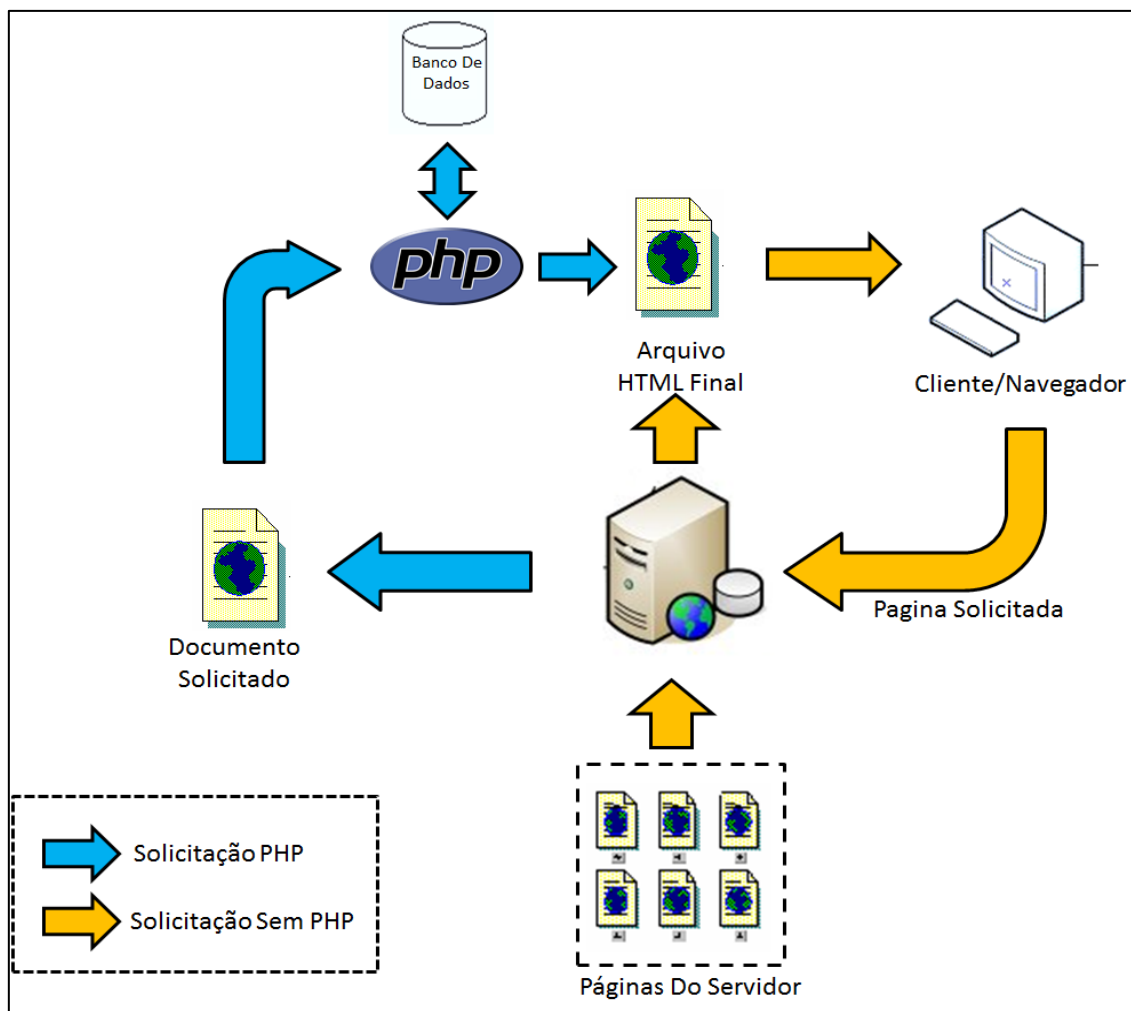


Figura 4 - Esquema de funcionamento do PHP

Note que o cliente não precisa de nenhum programa especial ou pacotes de instalações para que o PHP trabalhe. Este é compatível com todos os grandes navegadores. O PHP é um pré-processador de hipertexto. Em um sentido menos técnico, isso significa que quando o usuário aponta endereço em um navegador, o PHP tem a chance de fazer as alterações de última hora, tornando a página dinâmica.

Assim como o xCode, o PHP trabalha com orientações a objetos: Objective-C.

1.2.4. WAMP Server

Windows, Apache, MySQL, PHP – WAMP. O WAMP une essas quatro ferramentas em um só pacote de instalação realizando a instalação automática e, portanto, agiliza e facilita a instalação dos mesmos.

O sistema é indicado para os usuários que não tem instalado qualquer um dos programas para programar em PHP. Além disso, para os iniciantes, a utilização é recomendada, visto que, o software já faz as principais configurações recomendadas.

Outra vantagem é o uso dos servidores, ao instalar o WAMP 5, cria-se um grupo de programas denominado Wamp Server, onde é possível inicializar o servidor de forma bastante simples. Para testar o funcionamento do serviço, basta iniciar o navegador e escrever na barra de endereço a URL: `http://localhost/`. Dentro do diretório do WAMP 5, existe uma pasta chamada “www”, que corresponde a pasta onde deve-se colocar as páginas Web.

Como explicado, anteriormente, o WAMP é direcionado a plataforma Windows, de forma recíproca, existe a LAMP para o Linux e o MAMP para a plataforma da Apple, Macintosh.

1.3. Orientação a Objetos

1.3.1. Introdução

O termo orientação a objetos, nos trás a ideia de, uma organização de software em termos de coleção de objetos discretos incorporando estrutura e comportamento próprios [13]. Este tipo de abordagem de organização é essencialmente diferente do desenvolvimento tradicional de software, onde estruturas de dados e rotinas são desenvolvidas de forma apenas acopladas.

A computação utiliza métodos de modelagem que nem sempre são similares aos modelos humanos. A orientação a objetos busca conceitos que proporcionem uma aproximação do mundo real com a modelagem, da análise até a implementação computacional.

Na programação orientada a objetos, implementa-se um conjunto de classes que definem os objetos presentes no sistema de software [14]. Cada classe determina o comportamento (definido nos métodos) e estados possíveis (atributos) de seus objetos, assim como o relacionamento com outros objetos.

1.3.2. Objetos

Segundo Martin [14], objeto é “qualquer coisa, real ou abstrata, a respeito da qual armazenamos dados e os métodos que os manipulam”.

Na visão de Rumbaugh et al [15], um objeto é definido como um conceito, uma abstração, algo com limites nítidos e significado em relação ao problema em causa. Os objetos servem a dois objetivos: eles facilitam a compreensão do mundo real e oferecem uma base real para a implementação em computador. A decomposição de um problema em objetos depende do julgamento e da natureza do problema.

Coleman et al [16], encara os objetos como “átomos do processo de computação que trocam mensagens entre si.

Pode-se dizer então que objeto é uma unidade da qual queremos representar no sistema. Um exemplo, o ser humano é representado por informações a seu respeito, que são guardadas em uma ficha cadastral. Esses dados representam um ser humano em particular que ainda tem outras características não presentes nessa ficha. Por exemplo, as capacidades de andar, pensar, falar e outras.

Os objetos são constituídos pelos atributos que o representam e pelas operações que eles realizam. Nesse sentido, pode-se criar um paralelo com os seres humanos, exemplificando a cor, altura, peso entre outros como atributos, e andar, falar, pensar como operações.

Um conjunto de atributos é conhecido por formar o estado de um determinado objeto. As operações, que também são conhecidas por serviços ou métodos, tem a função de manipular o estado do objeto.

1.3.3. Encapsulamento

Também conhecido como ocultação de informação. Diante das leis da orientação a objetos, o único meio de manipular o estado de um objeto é através das operações associadas a ele [14]. Daí, o conceito de ocultação de informação, pois, os atributos não devem ser manipulados diretamente, mas somente, através de funções específicas para tal fim. Essas são visíveis e, por este motivo, podem ser chamadas de interfaces.

Um grande valor agregado a essa técnica é a visão da independência dos dados, pois, a implementação dos métodos pode ser alterada sem comprometer o uso do objeto ou mesmo modificar a forma de utilização.

1.3.4. Classe

Segundo Coleman et al [16], os objetos são agrupados em conjuntos, denominados classes. Uma classe é uma abstração que representa a ideia ou noção geral de um conjunto de objetos similares.

Uma classe é o agrupamento de objetos com características comuns, por exemplo, o indivíduo pertence à classe de seres humanos. Nesse exemplo, cada “homem” é um objeto da classe “humano”. Entende-se por características o agrupamento de atributos e operações, pois, todos tem cor, altura e peso e, além disso, os humanos podem andar, falar e pensar.

1.3.5. Herança

Para Rumbaugh et al [15], herança é compartilhar atributos e operações entre classes tomando como base uma relação hierárquica. É definida como a capacidade de uma nova classe em herdar atributos e métodos pertencentes a uma classe existente. Em termos gerais, pode-se definir uma classe que depois será refinada sucessivamente para produzir subclasses. Todas as subclasses possuem ou herdam cada uma das propriedades de sua superclasse e também suas propriedades exclusivas. Não é necessário redefinir as propriedades das superclasses em cada subclasse

Herança é um mecanismo que permite que as características comuns de um grupo de objetos diferentes sejam concentradas em uma única classe. Essas classes, com nível hierárquico superior, podem ser intituladas como classe-base, classe-mãe, classe-pai, superclasse ou ainda outra atribuição que se refira à sua superioridade. Já as classes herdeiras, assim podem ser chamadas, mas também, classe-filha, subclasse ou classe derivada.

1.3.6. Polimorfismo

Polimorfismo refere-se à capacidade de dois ou mais objetos responderem à mesma mensagem, cada um a seu próprio modo. A utilização da herança torna-se fácil com o polimorfismo. Significa que uma operação pode comportar-se de modo distinto em distintas

classes tendo o mesmo nome de método. Em termos práticos, o polimorfismo permite referir-se a objetos de diferentes classes por meio do mesmo elemento de programa e realizar a mesma operação de forma diferente, de acordo com o objeto a que se faz referência em cada momento.

No mundo real, uma operação é simplesmente uma abstração de comportamento análogo entre classes distintas de objetos. Objetos de classes distintas, podem realizar operações conceitualmente iguais, mas com implementações diferentes. O polimorfismo é a possibilidade de dois objetos de classes diferentes possuírem duas operações com o mesmo nome e implementações diferentes, ou seja, representa as várias formas que um método pode assumir.

As vantagens do polimorfismo residem em facilitar a implementação do conceito de delegação, evitar construções do tipo “case” (if..then e else if...then), simplificar a criação de novas subclasses e ser conceitualmente mais intuitivo. Uma outra vantagem é que uma mesma mensagem pode ser enviada para um conjunto de objetos de classes distintas, ativando vários métodos polimórficos. De maneira prática isto quer dizer que a operação em questão mantém seu comportamento transparente para quaisquer tipos de argumentos; isto é, a mesma mensagem é enviada a objetos de classes distintas e eles poderão reagir de maneiras diferentes. Um método polimórfico é aquele que pode ser aplicado a várias classes de objetos sem que haja qualquer inconveniente [17].

1.4. Controle De Processos Industriais

1.4.1. Introdução

A técnica para manter variáveis de um processo industrial em um valor pré-determinado é chamada de controle de processos. Além de manter essas variáveis, é interessante monitorá-las em tempo real. E o processo se torna mais vantajoso ainda, se for possível fazer a manipulação das mesmas [18], [19], [20].

As vantagens seriam:

- **Qualidade:** É possível determinar os melhores níveis de trabalho para as variáveis do processo. Com o monitoramento pode ser feitas intervenções no processo rapidamente e garantir o produto final.
- **Redução dos custos operacionais:** As leituras dos instrumentos ficam centralizadas, economizando tempo e funcionários especializados.
- **Maior desempenho de produção:** Através da rapidez da leitura dos instrumentos de campo, as intervenções são feitas mais rápida. Problemas de parada de máquina por defeitos podem ser diagnosticado mais pontualmente e os setup's de máquina também são agilizados.
- **Base para outros sistemas:** Os dados podem ser armazenados em banco de dados. Esses dados são utilizados pela a equipe de produção com a finalidade de calcular a eficiência dos equipamentos, estabelecer um planejamento e controle de produção e outros.

Para tanto, é desenvolvida uma interface para o controle dos processos, chamada de SCADA (*Supervisory Control And Data Aquisition*, em português, Sistemas De Supervisão E Aquisição de Dados). O objetivo desses sistemas é propiciar uma interface de alto nível do operador com o processo informando-o “em tempo real” de todos os eventos de importância da planta.

A desvantagem do processo é a necessidade de mão de obra capacitada para desenvolver as interfaces homem máquina (I.H.M).

O sistema SCADA está dividido em funções básicas de controle e supervisão. Podem-se ter funções de supervisão, funções de operação e de controle [18]. O primeiro consiste no monitoramento de processos tais como: sinóticos aninhados, gráficos de tendência de variáveis analógicas e digitais, relatórios em vídeo e impressos, etc. O segundo a função de ligar e desligar equipamentos, operação de malhas PID, mudança de modo de operação de equipamentos. E por último, as funções de controle, que são, por exemplo, o controle digital direto e controle do supervisor.

Na Figura 5, abaixo, é mostrada a arquitetura do sistema SCADA. Geralmente o sistema de controle de tempo real é constituído de unidade separada da estação de supervisão. Pode ser um Controlador Lógico Programável (CLP), controlador multi loop, controlador single loop, ou outros. Essas unidades são conectadas aos sensores e atuadores do processo. Convertem os sinais dos sensores para dados digitais e dados digitais de controle para os atuadores. Em alguns casos a estação de supervisão desempenha a tarefa de executar os algoritmos de controle (DDC).

Este é o esquema básico da arquitetura do sistema SCADA, existem diferentes tipos de arquitetura. Entretanto, em nenhuma delas o software de supervisão tem ligação direta com o equipamento de controle. Todo software de supervisão tem pelo menos uma interface de comunicação (OPC, DDE, Suitelink, ActiveX e etc). Esta interface possibilita “falar” com outro software e este por sua vez possui o protocolo de comunicação com o equipamento de controle. É chamado então de servidor de comunicação que pode possuir um ou mais drivers de comunicação para os equipamentos de controle e serão os próximos capítulos.

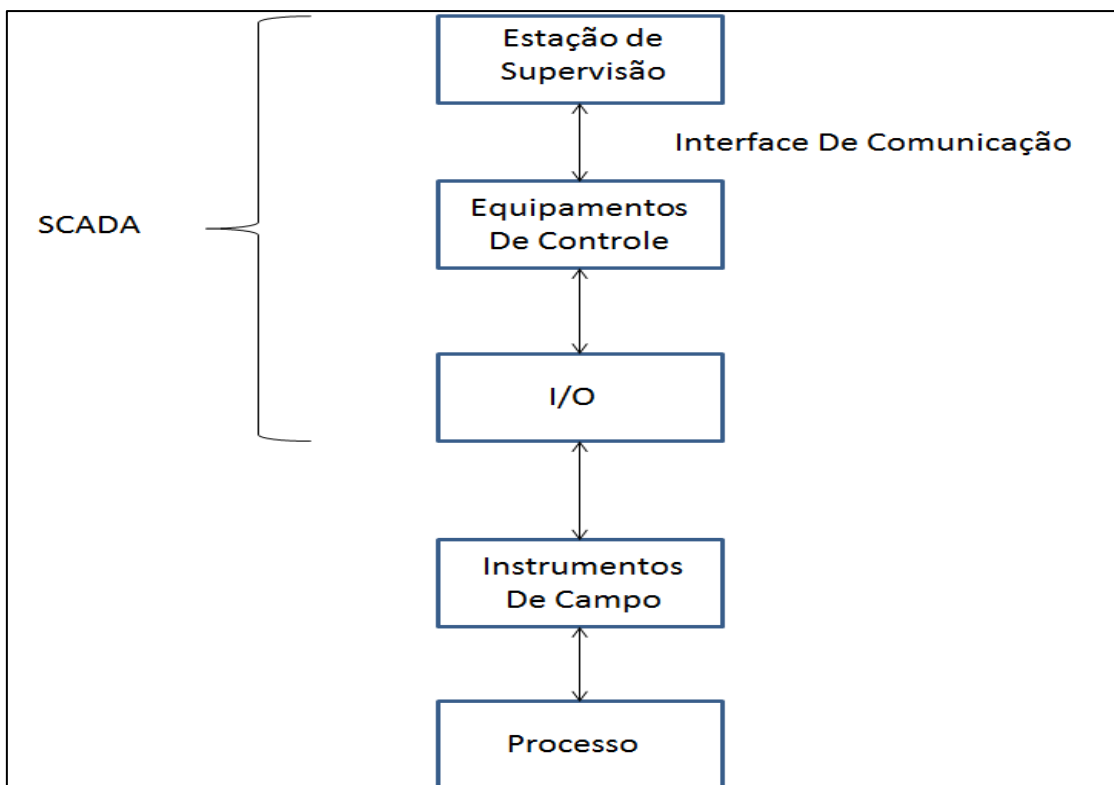


Figura 5 - Arquitetura Básica do Sistema SCADA

1.4.2. Tecnologia OLE/COM

A tecnologia OLE 1.0 foi desenvolvida pela Microsoft em meados de 1990, para suprir a necessidade de se integrar diferentes aplicações dentro da plataforma Windows, de forma a solucionar os problemas de desempenho e confiabilidade do até então utilizado padrão DDE (Dynamic Data Exchange) [18]. Nessa época, introduziram-se dois conceitos: Linking – Cria vínculos ou referências aos objetos, armazenando no documento principal apenas os dados realmente necessários para exibir, imprimir, etc. Embedding – Incorpora os dados dos objetos ao documento principal. Neste contexto, surgiram os conceitos de objeto vinculado e de objeto incorporado:

- Objeto Vinculado – São informações (objetos) criadas em um arquivo (arquivo origem) e inseridas em outro arquivo (arquivo destino). Embora o objeto vinculado não se torne parte do arquivo de destino, existe um vínculo, uma conexão entre os dois arquivos de forma que o objeto vinculado no arquivo de destino seja automaticamente atualizado quando o arquivo de origem é atualizado.
- Objeto Incorporado – São informações inseridas em um arquivo de destino. Ao ser incorporado, o objeto se torna parte do arquivo. Ao clicar duas vezes no objeto incorporado, ele é aberto no programa de origem em que foi criado. Qualquer alteração feita no objeto incorporado se refletirá no arquivo de destino.

Outro conceito importante na tecnologia OLE é o conceito de Cliente Servidor:

- Cliente – Aplicação que solicita os dados.
- Servidor – Aplicação que disponibiliza os dados.

No OLE 2.0, houve muitos aperfeiçoamentos dentre os quais o mais importante é a Automação OLE, pois permite que uma aplicação seja controlada por outra aplicação.

A tecnologia OLE é montada sobre a tecnologia COM (Component Object Model) que define um modo padronizado para um módulo-cliente e um módulo-servidor se comunicarem por meio de uma interface específica [18]. Módulo indica um aplicativo ou uma biblioteca (uma DLL – Dynamic Link Libraries). Os dois módulos podem ser executados no mesmo computador ou em máquinas diferentes conectadas através de uma rede. Muitas interfaces são

possíveis, dependendo do papel do cliente e do servidor, e podem-se incluir novas interfaces para propósitos específicos.

1.4.3. Tecnologia OPC

Geralmente, quando uma aplicação necessita acessar informações de um equipamento de controle deve ser desenvolvida uma interface customizada. Entretanto, muitas das aplicações não conseguem acessar essas informações devido a inconsistência entre fabricantes de drivers e hardwares [18].

Para contornar esses problemas, foi desenvolvida a tecnologia OPC, que é uma tecnologia para conectar aplicações Windows a equipamentos de controle. Além disso, o OPC é um protocolo de comunicação aberta que permite um método consistente de acesso aos dados de inúmeros equipamentos dos mais diversos fabricantes [19].

O OPC é construído usando o protocolo OLE/COM, para atender as necessidades gerais da indústria e não as necessidades específicas de qualquer fabricante de hardware e software.

No quesito arquitetura, este protocolo pressupõe três objetos básicos: servidor, grupo e item, que são mostrados na Figura 6. Um servidor é essencialmente uma estrutura de armazenagem para grupos, que por sua vez, tem como objetivo básico o armazenamento de itens. Os itens representam conexões ponto a ponto de entrada e saída, ou seja, este não representa um valor, e sim, um meio de acessá-lo. Os grupos reúnem os itens que interessam a um determinado cliente, assumindo o papel principal na interação cliente servidor.

O item é uma estrutura a qual estão associadas três propriedades:

- *Value*: último valor armazenado pelo servidor no cache de memória do item e que é atualizado sempre que o servidor faz uma leitura no dispositivo
- *Quality*: informação de estado que define a qualidade do dado que ser: *Good*; *Bad* ou *Uncertain*.
- *Time Stamp*: tempo de amostragem

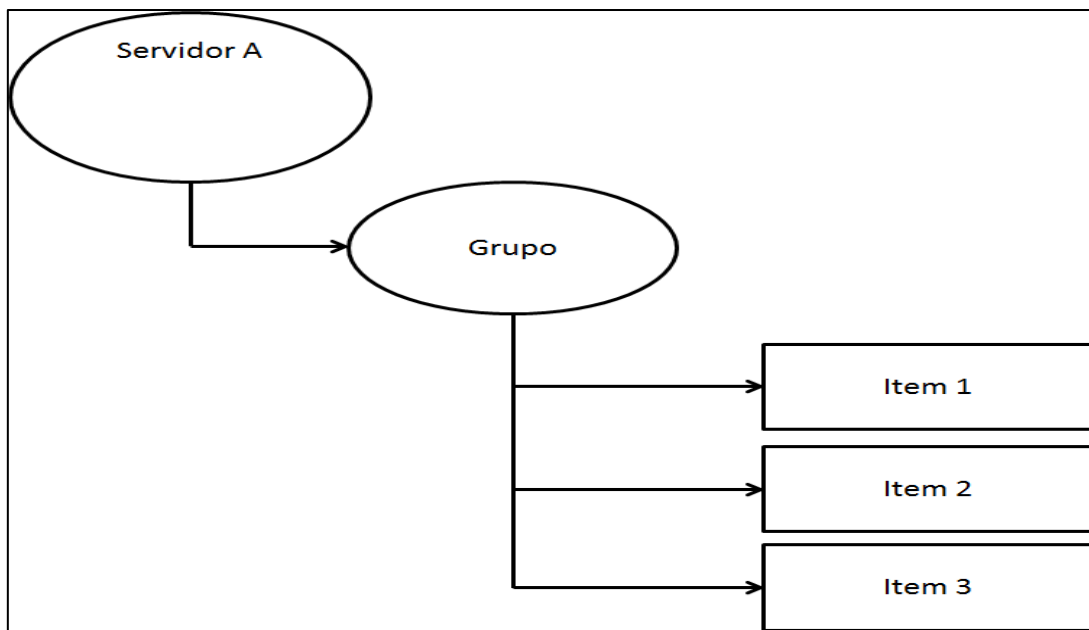


Figura 6 - Arquitetura OPC

1.4.4. Matlab e OPC

Matlab é um programa interativo, desenvolvido, para cálculos numéricos, uma poderosa ferramenta computacional. Muito utilizado no meio acadêmico. Além do programa principal, o programa inclui uma gama de toolboxes, uma coleção de arquivos chamados m-files que estendem a funcionalidade do programa principal [19].

A partir do Matlab 7.0 [19], contém o OPC toolbox, uma coleção de ferramentas capaz de estabelecer comunicação com os servidores OPC, assumindo o papel de cliente OPC.

2. Metodologia

2.1. Desenvolvimento Geral

A ideia geral do trabalho, aqui realizado, é mostrada na Figura 7. Consiste, então, no desenvolvimento de um aplicativo para o iPhone, para tanto é utilizado o xCode (uma poderosa ferramenta para a criação de aplicativos para os dispositivos da Apple), disponibilizado pelo próprio sistema operacional da Apple, Mac OS X.

O aplicativo será descrito com mais detalhes posteriormente. Mas, vale comentar, que ele se comunica com um servidor. O servidor é composto pelo o PHP, por um arquivo TXT e o Matlab. Ao receber as instruções via web, o PHP escreve as solicitações no arquivo TXT. O Matlab, por sua vez, lê este arquivo TXT e envia o comando para a planta. O caminho da comunicação de “volta” também foi implementado e o princípio de funcionamento é idêntico. As variáveis controladas pelo cliente são duas bombas presente na planta.

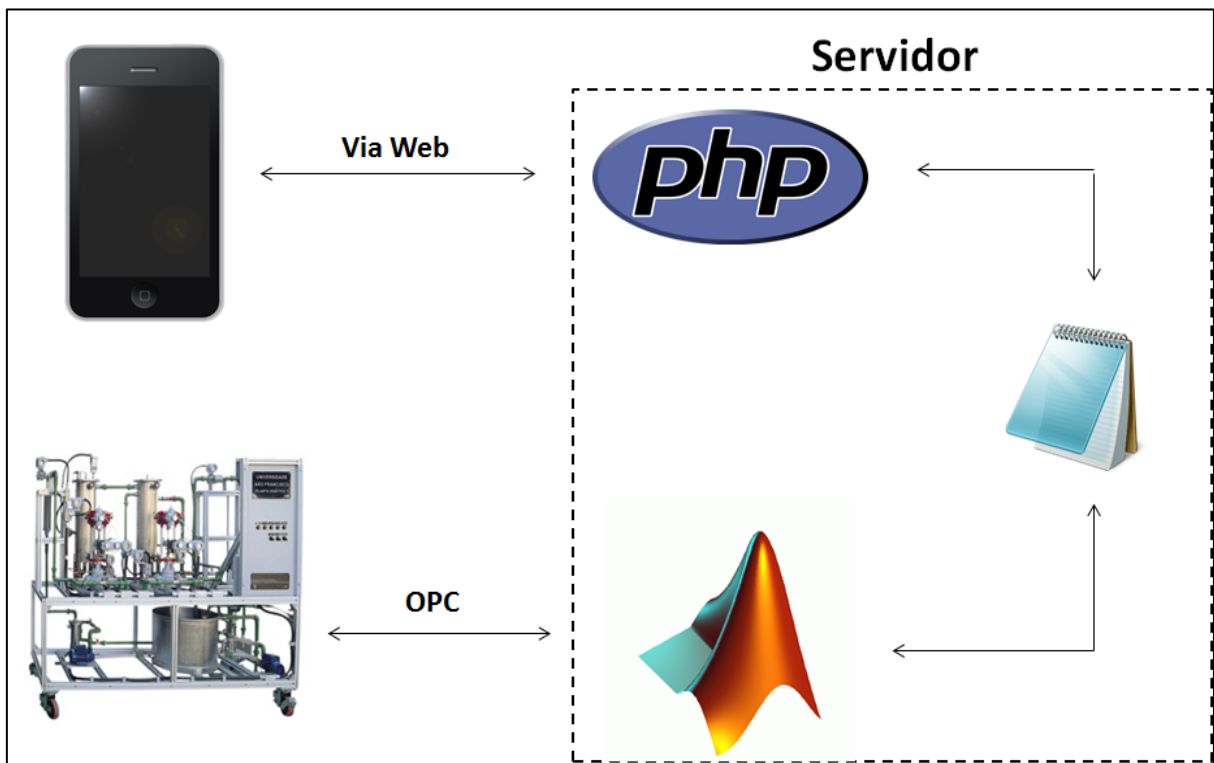


Figura 7 - Esquema geral

Para que o cliente tenha informações do que está acontecendo na planta, foi desenvolvido o processo de leitura da planta. Ou seja, da mesma forma que o smartphone envia informações, é desejável que o cliente receba informações, e isto é feito através de um gráfico, implementado no iPhone, que gravará, até mesmo o histórico das bombas.

2.2. Desenvolvimento do Aplicativo

Nesta seção é descrito detalhadamente como foi feito o desenvolvimento do aplicativo.

Primeiramente, deve-se ter em mente que o protocolo de comunicação OPC, é desenvolvido para a plataforma Windows. E o software xCode é disponível apenas para a plataforma Mac OS X.

É preciso, então, que as duas plataformas rodem ao mesmo tempo e por questões de praticidade e econômica, é necessário que isso aconteça em apenas um computador. Portanto o primeiro passo consiste na instalação de uma máquina virtual que comporte um dos sistemas operacionais. A máquina virtual escolhida chama-se VMWare Workstation. Com isto, foi possível a instalação do sistema operacional Mac OS X, Figura 8.



Figura 8 - Windows e MAC OS X trabalhando juntos no VMWare Workstation

Após instalado o VMWare Workstation e o Mac OS X, fez-se o download do xCode, o download é feito através do iTunes. Este além de ser um reproduutor de áudio e vídeo, conta com uma espécie de mercado virtual que permite ao usuário comprar arquivos de mídia digital dentro do próprio programa. Um desses arquivos é o: xCode – software livre. Vale ressaltar que o iTunes, já está vem instalado no sistema da Mac, Figura 9.

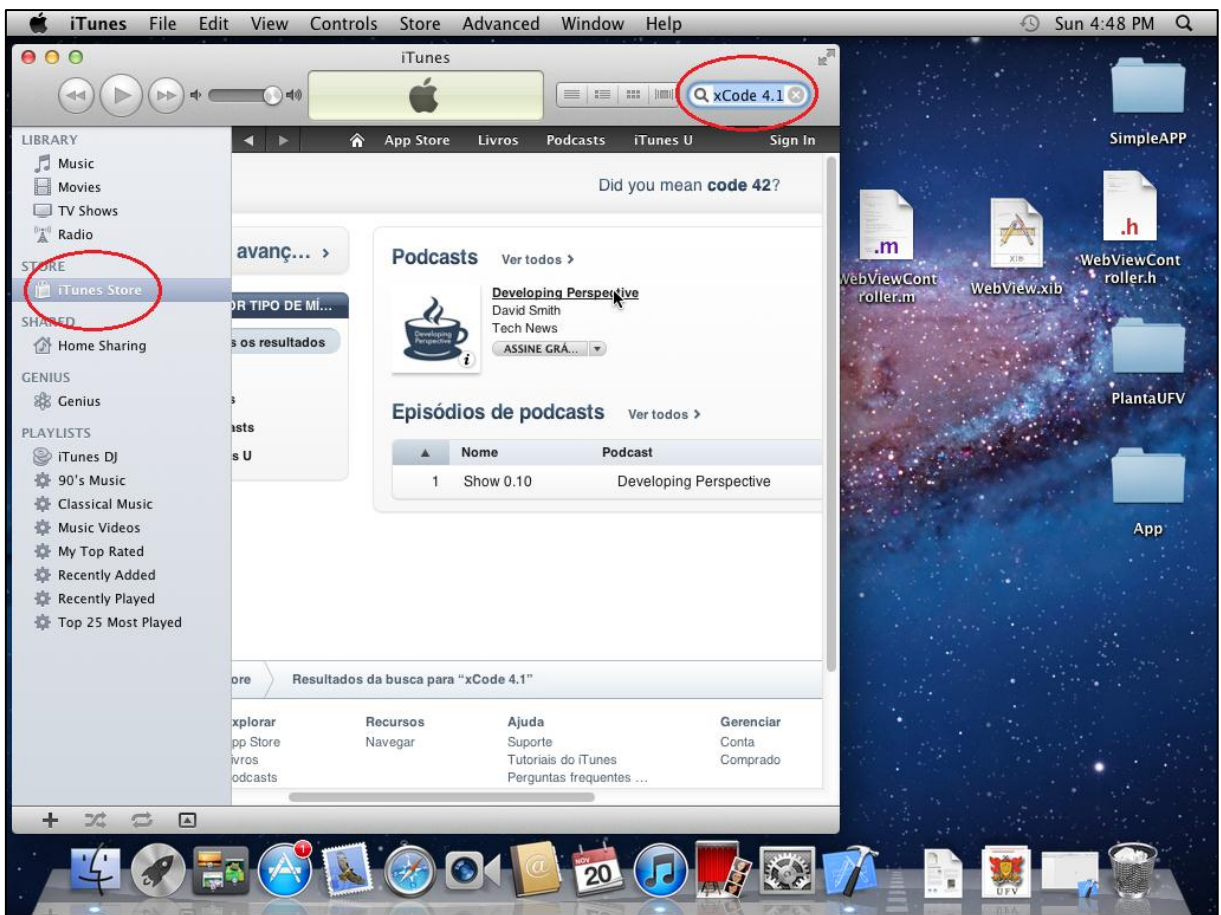


Figura 9 - Ambiente Mac OS X e iTunes

Tem-se em mãos agora o xCode. Inicia-se o xCode, cria-se um novo projeto com o xCode. Uma nova janela é aberta e devemos escolher um modelo de projeto. Observe que existem duas categorias: A primeira a categoria é dirigida aos dispositivos da Apple (caso aqui presente); e a segunda, aplicativos para o computador portátil Mac. Dentro de cada um dessas categorias é possível ainda escolher o modelo do projeto, foi escolhido o *View-Based Application*. Informações de cada modelo podem ser visto, ao clicar, em cima de cada modelo Figura 10.



Figura 10 - Modelos para a criação de aplicativos

Seguindo adiante, será aberto o editor. As primeiras configurações são a escolha do tipo de dispositivo, a imagem do ícone, e se o aplicativo iniciará “deitado” ou “em pé”. Também, são criados vários arquivos a esquerda do editor, Figura 11. Os arquivos com extensão .h, .m e .xib. No primeiro, são definidos os objetos, por exemplo: se desejar adicionar um botão ao aplicativo é nesse script que se cria o botão. No plano .m, é implementado as ações do botão. E, para finalizar, no arquivo .xib é montada o aplicativo, no sentido físico.

Ainda no arquivo .xib para que o botão físico haja de acordo com o que é programado, é necessário instancia-lo. Para tanto o xCode possui um pequeno cubo (Figura 11 - Canto superior do quadrado amarelo), onde ao clicar com o botão direito em cima dele, está todas as funções criadas no arquivo .h. E então, basta puxa-lo ou instanciá-lo até o objeto físico, Figura 11 (Mouse puxando a função do botão: *SlideView*).

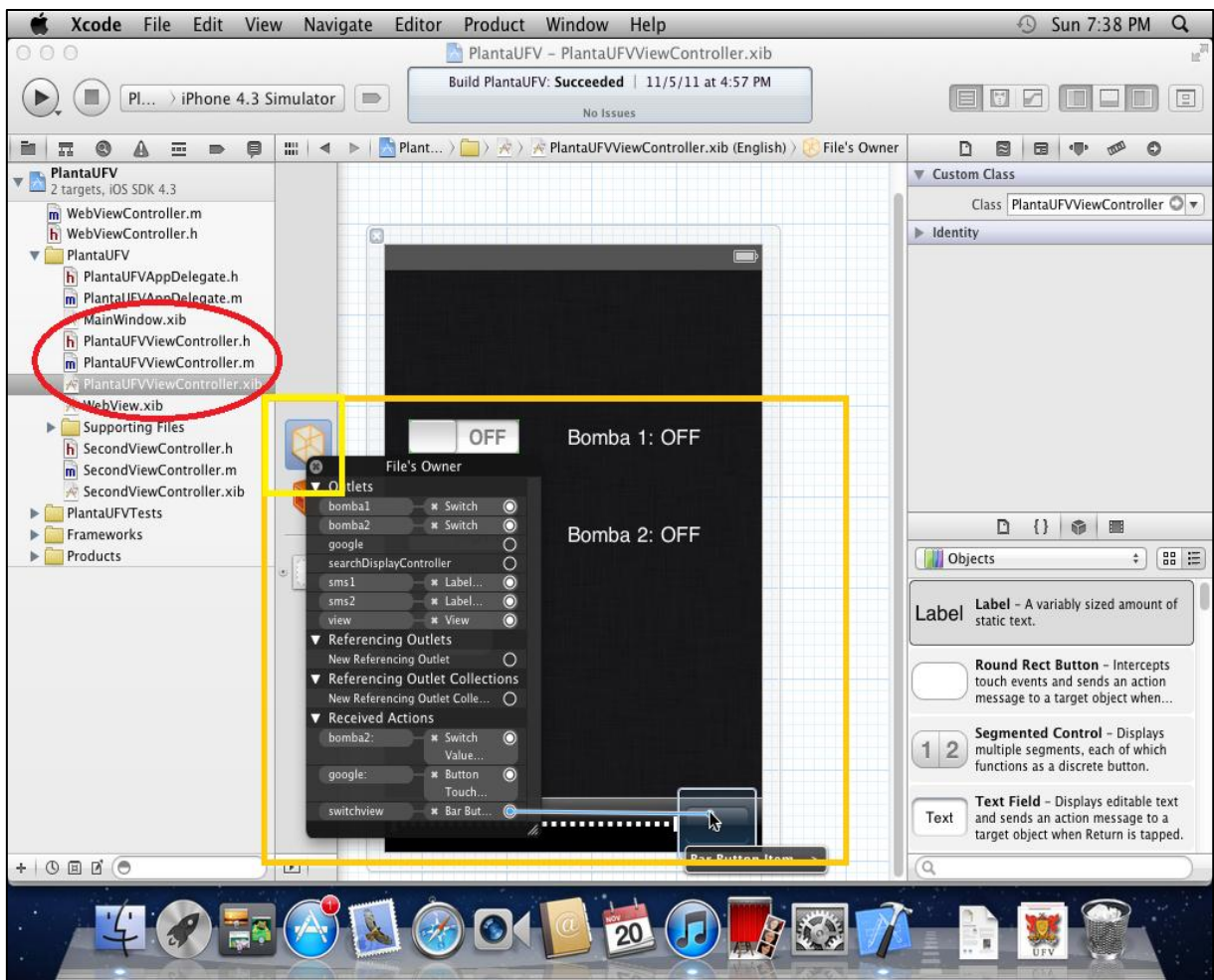


Figura 11 - Arquivos da view-based application

No presente caso, foram adicionados dois botões do tipo *ON/OFF*, um botão tipo *PUSH*, duas caixas de texto, um botão de *SlideView* (este gira a tela, e inicia uma nova tela), nesta nova tela tem-se um botão do tipo *PUSH*, uma *label* de *UIImageView* (uma caixa destinada somente a imagens) e um botão de *SlideViewBack*, ver Tabela 1 e Tabela 2.

Para tanto, todos os elementos foram definidos no arquivo *.h*, e suas ações implementadas no arquivo *.m* e foram instanciados na *BOX* de instância, Figura 11.

A função de cada um dos botões tipo *ON/OFF* é de chamar uma página específica da web. A função das caixas de texto é informar o *status* da variável. E o outro botão do tipo *PUSH* da segunda *View* é de carregar uma imagem de uma página específica, Tabela 1 e Tabela 2.

Tabela 1 - Objetos da *view* principal

Item	Tipo	Função
Botão	ON/OFF	Ligar e Desligar Bomba 1
Botão	ON/OFF	Ligar e Desligar Bomba 2
Botão	PUSH	Ir para View Secundária
Botão	PUSH	Uso Geral
Caixa De Texto	-	Informar <i>Status</i> da Bomba 1
Caixa de Texto	-	Informar <i>Status</i> da Bomba 2
TAB BAR	-	Barra de Navegação

Tabela 2 - Objetos da *view* secundária

Item	Tipo	Função
Botão	PUSH	Carregar Histórico
Botão	PUSH	Voltar para View Principal
UIImageView	-	Carregar Imagem
TAB BAR	-	Barra de Navegação

Nas barras de navegação estão localizados os botões de “ida” e “volta” da *View*. Não apresentam função nenhuma em si, são apenas para organização e estética do aplicativo. Além disso, podem obter outros botões para outras funções.

Observe que na Figura 11, existem outros arquivos, *WebView* e o *SecondControllerView*. Como foi mencionado, anteriormente, os botões ON/OFF chamam uma página da Web, então o programa *WebView* faz esse papel, este faz com que suba uma janela que é a página da Web chamada. Essa janela já possui um botão para voltar para o painel principal. O segundo é apenas para trazer a segunda *view* do aplicativo, que é onde se encontra o histórico das variáveis.

2.3. Desenvolvimento Web

Continuando o nosso procedimento, agora no lado do servidor. O WAMP Server libera a porta 80 e então é possível que qualquer pessoa tenha acesso a este website. Ao acessar este website, o PHP toma a decisão de acordo como foi implementado.

Então, neste trabalho, toda vez que determinado site é acessado, o PHP escreve em um arquivo TXT a solicitação feita. Se o botão *ON/OFF* da bomba 1, for colocado no estado ON, então é chamado um website. Neste momento é enviada para o cliente a mensagem: “A bomba 1 foi ligada”. Além disso, o PHP escreve no arquivo de texto a seguinte informação: bomba 1: 1. Indicando que a bomba 1 está no estado alto.

Se a bomba já estiver ligada, ou seja, se no arquivo de texto estiver escrito: “bomba 1: 1”, o servidor envia a seguinte mensagem: “A bomba 1 já está ligada”.

A programação em PHP é feita em um arquivo de TXT, ou seja, tudo é escrito em ASCII (arquivos de texto), no momento em que for salvar o arquivo, este deve estar com a extensão .php, e deve ser salvo dentro da pasta “www” do WAMP Server. O Wamp Server carregará as páginas online. Vale lembrar, que a sintaxe da programação PHP é a mesma da xCode, que por sua vez é similar a C/C++ e Perl

Além disso, a extensão dos URL foi escolhida de forma bastante intuitiva. Para estabelecer esta, basta que no momento em que for salvar o arquivo com a extensão .php salvar com o nome desejado. Na Tabela 3, abaixo, pode-se ver a URL escolhida para cada uma das ações.

Tabela 3 - Ação e URL

Ação	URL
Ligar Bomba 1	http://localhost.com/Bomba1_ON.php
Desligar Bomba 1	http://localhost.com/Bomba1_Off.php
Ligar Bomba 2	http://localhost.com/Bomba2_ON.php
Desligar Bomba 2	http://localhost.com/Bomba2_Off.php
Verificar Histórico	http://localhost.com/gráfico.php

2.4. Desenvolvimento da comunicação com a planta

Uma vez que as solicitações do cliente estão escritas no arquivo de texto, é preciso que essas informações sejam lidas e enviadas para a planta. Para que isso ocorra, ainda no servidor, ou seja, na máquina, o software Matlab faz a leitura dos arquivos de texto e manda as informações via OPC à planta.

O primeiro passo, é implementar um loop infinito de leitura dos dados. Assim que, a informação mudar de estado, é enviado o estado para a planta. Por exemplo: se no arquivo TXT está definido como “Bomba 1: 0” e o PHP alterar esta informação para “Bomba 1: 1”, então o Matlab envia a informação para a planta e a bomba é ligada.

O segundo passo para se conectar é entender como o OPC toolbox é implementado. Mas, antes disso, é preciso implementar a lógica da planta. Este é feito através do *LogicView* usando a programação *Ladder* – Uma linguagem gráfica para a programação dos Controladores Lógicos Programáveis (CLPs) no qual as funções lógicas são representadas através de contatos e bobinas.

Como o objetivo do trabalho aqui desenvolvido é apenas o controle *ON/OFF* de duas bombas específicas, a programação *Ladder* é simples e mostrada na Figura 12. A bomba número 2 é implementada de forma análoga.

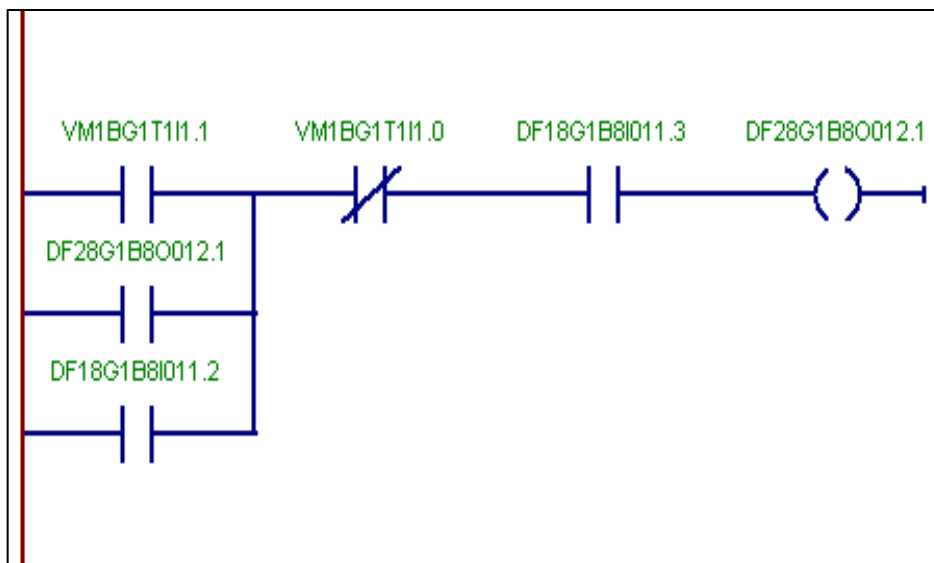


Figura 12- Lógica *Ladder* da bomba 1

Na Tabela 4, abaixo é mostrada o que representa cada item da Figura 12.

Tabela 4 - Nome e representação dos itens da lógica *Ladder*

Nome Do Item	Representação	Estado Inicial
VM1BG1T1 1.1	Botão Virtual: ON	0
VM1BG1T1 1.1	Botão Virtual: OFF	0
DF18G1B81011.2	Botão Manual: ON	0
DF18G1B81011.3	Botão Manual: OFF	0
DF28G1B80012.1	Bomba 1	0

Observa-se que a bomba 1, tem parâmetro como entrada e como saída.

Na Tabela 5, é mostrado alguns possíveis resultados para o acionamento da bomba 1. Esta foi montada seguinte um ciclo, note que toda vez que BT Virtual *OFF* ou BT Manual *OFF*, for acionado a bomba é desligada. Para evitar que o operador ou o cliente fique ligando

ininterruptamente, quando esta já estiver ligada, utilizou-se um contato de “selo”. Novamente, a bomba 2 é de forma análoga e está omitida.

Tabela 5 – Exemplo de Ciclo de Funcionamento

BT Virtual ON	BT Virtual OFF	BT Manual ON	BT Manual OFF	Bomba 1 Entrada	Bomba 1 Saída
0	0	0	0	0	0
1	0	0	0	0	1
0	0	0	0	1	1
0	1	0	0	0	0
0	0	1	0	0	1
0	0	0	0	1	1

Feito esta etapa, é preciso entender alguns pontos da comunicação OPC, ligadas a programação. Este está dividido em três objetos básicos dispostos em uma hierarquia específica, que são descritos abaixo:

- *OPC Data Access Client Object* (OPC-DA): representado o cliente, tem a função de armazenar os grupos e administrar a sua conexão com este.
- *Data Access Group Objects* (dagroup): representam os grupos contidos no OPC-DA e armazenam os itens deste grupo. Tem a função de administrar como os itens devem ser lidos e armazenar históricos.
- *Data Access Group Objects* (daitem): representam os itens, também conhecidos como variáveis, que são lidos ou escritos pelo servidor nos dispositivos de campo. Os itens existem somente dentro de um dagroup.

E então, para iniciar a comunicação é necessária a função “*opcda*”, onde os parâmetros são o local onde está instalado o servidor e nome do servidor ao qual se deseja conectar.

Um *dagroup* pode ser acrescentado em um OPC-DA a partir da função “*addgroup*”. O mesmo pode ser feito em relação ao *dagroup* por meio da função “*additem*”. Para esta última, devem ser informados os nomes das variáveis que desejam obter dados ou enviar dados. Para ler ou escrever dados, deve-se usar as funções “*read*” e “*write*”, respectivamente.

Para finalizar a comunicação, deve-se usar a função “*disconnect*”.

Um passo importante que deve ser feito é a exportação de *tags*. Este irá exportar os nomes das variáveis para que o Matlab possa entender e adicioná-los como itens.

Até o momento, foi detalhado o procedimento de “ida” da Figura 7. Todo o procedimento de “volta” foi implementado de forma análoga. Em suma, se ocorrer um desligamento manual da bomba, o item adicionado no Matlab é alterado, esta informação é escrita no arquivo TXT, para que o PHP possa informar ao cliente.

Além disso, o Matlab, salva todo o histórico em um vetor. Assim que o cliente solicita o histórico, o Matlab plota esta figura e salva na pasta “*www*”, o PHP por sua vez, carrega esta figura em um website, e o smartphone a reproduz pela *UIImageView* – ferramenta com a função de carregar imagem.

3. Resultados

Os resultados apresentados a seguir, são feitas a partir do IOS Simulator. Ao clicar em *Product* e em *Run* da Figura 11 o simulador IOS reproduz o dispositivo virtual da Apple. O iPhone virtual simula todas as funções do dispositivo real, exceto, claro, ligações e serviços de mensagens, Figura 13.



Figura 13 - IOS Simulator

Observa-se que o ícone do aplicativo já está criado. Na Figura 14, pode-se ver as duas *views* do aplicativo. A *view* principal (à esquerda) e a *view* secundária (à direita). Ao clicar no botão de *Slide View*, em destaque vermelho, uma transição para a *second view* ocorre. Observa-se que a *second view* (à direita da Figura 14) possui um botão de *slide back*, ou seja, pode-se voltar para a *view* principal. Não é possível mostrar na Figura 14, porém, na transição

das *views*, tem-se uma pequena animação, como se estivesse virando uma página de um caderno.

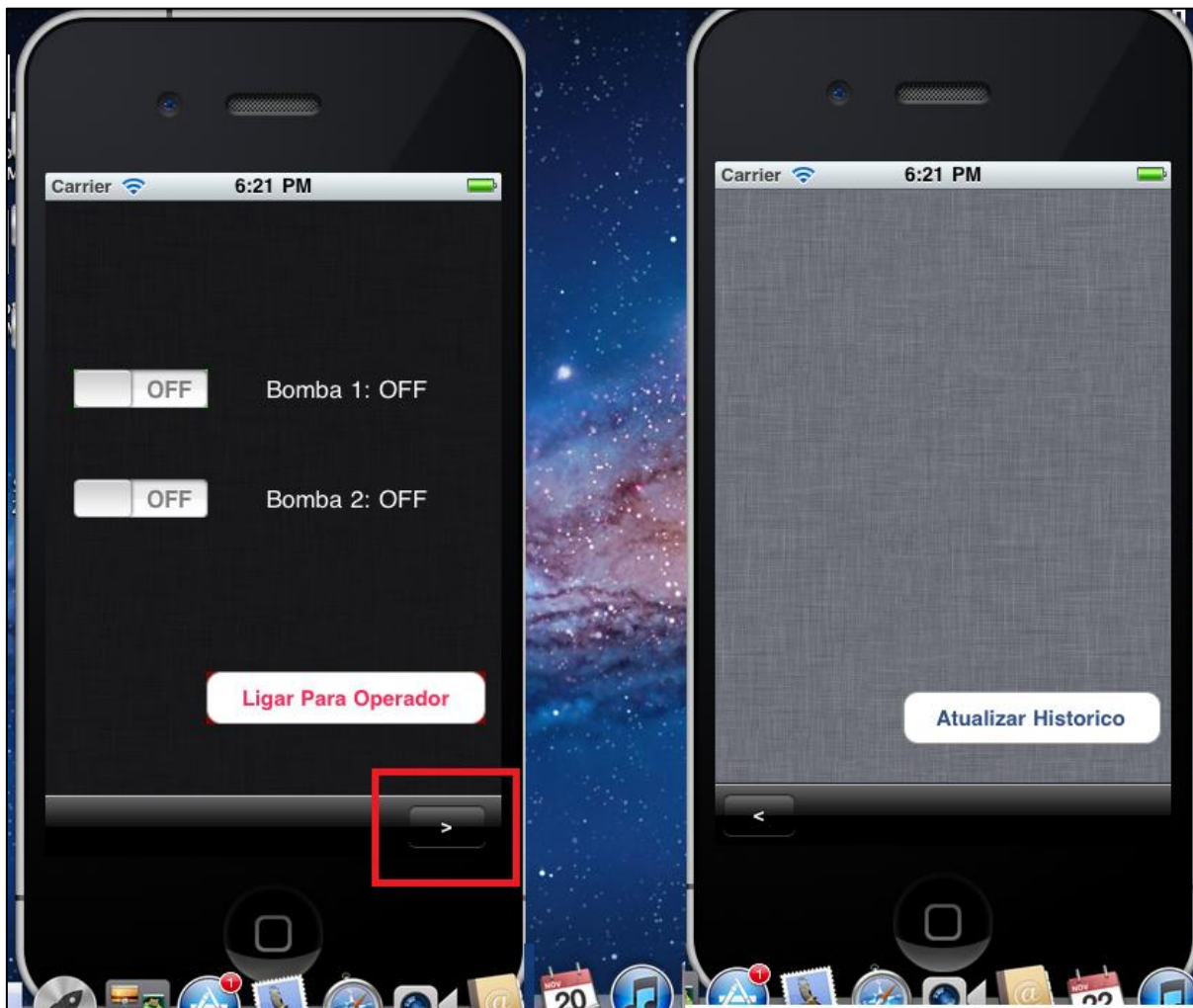


Figura 14 - *View* principal (esquerda) e secundária (direita)

Na, Figura 14, existe um botão de uso geral, chamado de: “Ligar Para Operador”. Como dito anteriormente, o *IOS Simulator* não possui essa função, entretanto, se for feito o download deste aplicativo para um iPhone, ao clicar no botão, um ligação será feita para um número pré-determinado (0123456). Ainda na Figura 14, mas com relação a segunda *view*, esta mostra-se bastante vazia, mas será visto, adiante, que o histórico do processo, ao ser carregado, ocupará praticamente toda a tela do dispositivo virtual.

Os botões *ON/OFF* são alterados ao escorregar a ponta do dedo em cima da sua parte mais sólida. No caso, o escorregar o mouse, uma janela sobe e é apresentada uma mensagem ao cliente, Figura 15.



Figura 15 - Resultado ao ligar bomba 1

Observa-se que a caixa de mensagem, ao lado do botão *ON/OFF* é atualizado também. Na Figura 15, a mensagem (“*Status da Bomba 1: A Bomba 1 Foi Ligada!*”), não está com uma resolução muito boa. Este fato é devido, as configurações do Word, porque na prática é possível ver claramente a mensagem. O mesmo acontece com a segunda bomba, Figura 16, o botão *ON/OFF* é atualizado e uma mensagem é enviada ao cliente sobre o *status* da bomba 2. Se a bomba já estiver ligada uma mensagem diferente é mostrada ao cliente, esse caso pode

ser visto na Figura 17, a mensagem é: “*Status* da Bomba 2: A Bomba 2 já está ligada!”. Novamente, ilegível na Figura 17, entretanto, na prática é bastante nítida.



Figura 16 - Resultado ao ligar a bomba 2

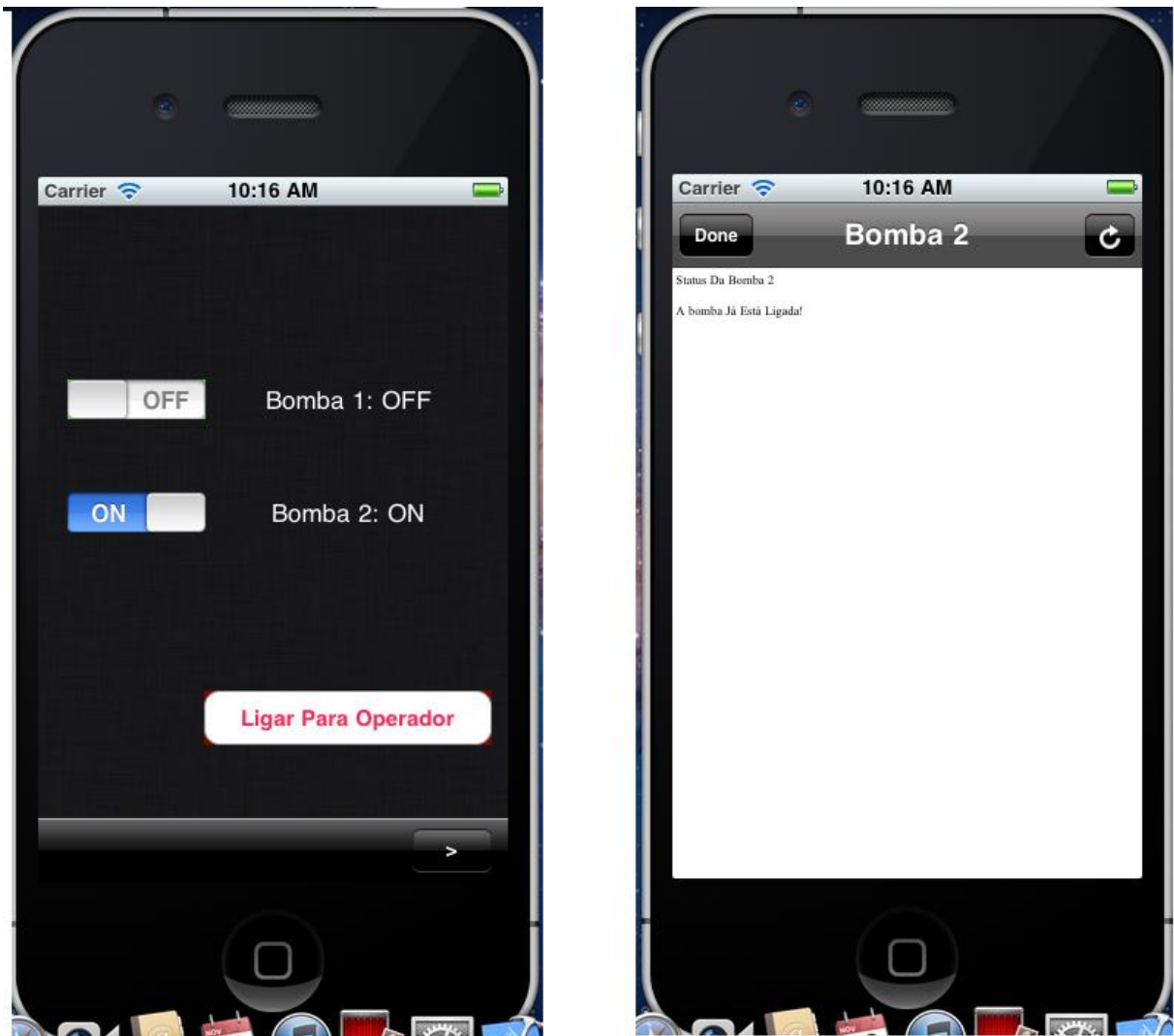


Figura 17 - Resultado ao ligar a bomba 2 que já está ligada

E para finalizar, analisa-se o resultado da função histórico. Supõe-se a seguinte situação: a bomba 1 é ligada e depois de alguns segundos desligada, ambas pelo aplicativo. E então é ligado, manualmente, e, por fim, desligado pelo aplicativo. Atualiza-se o histórico e verifica-se o *status* da bomba.

O resultado disso no servidor é mostrado na Figura 18. No eixo da abscissa (eixo x), tem-se a hora da ação tomada. No eixo das ordenadas, a bomba é interpretada da seguinte forma: se estiver no nível 1 (nível alto) a bomba está ligada, caso esteja em nível 0 (nível baixo), a bomba está desligada.

Na Figura 19 é mostrado o histórico do smartphone.

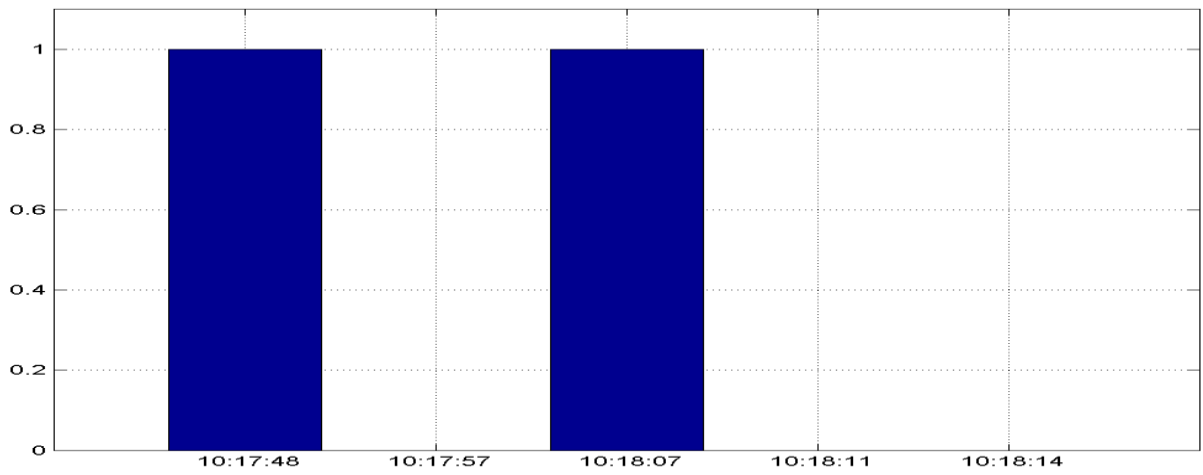


Figura 18 - Histórico salvo no servidor



Figura 19 - Histórico no smartphone

Vê-se na Figura 19, que o download da imagem não é completamente visível, através do smartphone virtual. Porém, no dispositivo real é possível dar zoom com as pontas dos dedos e ter maiores detalhes do resultado. Observa-se, porém, que mesmo que a bomba desligue manualmente, nem o campo de texto e nem o botão *ON/OFF* são atualizados.

4. Conclusão

O aplicativo desenvolvido foi para o controle *ON/OFF* de duas bombas da planta didática da Smar. Com os resultados obtidos, foi possível concluir que todo o protocolo de comunicação de enviar e receber dados pelo dispositivo funcionou.

Ainda que, os botões tipo *ON/OFF* não se atualizam automaticamente, este informa se a bomba já está ligada ou desligada. E o aplicativo permite o cliente checar o valor da variável naquele momento, assim como, verificar o histórico do estado da bomba 1, atendendo então as necessidades do cliente.

Os programas, desenvolvidos pela Apple, permitem desenvolver uma interface mais amigável para o cliente, e portanto, contribuem para o comodismo no mesmo.

Como sugestões para trabalhos futuros propõe-se a substituição do arquivo TXT para um arquivo tipo EXE (executável), afim de, diminuir o número de ciclos gasto pela máquina. Este tipo de protocolo é chamado de comunicação entre processos - IPC. E também aumentar o número de variáveis a serem controladas. O iPad, por ser maior, pode facilitar a visualização dessas variáveis.

5. *Referências Bibliografia*

- [1] ALTENBURG, B.; CLARKE, A.; MOUGIN, P. Cocoa. Become An XCoder - Start Programming the Mac Using Objective-C, 2008. Disponível em: <<http://www.cocoalab.com/BecomeAnXcoder.pdf>>. Acesso em: 15 ago. 2011.
- [2] MORIMOTO, C. E. Smartphone Guia Prático. In: MORIMOTO, C. E. Smartphone Guia Prático. 1ª. ed. Porto Alegre: Meridional Ltda., 2009. p. 432.
- [3] IDC - Analyze The Future. Disponível em: <www.idc.com>. Acesso em: 16 Novembro 2011.
- [4] Technology Research | Gartner Inc. Disponível em: <www.gartner.com>. Acesso em: 16 Novembro 2011.
- [5] STRATEGIC Analysis Inc. Corporate Website. Disponível em: <www.sainc.com>. Acesso em: 15 Novembro 2011.
- [6] WmCann. Disponível em: <www.wmccann.com.br>. Acesso em: 16 Novembro 2011.
- [7] NIELSON Brasil. Disponível em: <www.br.nielsen.com>. Acesso em: 5 Novembro 2011.
- [8] APPLE Inc. Disponível em: <www.apple.com>. Acesso em: 10 Novembro 2011.
- [9] HILLEGASS, A. Cocoa Programming FOr Mac OS X. 3. ed. [S.l.]: Addison-Wesley, Maio.
- [10] ONU United Nations. Disponível em: <www.un.org/en/>. Acesso em: 2011 Novembro 18.
- [11] BREITMAN, K. K. Web Semântica. Rio de Janeiro: LTC, 2005.
- [12] SICA, C. PHP Orientado A Objetos: Fale a Linguagem da Internet. 1ª. ed. Rio De Janeiro: Ciência Moderna, v. 1, 2006.
- [13] FERREIRA, M.; JARABECK, F. CA Visual Objects - O Livro. São Paulo: Express, 1995.

- [14] MARTIN, J.; ODELL, J. Análisis y diseño orientado a objetos. 2ª. ed. México: Prentice Hall, 1994.
- [15] RUMBAUGH, J. Modelagem e projetos baseados em objetos. Rio De Janeiro: Campus, 1994.
- [16] COLEMAN, D. Análise baseada em objetos. Rio De Janeiro: Campus, 1992.
- [17] JACOBSON, J. Object-oriented software engineering: a use case. California: Addison-Wesley, 1992.
- [18] FONSECA, M. O. Comunicação OPC - Uma abordagem Prática. VI Seminário de Automação De Processos, Associação Brasileira de Metalurgia e Materiais, Vitória - ES, Brasil, 2002.
- [19] DUARTE, C. R. M.; FIGUEIREDO, L. C. Utilização do Matlab no Ensino da Tecnologia OPC Aplicada a Controle De Processos. Anais Do XVI Congresso Brasileiro de Automática, 2006.
- [20] OPC Foundation. OPC Overview, 1998. Disponível em: <www.opcfoundation.org/>. Acesso em: Setembro 15 2011.
- [21] SOUZA, L. C. A. Padrão De Acesso a Dados OPC e sua Implementação em Driver OPC-MODBUS. V Simpósio Regional de Instrumentação/ II Congresso Mineiro de Automação, ISA/GRINST - IBP. Belo Horizonte: [s.n.]. 1998. p. 157-164.
- [22] DORF, R. C. Sistemas de Controle Moderno. 8ª. ed. Rio De Janeiro: Livros Técnicos e Científicos, 2001.