



**UNIVERSIDADE ESTADUAL DA PARAÍBA
CAMPUS VII GOVERNADOR ANTONIO MARIZ – PATOS-PB
CENTRO DE CIÊNCIAS EXATAS E SOCIAIS APLICADAS
CURSO DE LICENCIATURA EM COMPUTAÇÃO**

EDNALDO NUNES DE SOUZA SEGUNDO

**UM APLICATIVO USANDO J2ME E PHP PARA MELHORIA
DO PROCESSO DE DETERMINAÇÃO DE TEMPO-PADRÃO
ATRAVÉS DE CRONOMETRAGENS**

**PATOS – PB
2011**

EDNALDO NUNES DE SOUZA SEGUNDO

**UM APLICATIVO USANDO J2ME E PHP PARA MELHORIA
DO PROCESSO DE DETERMINAÇÃO DE TEMPO-PADRÃO
ATRAVÉS DE CRONOMETRAGENS**

Trabalho de Conclusão de Curso apresentado ao Curso de Licenciatura em Computação da Universidade Estadual da Paraíba, em cumprimento à exigência para obtenção do grau de Licenciado em Computação.

Orientador (a): MSc. Ana Carolina Costa de Oliveira

PATOS – PB
2011

S719a SOUZA SEGUNDO, Ednaldo Nunes de

Um aplicativo usando J2ME e PHP para melhoria do processo de determinação de tempo-padrão através de cronometragens. /Ednaldo Nunes de Souza Segundo. – Patos: UEPB, 2011.
20f

- Artigo (trabalho de conclusão de curso -
(Tcc) - Universidade Estadual da Paraíba.
Orientador: Prof^ª. Msc^ª. Ana Carolina Costa de Oliveira

1.Computação 2.Desenvolvimento de software I.
Titulo II.Oliveira, Ana Carolina Costa de

CDD 629.895

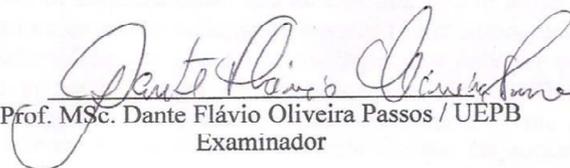
EDNALDO NUNES DE SOUZA SEGUNDO

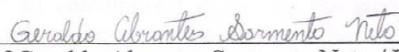
**UM APLICATIVO USANDO J2ME E PHP PARA MELHORIA
DO PROCESSO DE DETERMINAÇÃO DE TEMPO-PADRÃO
ATRAVÉS DE CRONOMETRAGENS**

Trabalho de Conclusão de Curso apresentado
ao Curso de Licenciatura em Computação da
Universidade Estadual da Paraíba, em
cumprimento à exigência para obtenção do
grau de Licenciado em Computação.

Aprovada em 7 de junho de 2011.


Profª MSc Ana Carolina Costa de Oliveira / UEPB
Orientadora


Prof. MSc. Dante Flávio Oliveira Passos / UEPB
Examinador


Prof Geraldo Abrantes Sarmiento Neto / UEPB
Examinador

UM APLICATIVO USANDO J2ME E PHP PARA MELHORIA DO PROCESSO DE DETERMINAÇÃO DE TEMPO-PADRÃO ATRAVÉS DE CRONOMETRAGENS

NUNES, Ednaldo de Souza Segundo

RESUMO

O objetivo deste artigo é desenvolver um aplicativo usando J2ME e PHP para melhoria do processo de determinação de tempo-padrão através de cronometragens. A ideia desse sistema nasceu em consideração à atividade do cronometrista ser bastante difícil tendo em vista as tecnologias empregadas tais como: cronômetro, prancheta, formulário e muita concentração. O uso de novas tecnologias não só para computadores, mas, também para dispositivos móveis auxiliam na eficiência desse processo de cronometragem. O artigo apresenta caráter tecnológico ou aplicado – prático. Para o desenvolvimento do sistema realizou-se uma pesquisa bibliográfica para o embasamento sobre as tecnologias, os conceitos relacionados a estudo de tempos, e a sua execução através de cronometragens. Como resultado, foi desenvolvido um aplicativo para dispositivos móveis na plataforma J2ME, capaz de realizar a coleta e o envio de informações pertinentes ao estudo de tempo através de cronometragens para uma central de processamento (servidor de acesso) em que *scripts* na linguagem PHP realizam todos os cálculos que envolvem o estudo em questão e a geração de relatórios para consultas. A sintonia entre essas duas tecnologias é que torna o sistema prático e eficiente para o estudo de tempos.

PALAVRAS-CHAVE: J2ME, PHP, Estudo de Tempo.

ABSTRACT

The purpose of this paper is to develop a application using J2ME and PHP to improve the process of determination of standard time. The idea of this system arose account activity of timer to be so difficult in view of the technologies applied like: chronometer, clipboard, form and a lot of concentration. The use of new technologies not only for computers, but also mobile devices assist in the efficiency of this process of timing. The paper presents a technological nature or applied - practical. For the development of the system a literature search was conducted for the basement about the technologies, the concepts in the study of time and its implementation through timings. As a result was developed an application for mobile devices in the J2ME platform, able to perform the collection and sending of relevant information to the study of time through timings to a central processing (access server), where scripts in the PHP realize all calculations involving the study in question and generating reports for queries. The harmony between these two technologies is what makes the system practical and efficient for the study or time.

KEYWORDS: J2ME, PHP, Time Study.

1 INTRODUÇÃO

O aumento do número de dispositivos móveis nos últimos anos, em especial dos telefones celulares, abriu um universo de possibilidades que deram início aos desenvolvedores de aplicações para tais equipamentos (CREZI *et al*, 2008). De acordo com a Agência Nacional de Telecomunicações (ANATEL, 2011) o Brasil encerrou o ano de 2010 com 202,94 milhões de novas habilitações de acessos móveis no país (ANATEL, 2011).

De acordo com essa afirmação é crescente o número de aplicações que utilizam os recursos oferecidos por tais tecnologias. Tal fator aliado a expansão da Internet criou o ambiente ideal para implementar os mais diversos aplicativos que atendam as necessidades crescentes por comunicação em tempo real.

Segundo CREZI *et al* (2008), a demanda por aplicações móveis impulsionou o surgimento de novas tecnologias, como o *JAVA 2 Platform Micro Edition* (J2ME) o qual se destaca como uma das principais plataformas de desenvolvimento de aplicações para pequenos dispositivos presentes em vários processos como o produtivo, agropecuário, comercial e outros.

Para Pinto (2011), o conhecimento a respeito da capacidade de produção de uma empresa, ou de um processo específico, possibilita a gestão adequada dos recursos produtivos. Ainda de acordo com o autor, tal gerenciamento está ligado à utilização de novas tecnologias, bem como através da criação de mecanismos de controle, ou ainda, por meio da existência de pontos de partida a serem utilizados como parâmetros como, por exemplo, a aplicação de dispositivos móveis no sistema produtivo.

O estabelecimento do valor relativo à capacidade instalada em uma organização depende, entre outros fatores, do tipo de tecnologia empregada na produção, pois desta deriva a forma mais adequada de mensuração da capacidade de uma operação específica, seja através da capacidade das máquinas, do histórico de produção, de cronometragens, da utilização de tempos pré-determinados, da análise de filmes, entre outras (RESENDE *et al*, 2009).

Dentre as referidas técnicas, a cronometragem se destaca por ser uma das mais difundidas nas organizações, principalmente no que tange a processos nas quais são utilizadas atividades manuais. Apesar desta disseminação da técnica, e de sua precisão, verifica-se que a atividade do cronometrista é muito dificultada pelos tipos de tecnologia empregados; cronômetro, prancheta e formulário; bem como pelo alto nível de concentração exigido na tarefa (PINTO, 2011). Sendo assim, o presente estudo tem o objetivo de desenvolver um aplicativo usando J2ME e PHP para melhoria do processo de determinação de tempo-padrão através de cronometragens. A junção de tecnologias que darão maior dinamicidade ao processo tanto na coleta dos dados quanto no cálculo dos tempos.

2 REFERENCIAL TEÓRICO

Em seguida serão discutidos alguns tópicos relevantes para a consolidação do objetivo abordado neste artigo, destacando os conceitos sobre Java para dispositivos móveis, PHP e alguns aspectos ligados ao estudo de tempos.

2.1 Java para celulares (J2ME)

A linguagem Java foi criada com o conceito do *write once, run anywhere* (“escreva uma vez e execute em qualquer lugar”), cuja ideia é escrever e compilar o código apenas uma vez, a partir de então, executá-lo em qualquer dispositivo que suporte uma máquina virtual Java (MATTOS, 2005). A linguagem Java tem uma vasta biblioteca, o que torna possível

escrever programas portáteis independente de sistemas operacionais proprietários (HORSTMANN, 2004).

De acordo com Mattos (2005, p. 16), para que a plataforma Java atenda a variados tipos de dispositivos foi realizado o agrupamento das tecnologias:

- *Java 2 Platform, Standard Edition* (J2SE) – projetada para máquinas simples e estações de trabalho;
- *Java 2 Platform, Enterprise Edition* (J2EE) – específica para aplicações baseadas em servidores;
- *Java 2 Platform, Micro Edition* (J2ME) – destinada a dispositivos móveis com limites de memória, vídeo e processamento.

Nesta pesquisa será utilizado o *Java 2 Platform, Micro Edition* (J2ME) que em conformidade com Rabello (2006), pode ser determinado como uma especificação da plataforma Java que foi planejada para aparelhos móveis sem fio, como PDAs, *paggers* e telefones celulares.

Na Seção 2.11 será descrito o processo de configuração do J2ME.

2.1.1 Configurações do J2ME

Para *Sun Microsystems* (2002), a configuração do J2ME possui dispositivos que restringe e limita os recursos, como poder de processamento, de memória e de capacidade gráfica. Para Kim (2002), o J2ME suporta dois tipos de configurações, denominadas CLDC (*Connected Limited Device Configuration* – Configuração para Dispositivos Limitados e Conectados) e CDC (*Connected Device Configuration* – Configuração para Dispositivos Conectados).

Ainda segundo o autor, a CLDC se referencia aos dispositivos como telefones celulares e *paggers*, que possuam memória em torno de 512 KB. Na literatura existem algumas características que fazem com que dispositivos pertençam à configuração do J2ME.

- O dispositivo deve ter um processador de 16 ou de 32 bits com velocidade de *clock* igual ou superior a 16 MHz;
- No mínimo de 160 KB de memória não volátil para a máquina virtual e as bibliotecas.

Enquanto a configuração CDC é usada em dispositivos mais robustos com maior poder de processamento como PDAs e TV *set-top boxes*, por isso, essa configuração é pouco difundida já que os valores desses dispositivos são mais elevados (MATTOS, 2005).

Na Seção 2.1.2 será apresentado o conjunto de padrões APIs para uma melhor compreensão das bibliotecas.

2.1.2 Perfis do J2ME

Um perfil do J2ME é um conjunto de padrões de APIs (*Application Programming Interface*) que oferece suporte para uma gama de dispositivos que são construídos sobre uma determinada configuração (SUN MICROSYSTEMS, 2002). A junção entre um perfil e uma configuração define uma aplicação em J2ME.

Para Mattos (2005), o MIDP (*Mobile Information Device Profile*) é o perfil agregado a CLDC que proporciona os seguintes benefícios:

- suporte a interface gráfica com o usuário (GUI) com tamanho mínimo de 96 x 54 *pixels*;
- mecanismos de entrada de dados seja teclado ou telas de toque (*touch*);
- conectividade – suporte ao desenvolvimento de aplicações cliente/servidor com os protocolos HTTP, HTTPS e *sockets*.

Na próxima seção, serão discutidos MIDlets a partir no modelo CLDC.

2.1.3 MIDlets

Uma MIDlet é o coração de uma execução no modelo CLDC, ou seja, é a aplicação em Java para dispositivos móveis em que o usuário entra com os dados e a MIDlet retorna os resultados. Segundo Rischpater (2008), as MIDlet são classes que interagem com o sistema respondendo a requisições e até interrompendo a aplicação.

Para Cardoso (2007) uma MIDlet tem um ciclo de vida muito semelhante ao de programas Java feitos para navegadores *web*, os chamados *applets*, tendo três estágios possíveis:

- **Ativo** – inicializa a MIDlet e ativa a mesma;
- **Pausa** – estado de pausa onde a aplicação pode suspender sua execução ou operar alguma rotina específica durante um evento externo como, por exemplo, uma chamada telefônica;
- **Destruição** – destruição da aplicação.

Na sequência, apresenta-se uma breve discussão sobre linguagem de programação PHP (*Hypertext Preprocessor*).

2.2 PHP (*Hypertext Preprocessor*)

De acordo com CREZI *et al*, (2008) O PHP é uma linguagem para *scripts* licenciada como *software* livre, ou seja, são *softwares* cuja licença permite a qualquer um copiar, usar e distribuir, com ou sem modificações, gratuitamente ou não. No entanto, isto significa que o código fonte deve estar disponível.

Ainda de acordo com os autores a sigla é um acrônimo recursivo para PHP *Hypertext Preprocessor*. A linguagem foi, originalmente, chamada de *Personal Home Page Tools*, mas, como se expandiu em escopo, um novo nome foi sugerido pela comunidade de desenvolvedores.

Pode-se pensar nessa linguagem como uma coleção de *tags* de HTML que permitem montar dinamicamente uma página da *web*. O código tem pouca relação com a aparência de uma página da *internet*, pois a maior parte do que o *Hypertext Preprocessor* realiza é invisível para o usuário final. Isso porque o servidor interpreta o código e exibe o resultado no *browser* no formato de uma página HTML.

Segundo Moraz (2005) o PHP é facilmente encontrado em praticamente todos os sistemas operacionais como *Linux* e *Microsoft Windows*. Essa linguagem é reconhecida pela maioria dos servidores *web* atual como o *Apache* e o *Internet Information Service* (IIS).

Essa tecnologia também oferece algumas vantagens como: alto desempenho; interface para muitos sistemas de banco de dados; bibliotecas integradas; baixo custo; facilidade de aprender e utilizar e ótimo suporte a orientação a objetos em relação a outras linguagens (THOMSON; WELLING, 2005).

Na Seção 2.3 será realizada uma breve revisão na literatura sobre os conceitos de estudos de tempo por ser o objeto do aplicativo.

2.3 Estudo de tempos

O estudo de tempos foi iniciado por Frederick Taylor no final do século XIX, neste mesmo período o casal Gilbert desenvolveu o estudo de movimentos, o qual analisa a melhoria de métodos de trabalho (BARNES, 1977).

O estudo de tempos, na percepção de Barnes (1977), é uma metodologia que determina o tempo necessário para uma pessoa qualificada e treinada, trabalhando em ritmo

normal, executar uma tarefa específica. Outros autores perpetuam dessa definição, é o caso de Silvéria (2005), Oss (2007), Suarez (2007) e Slack *et al* (2002).

Os resultados do estudo de tempo são mostrados através do “próprio tempo, em minutos, que uma pessoa adaptada ao trabalho e completamente treinada no método específico levará para executar a tarefa trabalhando em um ritmo considerado normal”. Este tempo é conhecido como sendo o tempo padrão da operação (BARNES, 1977).

O estudo de tempos tem várias finalidades em um sistema de produção, conforme indica Barnes (1977, p. 273) e Oss (2007, p. 41):

- Estabelecer programações e planejar o trabalho;
- Determinar os custos-padrão e auxiliar no preparo de orçamentos;
- Estimar o custo de um produto antes do início da fabricação;
- Determinar a eficiência de máquinas, o número de máquinas que uma pessoa pode operar o número de homens necessários ao funcionamento de um grupo, e como um auxílio ao balanceamento de linhas de montagem e de trabalho controlado por transportadores;
- Determinar tempos-padrão a serem usados como base para o pagamento de incentivo à mão-de-obra direta;
- Determinar tempos-padrão a serem usados como base para o pagamento de mão-de-obra indireta, tais como os movimentadores de materiais e os preparadores;
- Determinar tempos-padrão a serem usados como base do controle de custo da mão-de-obra.

2.3.1 Execução do estudo de tempos (cronometragem)

Para iniciar a execução do estudo de tempos Oss (2007) apresenta os procedimentos a serem tomados, iniciando pela informação e conhecimento dos funcionários quanto ao estudo e seus objetivos. Antes de iniciar o trabalho, é fundamental avaliar se a operação está suficientemente, preparada para um estudo de tempos, para isso, deve-se observar se pode ocorrer redução do tempo de ciclo, alteração das ferramentas, e se é possível aproximar os materiais da área de trabalho a fim de reduzir o tempo de manuseio.

Os procedimentos para análise do estudo de tempos irão depender do tipo de operação ou da forma de execução do estudo:

De acordo com Barnes (1977, p. 277), oito passos são fundamentais.

1. Obtenha e registre informações sobre a operação e o operador em estudo;
2. Divida a operação em elementos e registre uma descrição completa do método;
3. Observe e registre o tempo gasto pelo operador;
4. Determine o número de ciclos a ser cronometrado;
5. Avalie o ritmo do operador;
6. Verifique se foi cronometrado um número suficiente de ciclos;
7. Determine as tolerâncias;
8. Determine o tempo-padrão para a operação.

Do procedimento, apresentado por Barnes (1977), para a realização do estudo de tempos, a avaliação de ritmo do operador é a etapa que constitui o objeto de interesse deste artigo, portanto será apresentada uma revisão sobre a literatura deste item.

2.3.2 Avaliação de ritmo

De acordo com Barnes (1977) e Suarez (2007), avaliação de ritmo é o processo durante o qual o analista de tempos compara a rapidez e precisão do operador em observação com o seu próprio conceito de ritmo normal.

Segundo Slack *et al* (2002, p. 288) avaliação de ritmo é denominada como desempenho padrão no qual “a taxa de saída que é atingida por trabalhadores qualificados sem esforço excessivo na media do dia de trabalho, desde que estejam motivados a aplicar-se em seu trabalho”.

Na percepção de Suarez (2007, p.21) o ritmo normal “é o ritmo de trabalho de um operador normal, que pode ser mantido dia após dia, sem fadiga mental ou física excessiva e, é caracterizado pelo exercício quase ininterrupto e de esforço razoável”.

A fase de avaliação de ritmo é considerada a mais importante e difícil do estudo de tempos, pois consiste na avaliação do ritmo do trabalhador através de um avaliador que analisar o trabalho de um operário no momento da coleta de dados.

Para Suarez (2007) o objetivo da avaliação é o de estabelecer o tempo que possa representar a eficiência normal, ou seja, pode-se definir a avaliação como sendo um meio de calcular o tempo correspondente a um operador com habilidade normal, trabalhando com esforço habitual. Por isso não há uma maneira de estabelecer o tempo padrão para uma atividade sem termos o julgamento do avaliador de estudos de tempo. No entanto o avaliador de tempos é um funcionário treinado que possui sensibilidade de avaliar as variações no desempenho dos funcionários no momento que os dados estão sendo coletados.

Entretanto, o avaliador (cronometrista) por meio de treinamento específico, para poder comparar, julgar e avaliar, segundo padrões de desempenho, definidos e determinados, o ritmo do operador em observação durante um estudo de tempos (SUAREZ, 2007).

Para Barnes (1977), existem alguns sistemas utilizados para a avaliação de ritmo que são: avaliação do ritmo através da habilidade e esforço; sistema *westinghouse* para avaliação de ritmo; avaliação sintética do ritmo; avaliação objetiva do ritmo; avaliação fisiológica do nível de execução; e desempenho do ritmo.

De acordo com Barnes (1977), estabelecer um padrão para avaliar o ritmo é considerado uma medida válida. E necessário a definição de qual será o normal ou o padrão.

2.3.3 Determinação do tempo normal

O tempo normal é definido por Moreira (2000), como sendo o tempo necessário para um trabalhador concluir a sua tarefa operando com velocidade normal, ou seja, executando suas atividades de forma eficiente no decorrer de um dia típico de trabalho sem fadiga indevida. Slack *et al* (2002), denominam tempo normal como tempo básico, o qual é traduzido como o tempo levado por um trabalhador qualificado que faz um trabalho específico com desempenho padrão. Dessa forma, pode-se observar as definições apresentadas são semelhantes e enfatizam o conceito.

Para determinar um tempo normal de acordo com Barnes (1977) é necessário que um trabalhador qualificado execute uma determinada operação levando em conta a velocidade que o operário realiza a operação.

TN = TS * R / 100	TN = Tempo normal TS = Tempo selecionado R = Ritmo porcentual
--------------------------	---

QUADRO 1 - Formula do tempo normal. Fonte: Barnes (1977).

Após a definição do tempo normal, acrescenta-se a tolerância para se obter o tempo padrão. O tempo padrão, para Barnes (1977, p. 313), é a “duração de todos os elementos da operação e, além disso, deve incluir o tempo para todas as tolerâncias necessárias”.

Em uma definição mais precisa, Suarez (2007) indica que o tempo padrão é a quantidade de tempo necessário para executar todas as tarefas de uma operação, em condições

normais, de acordo com o método estabelecido por um avaliador qualificado e, além disso, os tempos para todas as tolerâncias necessárias. O tempo padrão é igual ao tempo normal mais as tolerâncias. Outros autores também adotam essa definição. E o caso de Slack *et al* (2002) e Moreira (2000).

De acordo com Slack *et al* (2002), tolerância são concessões acrescentadas ao tempo normal para permitir descanso, relaxamento e necessidades pessoais, ou seja, o tempo padrão pode ser representado como:

TP = TN * FT	TP = Tempo padrão TN = Tempo normal FT = Fator de tolerância
---------------------	--

QUADRO 2 - Formula do tempo padrão. Fonte: Barnes (1977).

A seguir apresentam-se os procedimentos metodológicos que subsidiaram a pesquisa.

3 REFERENCIAL METODOLÓGICO

Delimitou-se como o objetivo geral do artigo a seguinte construção: desenvolver um aplicativo usando J2ME e PHP para melhoria do processo de determinação de tempo-padrão através de cronometragens. Etapas complementares ao alcance do objetivo geral do trabalho foram desenvolvidas na forma de objetivos específicos: criar um aplicativo usando J2ME para coleta e armazenamento dos tempos; enviar as informações para o servidor de acesso; realizar os estudos dos tempos; e Imprimir os dados.

De acordo com Rummel (1997, p. 3) esta pesquisa foi classificada como uma pesquisa tecnológica ou aplicada – prática por objetivar a aplicação dos tipos de pesquisa relacionados as necessidades imediatas dos diferentes campos da atividade humana. Primeiramente, realizou-se uma pesquisa bibliográfica para obter o conhecimento necessário para desenvolver uma contextualização, argumentos e observações, assegurando a qualidade das informações.

Foi desenvolvido um programa na linguagem Java para dispositivos móveis que contém a função básica de cronometragem e armazenamento dos ciclos de tempo do operador em questão. Com os ciclos de tempo dos operadores armazenados no dispositivo móvel foi desenvolvido um código fonte que tinha a função de enviar as informações coletadas através de uma conexão sem fio por meio do dispositivo móvel ao servidor.

Na sequência, foram tratadas as informações coletadas no dispositivo móvel por outro aplicativo desenvolvido na linguagem PHP que tinha como objetivo efetuar os cálculos necessários para o estudo dos tempos. E finalmente foram criados os relatórios que tinha como finalidade a apresentação dos dados coletados. Na sequência, apresentam-se os resultados alcançados de acordo com o objetivo estabelecido.

4 DADOS E ANÁLISE DA PESQUISA

Na sequência, apresentam-se os resultados alcançados de acordo com o objetivo geral estabelecido no corpo deste artigo. A estrutura desta seção discute os desdobramentos dos objetivos específicos que foram delimitados na Seção 3, abordando também observações relevantes para atingir a finalidade da pesquisa.

4.1 CRIAÇÃO DO APLICATIVO USANDO J2ME

Para realizar a coleta dos tempos foi necessário um dispositivo que fornecesse outras funcionalidades além da contagem de tempos, ou seja, o que acontece no cronômetro. Para tanto, foi desenvolvido um aplicativo capaz não só de medir tempos, mas também, armazenar e transferir outras informações pertinentes ao processo de cronometragem.

O aplicativo foi desenvolvido sob a tecnologia J2ME para celulares tendo em vista todas as funcionalidades que essa linguagem pode oferecer visto no referencial teórico deste artigo.

A MIDlet, principal classe do aplicativo, chamada de *StudyTime*, controla todas as demais classes que compõem o programa e é responsável pela mudança das telas no visor do dispositivo que pode ser entre listas, formulários e o cronômetro. A estrutura do arquivo *StudyTime.java* que contém a MIDlet e seus principais métodos: *startApp()*, *pauseApp()* e *destroyApp()* pode ser descrito abaixo.

```

1  classe StudyTime estende MIDlet
2      início da classe
3      método startApp()
4          início do método
5              // iniciar o programa no dispositivo
6          fim do método startApp
7      método pauseApp()
8          início do método
9              // tratamento de interrupções
10     fim do método pauseApp
11     método destroyApp()
12         início do método
13             // encerrar o programa no dispositivo
14         fim do método destroyApp
15 fim da classe StudyTime

```

Assim que o aplicativo é iniciado através o método *startApp()*, a primeira tarefa será mostrar uma lista com duas opções iniciais que são: nova cronometragem e sair do aplicativo. Se o usuário escolher em sair do aplicativo então o método da MIDlet *StudyTime destroyApp()* é invocado e a aplicação será encerrada, se o usuário escolher uma nova cronometragem então segue uma outra lista contendo os nomes dos operários já cadastrados e uma opção para cadastrar um novo operário, os nomes dos operários são obtidos diretamente no servidor para evitar inconsistência de dados entre os sistemas do dispositivo móvel e do servidor. A conexão com o servidor é realizada através da classe *conexao*, ela é responsável por enviar e receber as informações junto ao servidor. A referida classe será discutida no próximo tópico.

Para popular a lista com os nomes dos operários cadastrados foi criado um método chamado *preencheListaOperarios()*. Esse método recebe a quantidade de nomes a inserir na lista e através de um comando de *loop* começa a colocar o primeiro até o último nome. Abaixo segue o método.

```

1  método preencheListaOperarios (nomes)
2  início do método preencheListaOperarios
3      listaOpe <- nova Lista(); // cria uma nova lista
4      Para i de 0 até nomes.tamanho - 1 faça
5          início Para

```

```

6         listaOpe <- nomes[i];
7         fim Para
8     fim do método preencheListaOperarios

```

A Figura 01 mostra a lista depois de preenchida através do método *preencheListaOperarios()*.

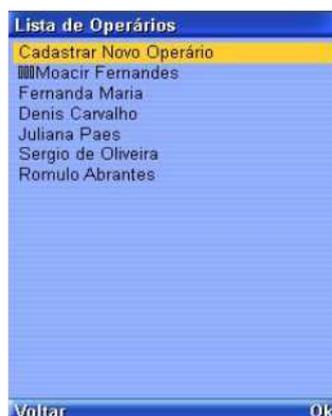


FIGURA 01 - Lista dos operários com a opção de cadastrar um novo operário.

Se o usuário optar em cadastrar um novo operário então a MIDlet cria o formulário para cadastro de novos operários através do método *getCadOperarios()* descrito abaixo.

```

1 método getCadOperarios ()
2     início do método getCadOperarios
3         CampoTexto campoNome <- novo CampoTexto ("Nome do
4         Operário");
5         cadOperarios <- novo Formulário (campoNome);
6     fim do método getCadOperarios

```

Depois do operário cadastrado, a lista com os outros nomes é atualizada de imediato dando uma maior velocidade ao trabalho. A Figura 02 exemplifica bem esse processo.



FIGURA 02 - Processo de cadastro de operários e atualização da lista.

Antes do início da cronometragem, o operador deverá realizar um novo cadastramento contendo outras informações que completam o processo de estudo como: nome do produto, o modelo, departamento, a operação, nome do cronometrista, fábrica e a data da coleta, para isso, a MIDlet cria mais um formulário para realizar esses cadastro através do método

getCompletarCadastro(). Esse método é bem semelhante ao *getCadOperarios()* onde, o que os diferencia, são os campos de textos utilizados para a digitação das informações. A Figura 03 demonstra o resultado do método *getCompletarCadastro()* e os campos de textos já preenchidos.

FIGURA 03 - Dados complementares do estudo

Depois dos dados cadastrados é dado início a cronometragem. Nessa parte do aplicativo, o cronômetro recebe os comandos do operador para iniciar a contagem do tempo.

Para o desenvolvimento do cronômetro foi utilizado duas classes: a *StopWatchThread* responsável literalmente pela contagem dos tempos e a *StopWatchCanvas* responsável em jogar na tela do dispositivo os dados gerados pela *StopWatchThread*.

A classe *StopWatchThread*, possui os métodos que caracterizam um cronômetro comum como: *startWatch()* responsável em iniciar o cronômetro; *pauseWatch()* possui a função de pausar a contagem do tempo e o *resetWatch()* que zera a cronometragem. A atualização dos tempos é dada através do método *run()* descrito em seguida.

```

1 método run ()
2 início do método run
3 enquanto o programa estiver rodando então
4 início enquanto
5 se pausado então
6 início se
7     setarCor(branco);
8     desenharRetangulo();
9     setarCor(azul);
10    desenharRetangulo();
11    setarCor(preto);
12    desenhaTexto(tempo);
13 senão
14     sincroniza();
15     espera();
16 fim se
17 fim enquanto
18 fim do método run

```

A classe *StopWatchCanvas* tem a função de mostrar na tela os dados da classe *StopWatchThread* e para isso ela possui o método *paint()*.

```

1 Método paint ()

```

```

2 Início do método paint
3 // Desenho do Fundo
4 setarCor(branco);
5 desenharRetangulo(); // retângulo com o tamanho da tela
6 setarCor(azul);
7 desenharRetangulo();
8 desenhaTexto(tempoDoCronometro);
9 fim do método paint

```

O cronometrista pode a qualquer momento registrar os lapsos de tempo conforme for necessário através do botão “Lapso”.

Cada espaço de tempo é registrado em uma lista para depois ser anexada a todas as outras informações antes informadas. O cronometrista pode visualizar essa lista bastando apenas pausar a cronometragem e depois apertar o botão “Visualizar Lista”. A Figura 04 ilustra esse processo.

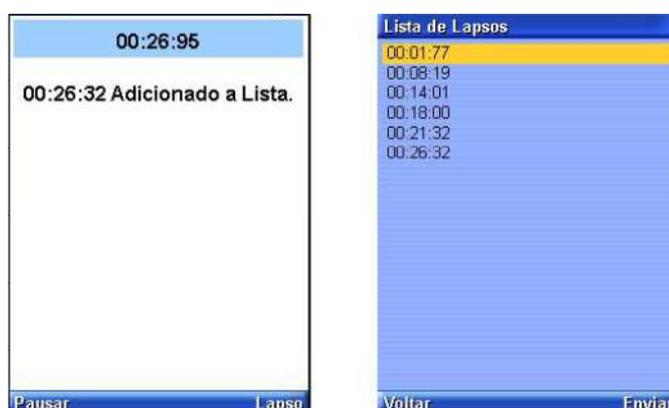


FIGURA 04 - Cronômetro e lista de tempos armazenados

Com a cronometragem finalizada, os dados estão prontos para serem enviados, mas antes disso, o operador deverá informar dois dados que são importantes para o estudo dos tempos que são, respectivamente, o ritmo de trabalho e o fator de tolerância, para isso, a MIDlet cria mais um formulário através do método *getRitmo()*. Como esse é o último passo de todo o processo da cronometragem, esse formulário tem a função de enviar para o servidor todos dados guardados no aparelho até agora. A Figura 05 ilustra esse formulário.

FIGURA 05 - Ritmo e Fator de Tolerância

4.2 ENVIANDO AS INFORMAÇÕES PARA O SERVIDOR DE ACESSO

O próximo passo, logo após a cronometragem, é o envio das informações para o servidor de acesso.

O aplicativo em J2ME possui uma classe chamada *conexao* unicamente para o envio e o recebimento de dados.

A classe *conexao* trabalha com mais duas classes nativas da linguagem Java específicas para redes que são: *HttpConnection*, contida no pacote *javax.microedition.io*; e *InputStream*, contida no pacote *java.io* herdada do J2EE.

Para tratar das inúmeras requisições feitas ao servidor realizadas pela aplicação, seja para cadastros ou buscas, a referida classe possui o método *requisicao(String url)* em que possui como parâmetro um texto contendo o endereço do servidor na rede e os parâmetros de envio das informações. A classe *conexao* junto com seu método *requisicao(String url)* estão dispostos a seguir:

```

1  classes Conexao
2      início da classe Conexao
3      método requisicao(url)
4      início do método requisicao
5          abrirConexao(url);
6          // verifica se a conexão foi feita com sucesso
7          se (conexão com sucesso) então
8              início se
9                  retorna dadosServidor();
10             senão
11                 retorna "Não conectado";
12             fim se
13         fim do método requisicao
14 fim da classe conexao

```

4.3 OS ESTUDOS DOS TEMPOS

Até o presente momento, todas as operações referentes à coleta dos tempos de responsabilidade da aplicação móvel já foi realizada, agora o próximo passo do processo é tratar as informações enviadas pelo dispositivo móvel e, para isso, no servidor é executado mais uma aplicação para realizar essas operações através de *scripts* em PHP.

Os *scripts* terão a tarefa de calcular os dados recebidos e mostrar os resultados no navegador de *Internet*, para tanto, foram criados três *scripts* para realizar esses cálculos que são: *index.php*, é a página inicial em que o usuário terá acesso aos estudos e executar a observação dos mesmos; *Tempo.php*, classe em PHP que possui métodos para a execução dos cálculos e retorno dos resultados para a página *index.php* e, por último, a *tempoPrinter.php* que é mais uma página para relatórios com padrões de configuração voltados para a impressão dos resultados.

A página *index.php* é a porta de entrada para que o usuário possa observar os estudos cadastrados, é a interface gráfica da classe *Tempo*. Nessa página, existem rotinas para organização do *menu* e os comandos para a execução dos cálculos na referida classe. Em seguida é apresentado um *script* que mostra os nomes dos operários em que já foram realizados estudos retornando *links* com comandos para a classe *Tempo*.

```

1 <?php
2     $busca = mysql_query("SELECT o.Nome, c.IdOpe FROM
operarios AS o, cronometragens AS c WHERE c.IdOpe =
o.Id");
3     if (mysql_num_fields($busca) == 0) {
4         echo "<p id='resposta'>Nenhum estudo
cadastrado</p>";
5     } else {
6         echo "<ul>";
7         while ($linha = mysql_fetch_array($busca)) {
8             echo "<li><a href='index.php?op=1&operario=" .
$linha['IdOpe'] . "'> . $linha['Nome'] . "</a></li>";
9         }
10        echo "</ul>";
11    }
12 ?>

```

A partir do momento em que o usuário clicar no *link* do nome do operário o *script* *index.php* instancia a classe *Tempo* e a mesma retorna todos os cálculos já realizados. A seguir o script de instanciação da classe *Tempo* e outro que lista os tempos desse operário.

Instanciação:

```

1 <?php
2     $tempos = new Tempo();
3 ?>

```

Lista dos tempos:

```

1 <?php
2     if (isset($_GET['op']) && $_GET['op'] == 1) {
3         $outraLinha = explode("|", $re);
4         echo "<ul id='lista'>";
5         for ($c = 0; $c <= sizeof($outraLinha) - 2; $c++) {
6             echo "<li>" . $outraLinha[$c] . "</li>";
7         }
8         echo "</ul>";
9     }
10 ?>

```

Na Figura 06 é demonstrada a página inicial do sistema em que o usuário pode escolher o operário e automaticamente a mesma irá mostrar os cálculos devidos conforme os dados coletados pelo cronometrista.

FIGURA 06 - Página inicial para a escolha do estudo a analisar

A classe *Tempo* possui métodos que calculam os itens do estudo de tempo como tempo total, número de observações, média, tempo normal e o tempo padrão.

Para realizar o cálculo do tempo total foi atribuído ao método *tempoTotal()* em que simplesmente ocorre a soma dos tempos cadastrados pelo dispositivo móvel, para isso, foi necessário uma função extra chamada *converteMinutoSegundo(\$t)* para converter minutos em segundos para que a soma seja realizada. A seguir o método *tempoTotal()*.

Tempo total:

```

1 // Soma todos os tempos
2 public function tempoTotal($idCrono) {
3     $busca = mysql_query("SELECT Tempo FROM tempos WHERE
4     IdCrono = " . $idCrono);
5     $t = NULL;
6     while ($linha = mysql_fetch_array($busca)) {
7         $t = $t . $linha['Tempo'] . "|";
8     }
9     $times = explode("|", $t);
10    $somatorio = NULL;
11    for ($i = 0; $i <= sizeof($times) - 2; $i++) {
12        $somatorio = $somatorio + $this->
13        converteMinutoSegundo($times[$i]);
14    }
15    return $somatorio;
16 }

```

O método *numObs()* retorna o número de lapsos de tempo registrados pelo cronometrista na hora da cronometragem. Esse método possui como parâmetro o código do estudo de tempo gravado no momento do envio das informações pelo dispositivo móvel.

```

1 public function numObs($idCrono) {
2     $busca = mysql_query("SELECT Tempo FROM tempos WHERE
3     IdCrono = " . $idCrono);
4     return mysql_num_rows($busca);
5 }

```

O cálculo da média é, simplesmente, o tempo total dividido pelo número de observações, sendo assim, o retorno do método *tempoTotal()* dividido pelo retorno do método *numObs()*.

4.4 CRIAÇÃO DOS RELATÓRIOS

A última parte do sistema é a geração do relatório final com todos os dados já calculados e dispostos de forma que o usuário possa imprimir.

Para a realização deste relatório foi necessário uma ligação proveniente da página inicial, para isso, tem-se na mesma um botão “Imprimir” em que o mesmo leva para a página de relatório para simples conferência.

A Figura 07 mostra com detalhes a página de relatórios.

The screenshot shows a web interface titled "Study Time - Relatório para Simples Conferência". The form contains the following data:

Nome do Operário		Data	
Roberto Carlos		28/05/2011	
Produto	Modelo	Departamento	
Carteira Masculina	De Couro	Produção	
Fábrica	Nome do Cronometrista		
Mitty	Maria Aparecida		
Tempo Total	Nº de Obs.	Média	% de Ritmo
89.61 s	6	14.94 s	90.00%
Tempo Normal	Fator de Tolerância	Tempo Padrão	
13.44	1.00	13.44 s	

Below the summary table is a section titled "CRONOMETRAGENS" containing a list of time entries:

- 00:01:77
- 00:08:19
- 00:14:01
- 00:18:00
- 00:21:32
- 00:26:32

FIGURA 07 - Relatório final com todas as informações descritas.

5 CONCLUSÃO

A utilização de recursos computacionais como ferramenta de melhoria de vida vem se consolidando através do contínuo avanço das tecnologias e é através dessa perspectiva que a criação do aplicativo tema deste artigo traz velocidade e confiabilidade ao processo de estudo de tempo através de cronometragens.

Diante dos problemas encontrados pela atual forma de coleta das informações para a geração do estudo dos tempos, o aplicativo para dispositivos móveis é uma ótima opção para agilizar esse processo e de manter a consistência das informações sobre cada estudo. O

aplicativo trouxe uma rapidez ao processo, tendo em vista a opção de enviar as informações a qualquer momento para uma central de processamento (servidor).

Através do poder da plataforma J2ME foram acrescentadas outras funcionalidades ao cronômetro convencional como, por exemplo, a captura de informações complementares ao estudo, a exemplo do nome do operário, do produto, da fábrica e a operação realizada pelo mesmo.

Como o sistema trabalha através de uma rede, há a possibilidade de se adicionar quantos dispositivos quiser aumentando, ainda mais, a velocidade do estudo considerando a quantidade de operários a analisar.

A central de processamento, ambientada no modelo *web*, traz uma maior facilidade de uso por parte do usuário, tanto para o manuseio ou para a impressão dos relatórios, benefício alcançado pela popularização da *Internet*.

É notável a contribuição desse sistema para a melhoria do estudo de tempos ficando como avanços para o futuro a adição de novas ferramentas para aplicação móvel como, por exemplo, suporte a aparelhos com tela *touchscreen*.

REFERÊNCIAS

ANATEL. Anatel: Brasil encerra 2010 com 202,9 milhões de celulares, **Administradores**, João Pessoa, 19 jan. 2011. Disponível em <<http://www.administradores.com.br/informe-se/tecnologia/anatel-brasil-encerra-2010-com-202-9-milhoes-de-celulares/42119/>>. Acesso em: 23 abr. 2011.

BARNES, M. R. **Estudo de movimentos e de tempos: projeto e medida do trabalho**. 6 ed. São Paulo: Edgard Blücher, 1977.

CARDOSO, J. **Java para Telemóveis MIDP 2.0**. 2007. Disponível em: <<http://livromidp.jorgecardoso.eu/html/index.html>>. Acesso em: 22 fev. 2011.

CENZI, D.; TEIXEIRA, I.; COSTA, J. G.; DOBGENSKI, J. **Tecnologia de aplicações para dispositivos móveis integrada a banco de dados remoto utilizando software livre**. Anuário da Produção Científica dos cursos de Pós-graduação. Vol. III, N^o. 3, Ano, 2008.

HORSTMANN, Cay. **Big Java**. Trad. Edson Furmankiewiez. Porto Alegre. Ed. Bookman, 2004.

KIM, C., BURGUER, D., KECKLER, S. W. **An adaptive, non-uniform cache structure for wire-delay Dominated On-Chip Caches**, ACM ASPLOS X – 10th International Conference on Architectural Support for Programming Languages and Operational Systems, 2002.

MATTOS, E. T. **Programação java para wireless**. Digerati books, São Paulo, 2005.

MORAZ, Eduardo. **Treinamento avançado em PHP 5.0**. São Paulo, Digerati books, 2005.

MOREIRA, D. **Administração da produção e operação**. São Paulo: Pioneira, 2000.

OSS, L. F. **Aumento da capacidade e produtividade em uma empresa de pintura E-COAT**. 87.f. Monografia (Graduação em Engenharia de Produção) - Universidade de Caxias do Sul - Caxias do Sul, 2007.

- PINTO, M. G. **Utilização do workflow como alternativa para melhoria do processo de determinação de tempo-padrão através de cronometragens**, 2011. (Não publicado).
- RABELLO, R. R.; TRECCANI, P. J.; JOHNSON, T. M. **Integrando a Tecnologia J2ME no Âmbito Acadêmico**. Universidade da Amazônia, Belém, Pará. 2006.
- RESENDE, S. R.; OLIVEIRA, A. C. C.; SILVA, L. B.; GOMES, M. L. B. **Proposta metodológica para o cálculo do tempo normal**. In: SIMPÓSIO DE ENGENHARIA DE PRODUÇÃO, 13, 2008; SIMPEP, 13, 2008. Anais..., 2008.
- RISCHPATER, R. *Beginning java ME platform*. APress, 2008.
- RUMMEL, J. Francis. **Introdução aos procedimentos de pesquisa em educação**. 3. ed. Porto Alegre: Globo, 1997.
- SILVÉIRA, M. **Aplicação de ferramentas da produção na melhoria da produtividade: estudo de caso**. 60 f. Monografia (Graduação em Engenharia de Produção) - Universidade Estadual de Maringá, Paraná, 2005.
- SUAREZ, J. I. **Melhorias no estudo de tempos e métodos no setor de calça sarja**. 52 f. Monografia (Tecnólogo do Vestuário) - União de Ensino do Sudoeste do Paraná, Faculdade Educacional de Dois Vizinhos – FAED, Dois Vizinhos, 2007.
- SLACK, N.; CHAMBER, S.; HARDLAND, C.; HARRISON, A. e JOHNSTON, R. **Administração da produção**. São Paulo: Atlas, 2002.
- SUN MICROSYSTEMS. Mobile Information Device Profile (MIDP); JSR 37, JSR 118 Overview, *JavaWorld*, 13 dez. 2002. Disponível em <<http://www.javaworld.com/javaworld/jw-12-2002/jw-1213-j2medesign.html>> Acesso em: 22 fev. 2011.
- THOMSON, L.; WELLING, L. **“PHP e MySQL Desenvolvimento web”**, traduzido por FURMANKIEWICZ, Edson e Kramer, Adriana. 3ª edição. Elsevier, Rio de Janeiro, 2005.