

CENTRO PAULA SOUZA



**Faculdade de Tecnologia de Americana
Curso Superior de Tecnologia em Análise e Desenvolvimento de
Sistemas**

DESENVOLVIMENTO DE APLITIVOS PARA WINDOWS PHONE

MARCELO HÉBER KUHL

**Americana, SP
2013**

CENTRO PAULA SOUZA



**Faculdade de Tecnologia de Americana
Curso Superior de Tecnologia em Análise e Desenvolvimento de
Sistemas**

DESENVOLVIMENTO DE APLICATIVOS PARA WINDOWS PHONE

**MARCELO HÉBER KUHL
MARCELO_KUHL@HOTMAIL.COM**

Trabalho Monográfico, desenvolvido em cumprimento à exigência curricular do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas da Fatec-Americana, sob orientação do Prof. Wladimir da Costa

**Americana, SP
2013**

BANCA EXAMINADORA

**Professor: Wladimir da Costa
(Orientador)**

Professor: Rogerio Nunes De Freitas

Professor: Fernando José Ignácio

AGRADECIMENTOS

Agradeço primeiramente ao meu professor orientador pelo apoio que me deu e pela dedicação de me guiar pelo caminho correto.

Aos meus companheiros de faculdade pela amizade e ajuda nos trabalhos desenvolvidos.

À minha família pelo apoio e incentivo que sempre me deram para estudar.

RESUMO

Atualmente o celular não é somente um aparelho em que realizamos ligações, nós nos conectamos a internet, jogamos, acessamos diversos aplicativos e junto com esse avanço o mercado de aplicativos para celulares está crescendo muito e por essa razão o desenvolvimento de softwares pode se tornar uma grande fruto de lucro para os desenvolvedores. A empresa Microsoft lançou um sistema operacional para celular, o Windows phone com grandes inovações e novidades dando a oportunidade de um possível mercado de desenvolvimento de aplicativos em ascensão. O presente trabalho tem como objetivo esclarecer algumas dúvidas sobre o desenvolvimento de sistemas para celulares, mostrar ao leitor um pouco sobre o sistema Windows Phone e exemplificar a criação de um aplicativo.

Palavras Chave: Desenvolvimento; Celular; Windows Phone; Tecnologia; Mobile.

ABSTRACT

Currently the phone is not just a device that perform connections, we connect to the internet, we play, we access different applications and with this advance the market for mobile applications is growing tremendously and for this reason the development of software can become a large fruit of profit for developers. The company launched a Microsoft operating system Windows Mobile phone with great innovations and novelties giving the opportunity of a possible market for application development on the rise. This work aims to clarify some doubts about the development of systems for mobile phones, show the reader a little about the Windows Phone system and exemplify the creation of an application.

Keywords: Development; Mobile, Windows Phone; Technology; Mobile.

LISTA DE FIGURAS

Figura 1: Blocos dinâmicos.	16
Figura 2: Ferramenta de busca por imagem.	17
Figura 3: Arquitetura de desenvolvimento.	21
Figura 4: Ferramentas de instalação.	22
Figura 5: Aplicativo de registro de celular.	25
Figura 6: Página inicial do Visual Studio.	26
Figura 7: Componente Pivot.	28
Figura 8: Componente Panorama.	29
Figura 9: Arquitetura de desenvolvimento.	32
Figura 10: Diagrama de casos de uso.	33
Figura 11: Diagrama de classes.	34
Figura 12: Arquitetura do projeto.	35
Figura 13: Caixa de ferramentas e tela inicial.	36
Figura 14: Tela de Cadastro.	37
Figura 15: Tela de informações do sistema.	38
Figura 16: Navegação por propriedade.	39
Figura 17: Portal da internet.	51
Figura 18: Tela de última localização.	52
Figura 19: Tela de localização por data.	52
Figura 20: Tela de edição de dados do usuário.	53

LISTA DE TABELAS

Tabela 1: Tipos de contas.23

SUMÁRIO

1. INTRODUÇÃO.....	9
2. DESENVOLVIMENTO DE APLICATIVOS MÓVEIS	14
2.1 WINDOWS PHONE.....	14
2.1.1 GRANDES INOVAÇÕES DO SISTEMA: WINDOWS PHONE.....	16
3. MERCADO DOS DISPOSITIVOS	18
3.1 PESQUISA DE CAMPO	18
3.2 PORQUE DESENVOLVER PARA CELULARES?	19
4. AMBIENTE DE DESENVOLVIMENTO	21
4.1 REGISTRO DO DESENVOLVEDOR NA MARKETPLACE.....	23
4.2 FERRAMENTAS DE DESENVOLVIMENTO	26
4.2.1 VISUAL STUDIO.....	26
4.2.2 EMULADOR DO WINDOWS PHONE	27
4.3 PRINCIPAIS COMPONENTES	27
4.3.1 PIVOT	28
4.3.2 PANORAMA	28
4.4 CONCEITOS DA PROGRAMAÇÃO.....	29
5. ESTUDO DE CASO - DESENVOLVIMENTO DO APLICATIVO.....	31
5.1 ESPECIFICAÇÕES	31
5.2 INTERFACE	35
5.3 NAVEGAÇÃO.....	38
5.4 VALIDAÇÕES.....	40
5.5 SERVIÇOS DA WEB	42
5.6 POSICIONAMENTO GEOGRÁFICO.....	43
5.7 APLICAÇÃO EM SEGUNDO PLANO.....	45
5.8 PORTAL DA INTERNET	51
CONCLUSÃO	54
GLOSSÁRIO DE TERMOS.....	55

REFERÊNCIAS.....56

1. INTRODUÇÃO

Em primeiro lugar você já se imaginou sem um celular hoje em dia? Sem poder quebrar a barreira da distância para falar com aquele parente ou amigo seu que está longe? Ou até mesmo sem aquele aplicativo de rede social que você utiliza tanto para saber das novidades? Atualmente, podemos fazer tudo isso com o celular, mas você já parou para pensar na história de como foi inventado o telefone? Segundo PACIEVITCH (2013) “Depois de muitas tentativas, em 1876, o sonho de Alexander Graham Bell, se tornou possível. Através de um aparelho, entre um cômodo e outro, Watson ouviu Bell dizendo: “Sr. Watson, preciso do senhor, venha.” Nascia, assim, o telefone.” No Brasil segundo o PORTAL SÃO FRANCISCO (2013) em “1879 - 15 de novembro. Surgia, no Rio de Janeiro, o primeiro telefone, construído para D. Pedro II nas oficinas da Western and Brazilian Telegraph Company. Foi instalado no Palácio de São Cristóvão, na Quinta da Boa Vista, hoje, Museu Nacional.” Se paramos para pensar o telefone foi inventado há mais de cem anos e hoje permanece dentro de nossas casas e dificilmente conhecemos alguém que não possui um dentro de sua residência, no Brasil esta invenção veio após três anos, porém como dito foi feito especialmente para D. Pedro II até então rei do Brasil, e foi instalado no Museu, ou seja, as pessoas menos favorecidas não tiveram um contato imediato com tal invenção.

Assim como qualquer outro artefato inventado o telefone também foi se moldando as necessidades do usuário e hoje em dia temos o celular em nossas mãos e até mesmo em nossos bolsos, porém segundo o portal CUNHA (2013) “o telefone celular causou um enorme espanto há 40 anos, quando foi realizada a primeira ligação pública utilizando-se um aparelho portátil, no dia 3 de abril de 1973. Na ocasião, o pesquisador da Motorola Martin Cooper ligou para um telefone fixo diretamente de uma esquina do centro de Nova Iorque, nos Estados Unidos.” Ou seja, depois de mais de noventa anos da invenção do telefone foi criado o primeiro celular com o objetivo de dar uma mobilidade maior aos usuários. Segundo JORDÃO (2009) “Atualmente não se fala muito em celular, porque o assunto da vez são os *Smartphones*. O termo *smartphone* foi adotado devido à utilização de um sistema operacional nos celulares.” Então podemos concluir que mesmo com o avanço consideravelmente dos celulares eles estão perdendo o espaço para os

smartphones que utilizam um sistema operacional para realizar a ligação da parte física com a parte virtual do celular.

Segundo a empresa de telecomunicações do Brasil a empresa de telecomunicações TELECO, (2012) no Brasil existem aproximadamente 264 milhões de celulares, ou seja, atualmente existem mais linhas telefônicas do que brasileiros, mostrando uma densidade de 133,99 celulares para cada 100 habitantes. Esses dados estatísticos estão fazendo com que a concorrência aumente cada vez mais entre os fabricantes de linhas móveis, com a concorrência acirrada os grandes nomes de companhias telefônicas estão arriscando em *smartphones* que são celulares com funcionalidades avançadas em comparação ao celular comum e também tem como objetivo de trazer mais facilidade em tarefas simples e até mesmo complexas a partir de um simples clique em um botão ou um toque na tela.

Atualmente, os celulares não servem somente para realizar ligações e encaminhar mensagens, um smartphone tem a capacidade de se conectar com a internet, fazer pagamentos de contas, assistir vídeos em 3D, jogar jogos em alta resolução, transformá-lo em um GPS (Sistema de Posicionamento Global) de alta capacidade, dentre muitas tarefas que são possíveis de serem realizadas através de um aplicativo. Portanto, não só o mercado de fabricantes de celulares está crescendo, mas o mercado de aplicativos tornou-se também bastante competitivo, como podemos ver segundo o PORTAL OLHAR DIGITAL (2013). “A indústria de aplicativos está crescendo como nunca. Os usuários gastam, em média, duas horas por dia com seus apps (aplicativos). E a previsão é de que as lojas cresçam 62% este ano, alcançado a marca de US\$ 25 bilhões em vendas.”

A empresa Microsoft segundo a TECHSPOT (2011) tem o seu sistema operacional em aproximadamente 87% dos computadores em todo o mundo e por isso é mundialmente conhecida, a Microsoft também tem um sistema operacional próprio para celulares o Windows Phone, que no Brasil ainda é pouco conhecido e pouco utilizado pelos usuários de linhas móveis, porém quando fala-se na marca Microsoft e no software Windows todos querem saber do sistema e como funciona em um celular, por esse motivo desenvolver aplicativos para *smatphones* com o

sistema operacional da Microsoft pode se tornar um grande investimento em um futuro próximo.

Desenvolver aplicativos para celulares é uma tarefa que requer tempo e preparo por parte do desenvolvedor que deve estar atento aos passos necessários da engenharia de software para que seu aplicativo possa ser de boa qualidade e também o profissional deve estar sempre atento às novas tendências do mercado, pois o mundo tecnológico e dos celulares é muito volátil, podendo acontecer mudanças em questão de dias, porém com sabedoria e qualidade em seu produto o desenvolvedor pode ser bem sucedido nesse mundo, criando aplicativos criativos e de grande aceitação no mercado.

Segundo TADASHI (2012) “O Ambiente de desenvolvimento para o WP7 é totalmente gratuito, você poderá utilizar o Visual Studio 2010 Express, o Visual Studio é a principal IDE da Microsoft para o desenvolvimento de aplicativos para a plataforma .net”, falaremos mais sobre o ambiente de desenvolvimento, porém até aqui é preciso saber que não teremos que pagar dinheiro nenhum com o desenvolvimento, pois como foi dito as ferramentas são gratuitas, outro ponto que vale destacar também que a linguagem de programação base para a plataforma é o .NET.

Para tanto o estudo se **justifica** pela falta de conhecimento por parte dos desenvolvedores dessa nova tecnologia e falta de pessoas principalmente no Brasil que desenvolvem para essa plataforma. Atualmente um profissional deve estar sempre atento as novas tendências do mercado se quiser ser bem sucedido, pois esse sistema operacional para celulares tem tudo para cair no gosto dos consumidores finais.

Já o **Problema** foi: Considerando que o sistema mais utilizado pela maioria dos fabricantes de aparelhos celulares no Brasil, seja o Android ou IOS (Sistema Operacional Iphone), percebe-se o desconhecimento no desenvolvimento dos aplicativos e até mesmo da falta de desenvolvedores para a plataforma Windows Phone.

Como **Pergunta** que se buscou responder: A falta de conhecimento da plataforma e do desenvolvimento de aplicativos para Windows Phone afeta a criação dos aplicativos?

As **Hipóteses** foram: a) A falta de conhecimento sobre o desenvolvimento prejudica o aplicativo final. b) O conhecimento de aplicativos para Windows Phone sem as ferramentas necessárias para a criação e testes do mesmo pode prejudicar o aplicativo desenvolvido. c) O bom conhecimento sobre a programação para Windows Phone, porém com a falta de conhecimento sobre o sistema e não estar atento com sobre as novas tendências do mercado e de seus consumidores pode prejudicar o desempenho do produto final. d) O conhecimento sobre o desenvolvimento, com as ferramentas necessárias e com o desenvolvedor ligado as novas tendências pode ser a fórmula para o sucesso de um aplicativo.

O **objetivo geral** consistiu em: Estudar sobre o que é o desenvolvimento móvel e o seu mercado, e o que é a plataforma Windows Phone e como é o desenvolvimento de um aplicativo para esse sistema, desde a criação do ambiente para o desenvolvimento, quais as novas tecnologias, a programação em si e as novas ferramentas disponíveis, bem como testar seus aplicativos, objetivando facilitar o processo de criação de aplicativos para dispositivos móveis com esse sistema operacional.

Os **objetivos específicos** foram: a) Fazer um levantamento bibliográfico de informações sobre o que esta em torno de smartphones e sobre o desenvolvimento de aplicativos para Windows Phone. b) Desenvolver um aplicativo desde o seu princípio, visando mostrar as novas ferramentas que podem ser utilizadas na criação e exemplificar os passos das etapas para o processo de desenvolvimento. c) Discutir todas as etapas necessárias sobre a análise e desenvolvimento de um aplicativo e também mostrar tudo que o novo sistema permite fazer, objetivando facilitar a criação de um aplicativo de grande qualidade para Windows Phone.

Como **metodologia** para o desenvolvimento deste trabalho, foi utilizada a pesquisas em diversos livros, artigos, e materias acadêmicos encontrados na internet ou na faculdade, com o objetivo de realizar um levantamento de documentos

importantes para usar como base para o desenvolvimento com abordagem quantitativa. Após o levantamento bibliográfico, será desenvolvido uma aplicação, sobre a plataforma dot.Net, que irá rodar sobre a plataforma Windows Phone. O estudo buscou entender sobre o desenvolvimento de aplicativos para Windows Phone e tudo aquilo que está em sua volta.

Nos próximos capítulos discutiremos sobre todas as questões levantadas, sendo assim este primeiro capítulo é uma introdução do tema proposto a ser discutido, o segundo capítulo reserva-se a uma referenciação bibliográfica sobre o que é o desenvolvimento para aplicativos móveis e o que é Windows Phone. O terceiro capítulo dedica-se ao mercado de dispositivos, onde discutiremos assuntos como: se vale a pena desenvolver para essa plataforma e quem são os consumidores. O quarto capítulo conheceremos e discutiremos o ambiente de desenvolvimento desde a instalação das ferramentas necessárias até as principais inovações do sistema conhecendo um pouco mais sobre os componentes que são utilizados no desenvolvimento. O quinto capítulo desenvolveremos na prática um aplicativo com todo nosso conhecimento adquirido até os capítulos anteriores e com o objetivo de exemplificar todo esse processo, e por fim, com base nas informações conseguidas a partir dos estudos realizados o sexto capítulo reserva - se às considerações Finais.

2. DESENVOLVIMENTO DE APLICATIVOS MÓVEIS

Em primeiro lugar temos que saber o que é esse desenvolvimento para dispositivos móveis e o PORTAL DA MICROSOFT, MSDN (2013) responde essa pergunta em seu portal.

“O desenvolvimento para telefones é um termo abrangente que se refere ao processo de criar aplicativos para smartphones. Os telefones executam um sistema operacional como uma plataforma para fornecer serviços aos aplicativos. Como resultado, o desenvolvimento para telefones é semelhante ao desenvolvimento para a área de trabalho, mas apresenta algumas armadilhas. Os telefones não têm a mesma memória ou o mesmo desempenho dos desktops. Para simplificar o desenvolvimento, vários desenvolvedores inicialmente executam, depuram e testam seus aplicativos em uma versão do software do telefone também conhecida como emulador. Isso permite a criação do aplicativo em um ambiente padrão e familiar, que depois poderá ser implantado em um telefone quando estiver pronto para uso.”

Ou seja, pelo que vimos os *smartphones* executam um sistema operacional como uma plataforma, e assim nós podemos realizar diversas aplicações (apps) para o nosso celular. Em breve veremos quais são os principais sistemas utilizados pelos usuários através de uma pesquisa, vale destacar o Windows Phone dentro desses sistemas que será o estudado por nós.

2.1 WINDOWS PHONE

Quando falamos em *Windows Phone* várias perguntas nos vêm a mente, a primeira delas deve ser o que é isso? Como funciona? Onde surgiu? Entre muitas outras, porém para entendermos tudo sobre esse sistema e sobre o desenvolvimento de aplicativos temos que entender alguns conceitos que estão expostos a seguir.

O *Windows Phone* segundo MOREIRA, (2011) é um “sistema móvel da Microsoft que tornou-se mais uma dentre tantas opções de compra para os usuários brasileiros, oferecendo uma nova proposta de uso, com uma interface simples e funcionalidades integradas ao próprio Windows”, ou seja o *Windows Phone* é um sistema operacional feito especialmente para celulares assim como *IOS* e *Android*

só que feito pela Microsoft. Em 2010, segundo POGUE, (2011) “o sistema parecia incompleto com alguns recursos não implementados, porém com a versão 7.5 apelidado por Mango, a Microsoft acabou com todo esse problemas.” Assim como tudo que é novo parece incompleto, e não pode chamar tanta a atenção como os mais antigos no mercado o *Windows phone* surgiu e não foi muito bem aceito, porém com a versão 7.5 a empresa pois de vez esse sistema no mercado e aos olhos dos consumidores. E assim o sistema operacional da Microsoft entrou com tudo no mercado e tem tudo para ser um grande concorrente, pois nome de peso o sistema já possui e também é uma grande área para quem quer aprender a desenvolver aplicativos para essa plataforma nova no mercado, sobre o desenvolvimento para essa plataforma móvel algumas pessoas podem ter algumas perguntas e duvidas sobre como é o processo de criação de um aplicativo para um celular, porém como vimos anteriormente os telefones executam um sistema operacional como uma plataforma para fornecer serviços aos aplicativos. Ou seja, o desenvolvimento para celulares segue os mesmos passos para o desenvolvimento de qualquer outro software seja para computadores, *tablets* ou qualquer outro dispositivo, porém assim como qualquer outro, o *Windows Phone* também tem características particulares que veremos em breve.

Quando falamos em Windows a maioria das pessoas fazem uma ligação com o computador da sua casa, que como dito anteriormente a maioria das pessoas que possuem computador tem como sistema operacional qualquer versão da Microsoft e querem saber sobre o sistema funcionando em um *smartphone*, e por isso a Microsoft tem uma grande vantagem no mercado com o *Windows Phone* e para garantir a boa qualidade em seus produtos, a marca estabelece uma série de requisitos para as fabricantes de celulares terem o sistema nos seus produtos. Segundo MÔNACO e CARMO, (2012) “o objetivo de Microsoft é oferecer o sistema e não o celular, assim os clientes poderão escolher o celular que mais se adapta e assim utilizar o Windows Phone e seus aplicativos”. Portanto, desse jeito o usuário tem a liberdade de escolher entre marca e modelo que o mesmo gosta, ou melhor, se adapta.

2.1.1 GRANDES INOVAÇÕES DO SISTEMA: WINDOWS PHONE

Como falado anteriormente só de ter no nome do sistema operacional a palavra Windows já chama grande atenção dos usuários de linhas móveis e também dos desenvolvedores, porém não só o nome trás um bom convite para os consumidores obterem celulares com esse sistema, a Microsoft buscou inovar ao máximo para entrar no mercado e algumas dessas novas inovações são de extrema importância para o desenvolvedor conhecer as mesmas, adotarem elas e usufruírem dessas novidades em seu favor enriquecendo assim o seu aplicativo. Uma das grandes inovações e que vale destacar segundo POGUE, (2011) é “A página inicial (Home) apresenta duas colunas de ícones coloridos. Cada um deles representa algo que você colocou lá para melhorar o acesso a determinado recurso: um aplicativo, um nome para discagem rápida, uma lista de músicas ou um arquivo de e-mails.” Essas colunas citadas são compostas por blocos coloridos que são atualizados automaticamente fazendo com que o usuário não precise ir até o aplicativo específico para se ter resposta daquilo que procura, segundo MONACO e CARMO (2012) a idéia da Microsoft é “entregar mais informações com menos cliques, dando ao usuário a possibilidade de obter muito de seus aplicativos sem que precise necessariamente entrar neles”. Um exemplo de bloco que podemos citar para melhorar o entendimento é o bloco de mensagem que atualiza automaticamente informando ao usuário que possui uma nova mensagem como podemos ver na figura 1.



Figura 1: Blocos dinâmicos.
Fonte: Windows Phone – Site oficial do sistema

Tratando também de grandes inovações o Windows Phone traz um sistema mobile, ou seja, o sistema proporciona uma mobilidade na computação, pois o

mesmo está disponível para o usuário vinte e quatro horas por dia onde o mesmo estiver. Outra grande inovação do sistema operacional *Windows Phone* é o seu sistema de busca, segundo POGUE, (2011) “o aplicativo de busca Bing oferece agora busca visual e de áudio. Ou seja, você pode segurar o telefone de forma que ele capte qualquer música que esteja sendo tocada em qualquer lugar; em cerca de três segundos, ele identifica a música e lhe oferece a oportunidade de adquiri-la na Internet.” Como a outra inovação citada anteriormente essa é outra novidade que pode fascinar os consumidores, pois os mesmos terão a possibilidade de pesquisar o que eles quiserem e necessitarem em questões de segundos através do toque, de uma imagem ou até mesmo do áudio, isso também enche os olhos dos desenvolvedores, pois os mesmos pode integrar seus aplicativos com todas essas novidades. Abaixo segue a figura 2 que mostra um celular com o sistema operacional da Microsoft utilizando da ferramenta de busca por imagem.



Figura 2: Ferramenta de busca por imagem.
Fonte: Lima – Site oficial do Windows Phone Brasil

3. MERCADO DOS DISPOSITIVOS

No capítulo anterior vimos quais são os principais sistemas operacionais que são executados em nossos celulares dando uma plataforma para o aplicativo funcionar e neste capítulo foi realizado uma pesquisa com o objetivo de levantar dados junto ao público referente a utilização de *smartphones* sendo que essa pesquisa nos ajuda a tirar conclusões que posteriormente discutiremos.

Esta pesquisa de campo terá como metodologia a análise quantitativa-descritiva, especificamente baseada em estudos de descrição de população que segundo LAKATOS E MARCONI (2009) vem a ser “exata descrição de certas características quantitativas de populações como um todo, organizações ou outras coletividades específicas. Geralmente contém um grande número de variáveis e utilizam técnicas de amostragem para que apresentem caráter representativo”.

3.1 PESQUISA DE CAMPO

A pesquisa foi elaborada contendo um questionário simples com apenas duas perguntas, aplicada a 80 pessoas, entre homens e mulheres. Esta pesquisa foi realizada nas cidades de Santa Bárbara d’Oeste e Americana.

A primeira pergunta realizada foi a: qual sistema operacional você utiliza em seu *smartphone*? E o gráfico 1 representa a quantidade usuários por sistema, sendo que 57% dos usuários utilizam celular com o sistema *Android*, 17% utilizam *IOS*, 8% *Windows Phone*, 3% utilizam outro sistema e 15% dos usuários entrevistados não utilizam celular com sistema operacional.

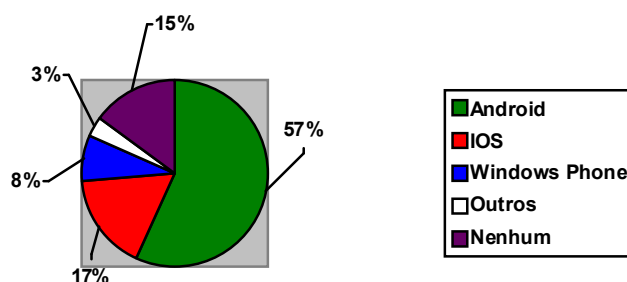


Gráfico 1 - Porcentagem de usuários com celular com sistema operacional.

A segunda pergunta realizada foi a seguinte: Você compraria um celular com Windows Phone? E o gráfico 2 exemplifica a resposta obtida sendo que 78% dos usuários responderão que sim e 22% responderão que não comprariam.

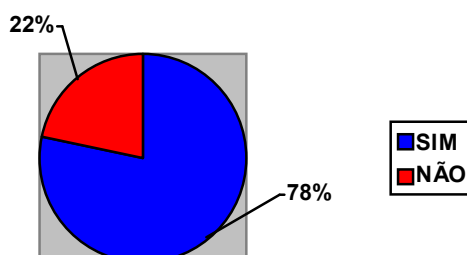


Gráfico 2 - Porcentagem de possíveis usuários de Windows Phone

Através dos dados do primeiro gráfico podemos ver que o sistema mais utilizado atualmente é o *Android* e os outros sistemas operacionais para celulares estão bem abaixo não ocupando um lugar de destaque. O *Windows Phone* que será tratado no trabalho como dito anteriormente ocupa um lugar baixo no gráfico, porém através da segunda pergunta podemos observar que 78% dos usuários comprariam um celular como o sistema da Microsoft se tivessem a oportunidade. Portanto podemos concluir que existe um grande número de usuários de *Windows Phone* em potencial, ou seja, a um curto ou médio prazo a marca pode entrar em ascensão.

3.2 PORQUE DESENVOLVER PARA CELULARES?

Até agora vimos sobre o que é o desenvolvimento de aplicativos e quais os principais sistemas operacionais utilizados, porém será que é um bom mercado desenvolver aplicativos para celulares? Segundo a revista ABRIL (2012) "O balanço da Anatel mostra que a "teledensidade" atingiu a marca de 131,16 acessos para cada cem habitantes, ou seja, há mais de um celular para cada brasileiro." Portanto, podemos ver que para 100 brasileiros existem aproximadamente 131 linhas de celulares, e assim podemos responder a pergunta feita acima que se é um bom negócio desenvolver aplicativos para celular? A resposta é sim, ou seja, o

desenvolvedor que quer explorar esse mercado no Brasil pode se dar muito bem, sendo que até mesmo em um curto prazo o mesmo poderá obter lucro.

Em um estudo feito pela Unisys Corporation segundo ARAUJO (2011) “mostra que no Brasil, 37% dos chamados trabalhadores de informação, ou iWorkers (funcionários que utilizam as tecnologias da informação como parte de seu dia a dia) utilizam smartphone para acessar aplicações de negócios nas suas empresas, tornando o país um dos líderes neste tipo de uso deste dispositivo no trabalho.” Anteriormente vimos que existe um número extremamente alto de celulares, e através desse dados citados podemos ver que boa parte dessa massa utiliza seus aplicativos dentro de um ambiente corporativo, sendo assim podemos tirar como conclusão que não apenas os usuários comuns utilizam *smartphones* mas também os usuários corporativos utilizam essa tecnologia abrangendo ainda mais o mercado de criação de aplicativos.

4. AMBIENTE DE DESENVOLVIMENTO

Segundo MACORATTI, (2013) a arquitetura da plataforma Windows Phone 7 é constituída de 4 principais componentes: Runtime, Tools, Cloud e Portal Services. Podemos ver essa arquitetura na figura 3 que está a seguir:



Figura 3: Arquitetura de desenvolvimento.
Fonte: Macoratti – Site de aprendizado sobre diversos temas

Os componentes citados na imagem anterior estão em inglês e são respectivamente *Runtime* (tempo de execução), *Tools* (ferramentas), *Cloud Services* (serviços em nuvem) e *Portal Services* (portal de serviços). O *Runtime* Segundo MONACO E CARMO, (2012) . É que “todo o desenvolvimento é feito em código gerenciado, em um conceito chamado sandbox, que permite o rápido desenvolvimento de aplicações seguras e protegidas”, ou seja, esse é um conceito esse que faz com que sua aplicação esteja protegida de qualquer interrupção externa. O outro componente da arquitetura são as ferramentas que são fundamentais no processo de criação do aplicativo, de acordo com TADASHI, (2012) “para iniciar o desenvolvimento com o WP7 você vai precisar o Windows Phone developer Tools ele está disponível para o Windows Vista e Windows 7” , todas as ferramentas para o desenvolvimento são:

- Visual Studio 2010 Express for Windows Phone;

- Windows Phone Emulator Resources;
- Silverlight 4 Tools For Visual Studio;
- XNA Game Studio 4.0;
- Microsoft Expression Blend for Windows Phone.

As ferramentas citadas acima podem ser encontradas no endereço: <http://developer.windowsphone.com/en-us> como mostrado também na figura 4 a seguir.

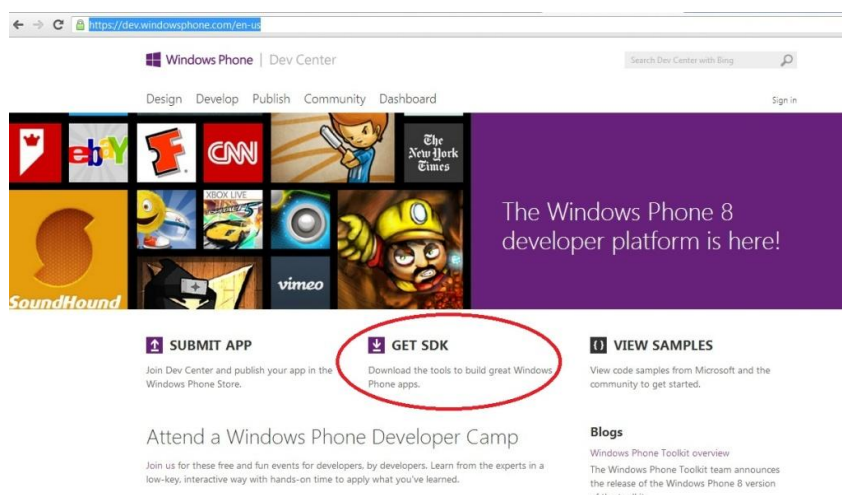


Figura 4: Ferramentas de instalação.

Fonte: Windows Phone – Portal da Microsoft de desenvolvimento.

O terceiro componente da arquitetura são os serviços em nuvem que é o fato segundo AMOROSO, (2012) de “você não precisa instalar aplicativos no seu computador para tudo, pois pode acessar diferentes serviços online para fazer o que precisa, já que os dados não se encontram em um computador específico, mas sim em uma rede.” Ou seja, esse conceito é bastante interessante, pois evita o consumo de memória do celular e da uma flexibilidade maior para acessar aplicativos diretamente da internet sem precisar instala – los em seu celular. O portal de serviços que é o último item da arquitetura segundo MONACO e CARMO, (2012) “é um local centralizado para que os desenvolvedores possam submeter e certificar

suas aplicações. Também é através dele que os consumidores podem localizar e baixar essas aplicações”. Esse local se chama Marketplace e é onde tudo acontece, ou seja, é o local onde acontece a venda de produtos e também o processo de colocar o aplicativo disponível para o usuário.

4.1 REGISTRO DO DESENVOLVEDOR NA MARKETPLACE

Como vimos anteriormente a Microsoft buscou centralizar todos os processos em único lugar, chamado Marketplace, sendo assim tanto o usuário quanto o desenvolvedor irão ao mesmo lugar para procurar e submeter aplicativos. Até agora vimos que em nenhum momento foi preciso pagar taxa nenhuma, porém para ter um acesso como vimos anteriormente na Marketplace é preciso de um registro, que segundo o PORTAL DA MICROSOFT, MSDN (2013)

Você pode escolher dentre muitos e diferentes caminhos para criar aplicativos que exercem maior impacto nos clientes. Quando você escolher o caminho a seguir, uma opção deverá ser considerada cuidadosamente: que tipo de conta de desenvolvedor você deseja criar. Durante o processo de registro, será perguntado se você quer criar uma conta individual ou uma conta empresarial. Escolher certo aqui é muito importante, pois, depois que você escolher, não será possível mudar.

Segundo o mesmo portal temos algumas diferenças de uma conta para outra como mostrado na tabela 1:

Tabela 1: Tipos de contas.

Conta individual	Conta empresarial
<ul style="list-style-type: none"> • Requer cartão de crédito para verificar sua identidade • Não pode usar funcionalidades específicas do aplicativo • Não pode listar um aplicativo de desktop na Windows Store • Custa aproximadamente US\$ 49 (o valor exato varia dependendo da moeda do seu país ou região) 	<ul style="list-style-type: none"> • Requer cartão de crédito para verificar sua identidade • Requer verificação adicional via Symantec • Permite acesso a todas as funcionalidades do aplicativo • Pode listar aplicativos de desktop na Windows Store • Requer que sua empresa seja reconhecida como tal no país ou região onde está localizada • Custa aproximadamente US\$ 99 (o valor exato varia dependendo do seu país ou região)

Fonte: Microsoft – Portal oficial da Microsoft

Porém dentro da conta individual existe uma outra opção que segundo o PORTAL DA MICROSOFT DREAMSPARK (2013) “Com o DreamSpark, os estudantes podem obter acesso gratuito à Microsoft Windows Store e Windows Phone Dev Center, podem enviar seus aplicativos gratuitamente e ganhar dinheiro!” Ou seja, se você é estudante e tem vontade de aprender sobre essa nova ferramenta um cadastro nesse portal poderá te isentar de qualquer taxa no desenvolvimento, e como dito você poderá ganhar dinheiro com seus aplicativos.

Portanto, através de qualquer tipo de cadastro você terá uma conta de desenvolvedor oficial para Windows Phone e poderá submeter seus aplicativos e quem sabe começar a ganhar dinheiro. Se você não for estudante o preço pode ser um pouco caro, porém com os seus aplicativos você poderá ter um retorno do investimento.

Após a realização desse cadastro você terá então uma conta de desenvolvedor oficial para Windows Phone, além disse segundo MONACO e CARMO, (2012) “Ao se tornar um desenvolvedor registrado, você ganha acesso para desbloquear três aparelhos, nos quais você poderá depurar diretamente do Visual Studio”, ou seja, com esse registro você não precisará mais testar seus aplicativos no emulador do computador, você poderá testar seus aplicativos diretamente do seu celular com Windows Phone e assim sentir ao vivo a criação sua. Segundo o portal da MICROSOFT MSDN (2013) os passos para o registro do seu telefone estão a seguir.

Para registrar seu telefone usando a Ferramenta de registro do desenvolvedor Windows Phone.

1. Ligue o telefone e desbloquear o tela do telefone.
2. No seu telefone, verifique se a data e a hora estão corretas.
3. Conecte seu telefone ao computador utilizando o cabo USB fornecido com o telefone.
4. Na tela Iniciar do seu computador, clique no todos os programas.

5. Execute o Windows Phone SDK 8.0, clique em Windows Registo desenvolvedor Telefone. O aplicativo é iniciado, como se mostra na figura 5.

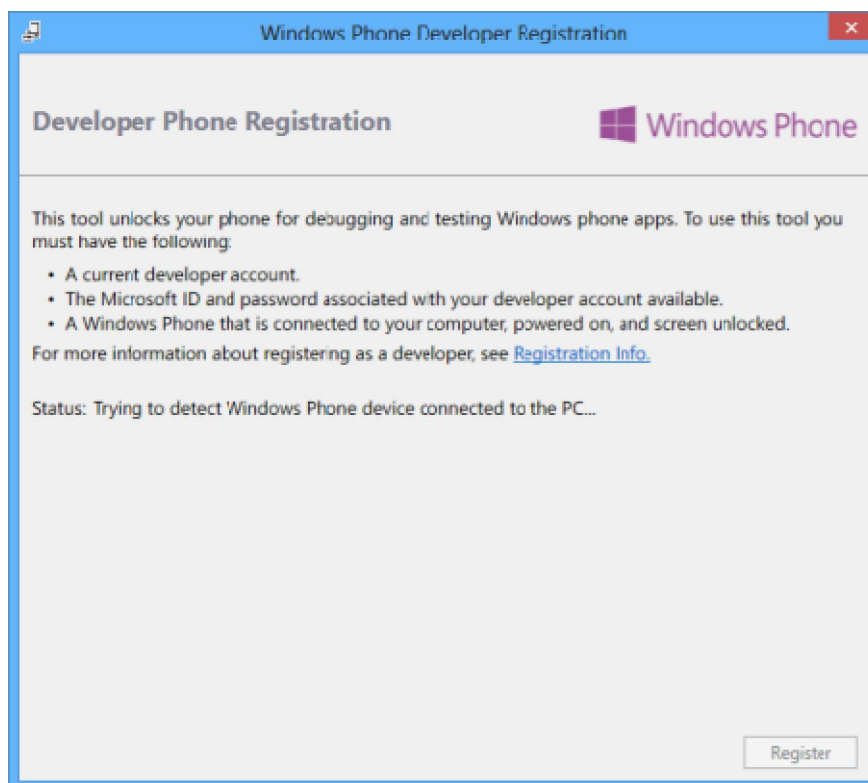


Figura 5: Aplicativo de registro de celular.
Fonte: Fonte: Microsoft – Portal oficial da Microsoft

6. Verifique se a mensagem de status exibe Identificado o dispositivo Windows Phone. Clique no botão Registre-se para desbloquear o telefone. Se o seu telefone já está registrado, a mensagem de status indica isso e você verá um botão Unregister.

7. Clique em registrar.

8. Na caixa de diálogo, entre com sua conta Microsoft (anteriormente conhecido como Windows Live ID), digite a conta Microsoft e senha que correspondem à sua filiação Dev Center. Clique em Conectar.

9. Após o seu telefone está registrado com êxito, a mensagem de status exibe Parabéns! Você desbloqueado com sucesso o seu Windows Phone.

4.2 FERRAMENTAS DE DESENVOLVIMENTO

Como falamos anteriormente na arquitetura do Windows Phone as ferramentas para o desenvolvimento podem ser facilmente encontradas e instaladas, e para o nosso projeto que realizaremos no próximo capítulo e também para o programador novo nessa área a ferramentas mais importante será o Visual Studio.

4.2.1 VISUAL STUDIO

Segundo o portal da MICROSOFT MSDN (2013) “O IDE (Ambiente de Desenvolvimento Integrado) do Visual Studio oferece um conjunto de ferramentas que ajudam você a escrever e modificar código e também detectar e corrigir erros.”. Ou seja, é no Visual Studio que criaremos nossos aplicativo e também onde o desenvolvedor passará a maior parte do tempo, seja criando códigos ou corrigindo os erros. Após a instalação em seu computador o programa conta com uma interface inicial simples onde o usuário poderá escolher o projeto que deseja utilizar como mostramos na figura 6.

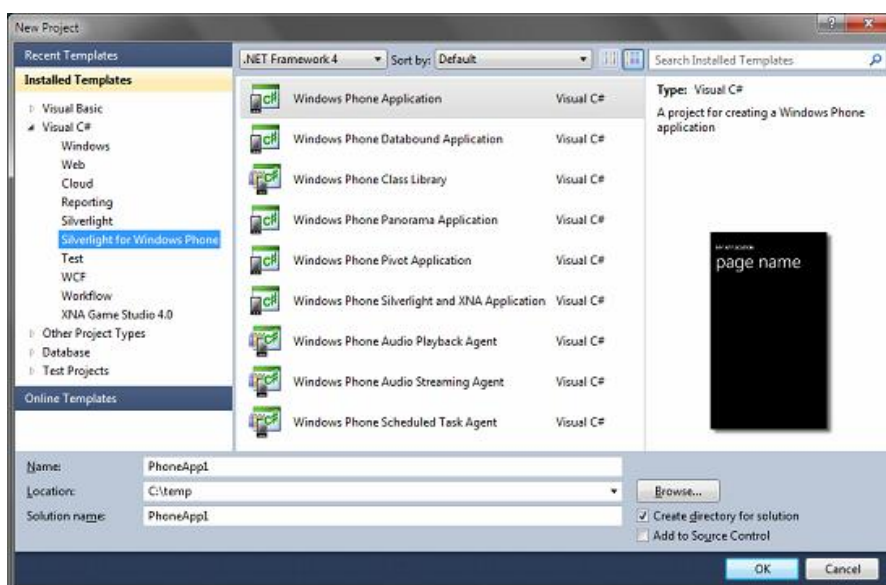


Figura 6: Página inicial do Visual Studio.
Fonte: Tezini – Site oficial do Windows Phone Brasil

Como mostrado na figura anterior temos vários projetos possíveis para serem utilizados, o Windows Phone Application segundo MONACO E CARMO, (2012) “É o template mais simples dentre todos os disponíveis”, ou seja, para uma aplicação simples que não exija recursos avançados do celular este é o melhor projeto que deve ser utilizado. O Windows Phone Panorama Application e o Windows Phone Pivot Application são duas novidades que falaremos logo adiante.

4.2.2 EMULADOR DO WINDOWS PHONE

Outra ferramenta que compõe a arquitetura de desenvolvimento é o emulador de Windows Phone, segundo o dicionário AULETE (2013) o emulador é um “dispositivo, programa etc. designado para fazer com que um computador, programa, sistema etc. emule outro, ou seja, admita as mesmas solicitações e dê os mesmos resultados, ou similares”, ou seja, com essa ferramenta teremos como simular diretamente do nosso computador o celular com Windows Phone e seu aplicativo instalado.

É importante testar nossos aplicativos criados no emulador do computador, porém quando realizamos o cadastro no site oficial da Microsoft como dito nos tópicos anteriores nós temos a possibilidade de cadastrar nosso celular para desenvolvimento, ou seja, assim nós podemos testar os aplicativos em nossos celulares físicos, o que pode gerar uma confiança melhor para colocar o aplicativo no mercado, pois estaremos vendo o mesmo na palma de nossas mãos.

4.3 PRINCIPAIS COMPONENTES

Todos as plataformas de desenvolvimento tem características pessoais e o *Windows Phone* também possui, segundo SATO (2012) “o Windows Phone 7 tem disponível dois controles que estão sendo utilizados com muita frequência na criação de aplicativos diversos, o Pivot e o Panorama”, como citado anteriormente o Pivot e o Panorama são dois tipos de projetos, assim como o Windows Phone Application e a seguir falaremos um pouco mais sobre cada um destacando os pontos principais.

4.3.1 PIVOT

Segundo ARAUJO (2012). O Pivot é um controle de navegação que existe no windows phone que facilita o gerenciamento e exibição de dados para os usuários. O Pivot Control proporciona uma forma de visualizar dados iguais ou similares de diversas formas onde eles possam ser separados por grupos, cada grupo tem um cabeçalho que pode ser clicado como se fosse um menu, levando o usuário até a página que contém os dados. Com ele podemos manipular facilmente um conjunto de informações, ou seja, esse componente tem a função de trazer uma melhoria na forma de visualização dos dados e pode ser facilmente aplicado no seu aplicativo através dos tipos de projetos do Windows Phone no Visual Studio, a seguir trazemos a figura 7 para exemplificar o uso do componente Pivot.

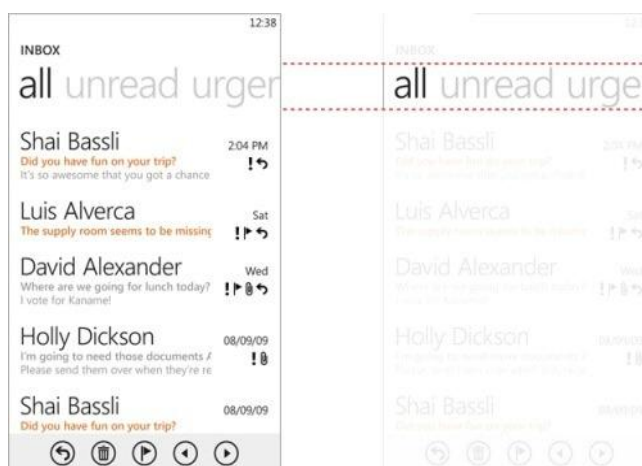


Figura 7: Componente Pivot.
Fonte: Ravi - Blog MSDN

4.3.2 PANORAMA

Segundo a COMUNIDADE WINDOWS MOBILE BRASIL (2011).

O Panoram Control é uma função nativa Windows Phone 7. Ao contrário de aplicações padrão que são projetados para caber dentro dos limites da tela do celular, aplicativos panorâmicos oferecem uma maneira única de ver os controles, dados e serviços usando uma tela horizontal que se estende muito além dos limites da tela.

Esse componente é muito interessante, pois da liberdade do desenvolvedor criar um layout maior que o do tela, afim de que o usuário possa se movimentar na tela do aplicativo através do toque na tela, ou seja esse componente que também pode ser encontrado nas opções de projeto para Windows Phone no Visual Studio é grande fita estendida onde poderá ser desenvolvido seu aplicativo ao longo da mesma, a seguir temos a figura 8 com que tem como objetivo deixar mais claro esse componente.



Figura 8: Componente Panorama.
Fonte: RAVI - Blog MSDN

4.4 CONCEITOS DA PROGRAMAÇÃO

Sobre a linguagem de programação utilizada pela programação para Windows Phone MARIOTTI (2011) fala:

Com o poder de flexibilidade do .NET Framework, não foi necessário a Microsoft criar uma nova linguagem de programação para plataforma móvel. Contudo, foi feita uma customização na framework para suportar a nova plataforma, e atualmente podemos escolher entre as linguagens C# e Visual Basic (VB) para criação de aplicativos.

É importante ressaltar o uso de ferramentas poderosas como Silverlight, XNA e Cloud services. O Silverlight (XAML) permite a criação de telas mais agradáveis para os usuários finais, e também a criação de jogos simples em 2D. Os recursos de desenvolvimento com XNA tem como principal objetivo o desenvolvimento de jogos em 3D. O cloud services disponibiliza o uso de serviços nas nuvens, como por exemplo se conectar a uma base de dados.

Portanto podemos ver que a Microsoft disponibilizou duas linguagens para nós escolhermos qual queremos utilizar para desenvolver o aplicativo, no nosso caso para desenvolvimento do aplicativo utilizaremos o C#, linguagem essa de fácil entedimento para inciantes, outro ponto importante que vale destacar é que para criação de interfaces é utilizado XAML que veremos na pratica na criação do software para o celular.

5. ESTUDO DE CASO - DESENVOLVIMENTO DO APLICATIVO

Nos capítulos anteriores nós falamos sobre o desenvolvimento de aplicativos para *Windows Phone*, vimos um pouco sobre a história do *smartphone*, como é o desenvolvimento para *Windows phone*, como montar o ambiente com as ferramentas necessárias e os principais componentes atuais de programação, porém não entramos na prática do desenvolvimento em si e esse capítulo é dedicado a um estudo de caso tendo como objetivo exemplificar o desenvolvimento de um aplicativo simples para essa tecnologia.

5.1 ESPECIFICAÇÕES

Em primeiro lugar o erro de muitos desenvolvedores e estudantes no desenvolvimento de qualquer aplicativo é não definir o objetivo, os requisitos e criarem diagramas para maior entendimento sobre o software, e assim acabam se perdendo no processo de desenvolvimento e desistem facilmente ou no final o aplicativo criado é totalmente diferente do inicial projetado, ou seja, antes de tudo precisamos definir alguns princípios e cenários básicos da nossa aplicação criada.

O aplicativo que criaremos como exemplo será um aplicativo que pode parecer um pouco complexo, mas desenvolveremos juntos passo a passo. O aplicativo terá como objetivo obter a localização do celular a um determinado período de tempo definido por nós onde enviaremos através de serviços de internet que segundo o PORTAL DE DESENVOLVIMENTO NETBEANS, (2013). “São componentes de aplicação distribuídos que estão disponíveis externamente. Você pode usá-los para integrar aplicações de computador que são escritas em linguagens diferentes e executadas em plataformas diferentes.” a latitude e longitude do mesmo que terá como objetivo gravar em um banco de dados essas informações, assim o usuário do celular poderá se por ventura o mesmo perder, for roubado ou esquecer o celular em qualquer lugar ele poderá entrar facilmente em um site onde o usuário através de um nome de usuário e senha realizado na primeira vez que o mesmo entrar no aplicativo pelo *smartphone* poderá obter a última posição do celular e assim o localiza-lo. Um ponto muito importante do aplicativo é que teremos duas frentes, após o usuário realizar o login no sistema pela primeira vez o aplicativo

inicializara em segundo plano e também será mostrado na tela do usuário a atual localização, ou seja, o aplicativo estará em segundo plano mandando as informações de localização dando a possibilidade do usuário abrir outras aplicações e serviços. Um requisito bastante importante para o aplicativo funcionar é que o mesmo necessita de internet para obter a localização e chamar os serviços de internet, ou seja, para o aplicativo funcionar completamente o mesmo necessitará de conexão com a internet. Nosso aplicativo terá a seguinte arquitetura como mostra a figura 9.

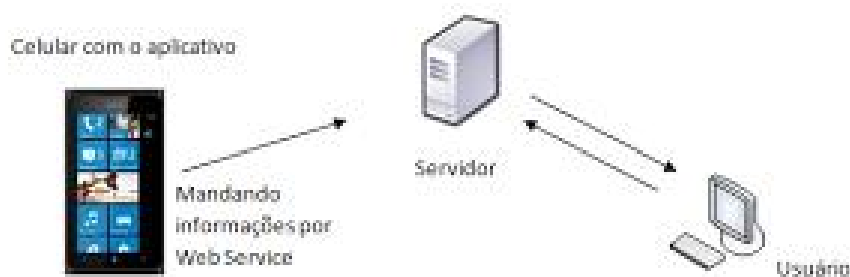


Figura 9: Arquitetura de desenvolvimento
Fonte: Blog windowsphonedsv

Teremos três pilares fundamentais para o desenvolvimento. O primeiro deles é o celular Windows Phone com o aplicativo instalado, o software terá uma tela inicial de acesso ao sistema onde o mesmo poderá cadastrar um usuário e uma senha para acesso ao sistema informando algumas informações adicionais, um ponto importante que vale destacar é que quando o usuário criar um nome de usuário para ele ao mesmo tempo será criado na tabela de banco de dados um registro para o celular do mesmo, evitando do usuário do celular criar mais de uma conta para o celular. Assim que o usuário acessar o sistema começa o processo de localização geográfica do mesmo, onde será exibida na tela a posição atual do celular e a mesma se atualizara sozinha caso venha a mudar a latitude e longitude, e também em segundo plano se inicializará a aplicação enviando as informações para o servidor. O segundo integrante como vimos anteriormente é o servidor que tem como função principal armazenar os serviços de internet e o banco dados, em nosso aplicativo o servidor será local, ou seja, em nossa própria máquina, e o terceiro e último integrante é o site que será o canal para o usuário obter a localização do

celular e outras informações, em nosso site o usuário poderá detectar a última localização gravada pelo sistema, poderá também pesquisar uma localização informando a data e hora e também alterar os dados de registro do usuário.

Em nosso trabalho focaremos o primeiro integrante do sistema que é o aplicativo, ensinando a criar interface, as validações necessárias, a navegação entre as telas, obter a localização, chamar os serviços de internet e colocar o aplicativo para rodar em segundo plano. Vale destacar também que não entraremos em normas e questões da segurança da informação, pois senão o trabalho ficaria muito EXTENSO.

Agora que estamos dando cara ao aplicativo que será desenvolvido vamos criar dois diagramas que facilitaram o entendimento do aplicativo e nos ajudará no processo de desenvolvimento. O primeiro deles será o diagrama de casos de uso onde mostraremos os personagens e as funcionalidades do sistema.

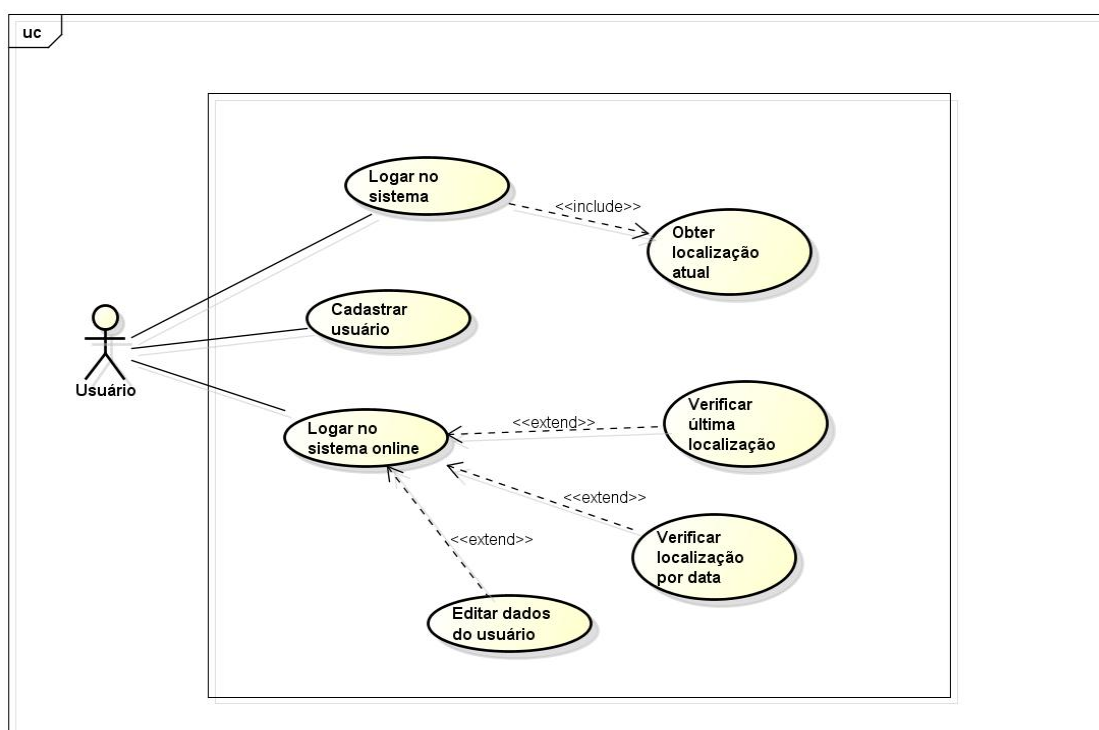


Figura 10: Diagrama de casos de uso.
Fonte: Blog windowsphonedsv

O outro diagrama será o diagrama de classe como mostra a figura 11.

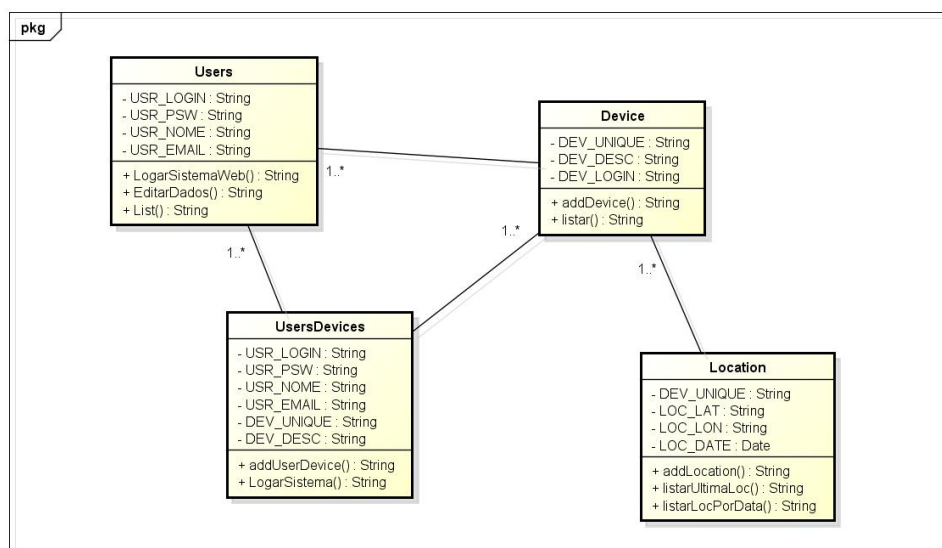


Figura 11: Diagrama de classes
Fonte: Blog windowsphonedsv

Pronto, agora podemos colocar em prática depois de entendido um pouco mais sobre o sistema, então o passo agora que devemos realizar depois de instalada as ferramentas e montado o ambiente é criar o projeto, como citado no capítulo anterior criaremos o projeto padrão de aplicativos o “Windows Phone Application” e nosso aplicativo será chamado de “SecurityPhone”. Após criarmos o aplicativo será mostrada a tela inicial padrão do projeto onde nela criaremos a interface de maneira bem simples. Inicialmente vamos montar a hierarquia do nosso sistema que basicamente terá quatro telas sendo: a primeira tela inicial já criada pelo visual Studio a “MainPage.xaml” onde o usuário entrará no sistema através do nome de usuário e senha, ele poderá entrar também na tela de cadastro e na tela de informações do aplicativo, a segunda tela será a tela de cadastro do usuário onde o mesmo informará informações básicas, a terceira tela será um mapa em tela cheia exibindo a localização atual do celular e a quarta e última tela terá informações sobre o aplicativo.

Então teremos que criar essas três páginas, o processo para criar as páginas é bem simples, precisamos clicar com o botão direito do mouse em cima do projeto criado no menu do Visual Studio que fica a esquerda do computador depois em

adicionae e posteriormente em novo item, então abrirá uma caixa de dialogo exibindo os itens que podem ser adicionados no projeto, ou seja, clicaremos em “Windows Phone Portrait Page” e escolheremos o nome da página, realizaremos esse processo três vezes colocando respectivamente os seguintes nomes nas páginas: “Cadastro.xaml”, “Mapa.xaml” e “Sobre.xaml” ficando de acordo com a imagem 12 apresentada a seguir.

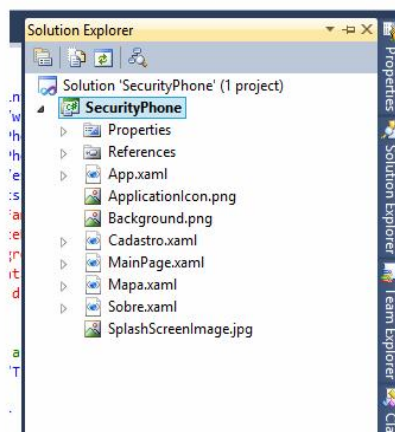


Figura 12: Arquitetura do projeto
Fonte: Blog windowsphonedsv

5.2 INTERFACE

Depois de criado as páginas e colocados os nomes nas mesmas temos que criar a interface, existem vários jeitos de criarmos um layout para a tela, porém o mais simples e fácil é selecionarmos a ferramenta que necessitamos no menu caixa de ferramentas que se localiza no lado direito do computador e arrastarmos para a tela do celular, e assim colocando item por item de acordo com o que necessitamos desse modo também podemos redimensionar os itens e posicionar da maneira que achamos melhor.

A tela inicial será composta basicamente de cinco itens diferentes o bloco de texto que é o texto que será exibido na tela do celular, a caixa de texto que será o campo onde o usuário digitara o seu nome de usuário, a caixa de senha onde o usuário digitara sua senha, o botão que quanto for clicado p usuário terá acesso ao sistema e o componente botão de link onde o usuário terá acesso a tela de cadastro

de usuário e a tela de informações do sistema. Lembrando que sempre quando arrastamos um item na tela do celular é gerado um código XAML para o mesmo ficando na tela da direita este código. A seguir, temos a figura 13 que representa a tela inicial do aplicativo junto com as ferramentas e a interface dessa primeira tela.

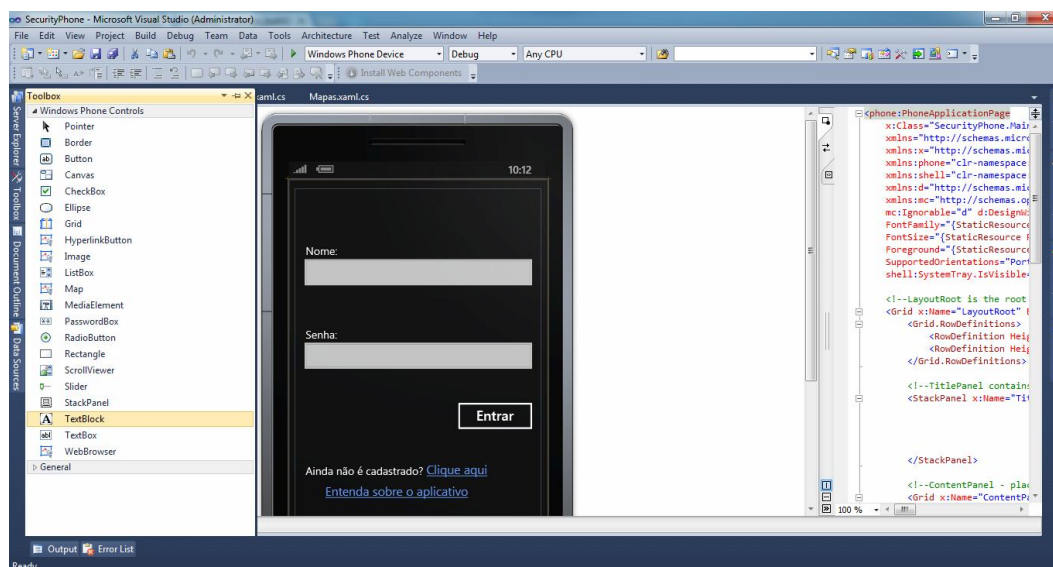


Figura 13: Caixa de ferramentas e tela inicial
Fonte: Blog windowsphonedsv

Depois de criado a interface e colocado todos os componentes na tela nós podemos alterar as propriedades dos itens, nós alteramos eles com o objetivo de personaliza - los da maneira que nós queremos, sendo assim para alterar as propriedades é bem simples, basta selecionarmos o item desejado, clicarmos com o botão em cima do mesmo e depois em propriedades. A primeira alteração necessária que deve ser realizada é no nome de cada componente, pois ficará mais fácil no desenvolvimento e no entendimento do código, por exemplo, entramos na propriedade da caixa de texto onde será colocado o nome de usuário e mudamos a propriedade para “txtLogin”, ou seja, sempre quando quisermos manipular esse componente basta chamar por esse nome.

Outra personalização que podemos realizar nessa página é a dos componentes de bloco de texto, pois quando inserimos os mesmos o texto exibido é padrão do visual Studio e temos que alterar, então nós podemos alterar a

propriedade texto para aquilo que queremos que seja exibido, note que os elementos de bloco de texto devem ficar igual a imagem anterior, para trocar o texto exibido no botão e no botão de link temos que alterar a propriedade conteúdo desses itens mencionados. Outra propriedade que podemos alterar também é a cor dos itens de texto, essa propriedade se chama primeiro plano e quando clicamos na mesma podemos escolher a cor que quisermos em uma lista de opções.

A tela de cadastro é composta por blocos de texto para exibir a informação correspondente ao item, por caixas de texto e senha e para a entrada das informações necessárias, por botão de link para a o usuário poder voltar a tela principal e do componente botão para a efetivação do cadastro. Nessa tela vale destacar que temos a possibilidade de limitar a quantidade de caracteres dos campos de entrada de dados com a opção comprimento de texto que se encontra nas propriedades dos elementos de caixa de texto. A seguir temos a figura 14 que mostra a tela de cadastro que deve ser seguida como exemplo.

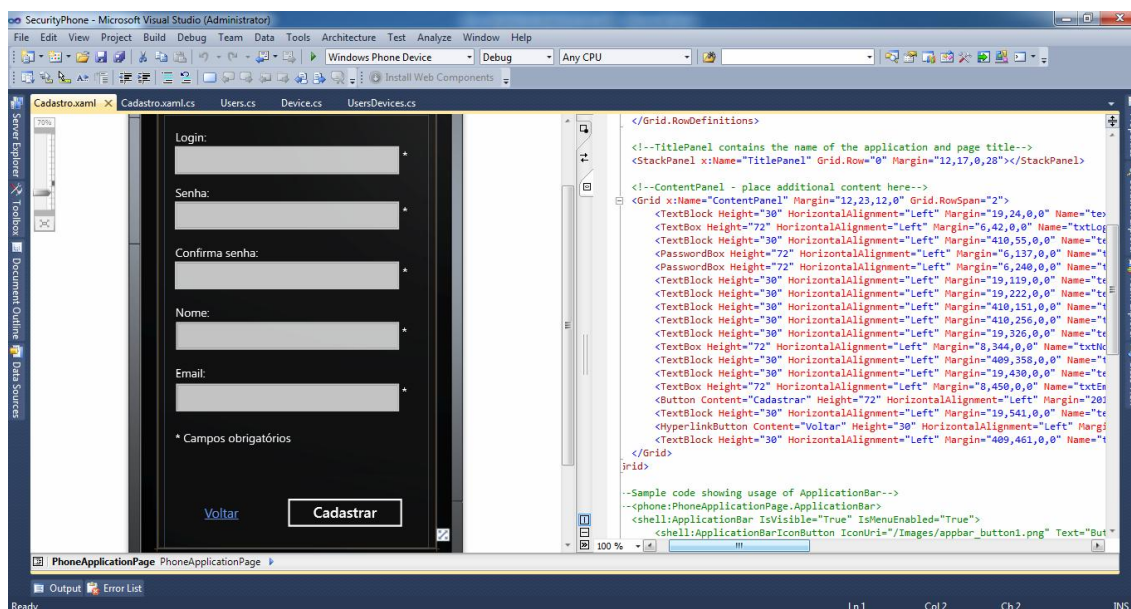


Figura 14: Tela de Cadastro
Fonte: Blog windowsphonedsv

Já a tela de informações do aplicativo é composta por blocos de texto informados o nome e algumas informações e especificações para o usuário e por um botão de link para o usuário poder voltar a tela principal do aplicativo. Nesta tela vale

destacar que podemos alterar também o tamanho da fonte do componente de bloco de texto na propriedade tamanho da fonte. A seguir temos a figura 15 que mostra a tela de informações do sistema que também deve ser seguida como exemplo.

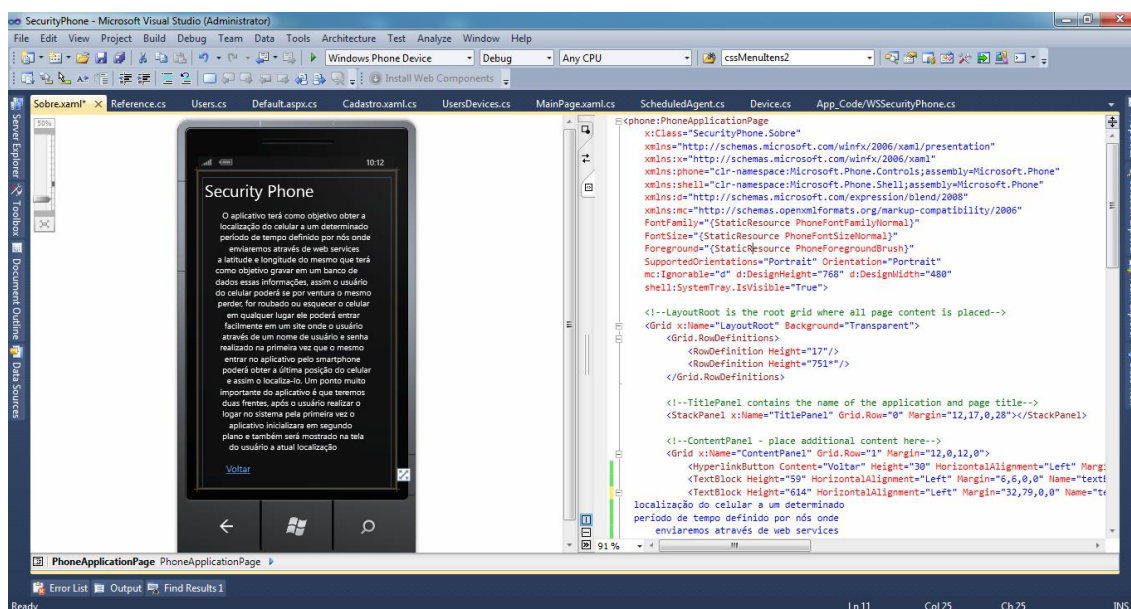


Figura 15: Tela de informações do sistema
Fonte: Blog windowsphonedsv

5.3 NAVEGAÇÃO

Criado todas as interfaces chegou a hora de realizarmos a navegações entre as páginas do aplicativo, para exemplificar melhor vamos usar a tela principal do aplicativo, quando o usuário quiser acessar o sistema o mesmo deverá clicar sobre o botão de acesso criado por nós na interface, então para isso o mesmo deve criar um evento em seu botão, para criar um evento o desenvolvedor somente clica duas vezes sobre o botão e assim o mesmo será redirecionado para a página de código onde o mesmo deverá colocar o seguinte código abaixo.

```

private void btnEnter_Click(object sender, RoutedEventArgs e)
{
    NavigationService.Navigate(new Uri("/Mapa.xaml", UriKind.Relative));
}

```


Outro modo de realizamos a navegação entre as páginas é quando colocamos o elemento de botão de link, e para realizáramos a chamada para outra página nós podemos simplesmente alterar uma propriedade do elemento que é a *NavigateUri* e assim teremos a navegação. Portanto na página inicial no link “Clique aqui” colocamos para chamar a tela “Cadastro.xaml”, no link “Entenda como funciona” colocamos para chamar a página “Sobre.xaml” e nas página de cadastro como na tela de informações sobre o sistema no link “voltar” colocamos para chamar a tela principal. A seguir temos na figura 16 um exemplo de navegação pela propriedade.

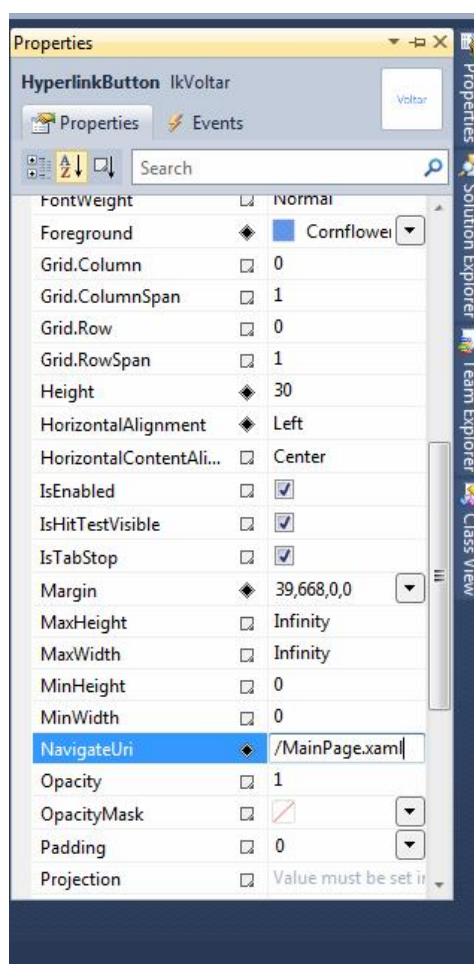


Figura 16: Navegação por propriedade
Fonte: Blog windowsphonedsv

5.4 VALIDAÇÕES

As validações estão presentes em qualquer outro aplicativo e no nosso não podemos esquecer, pois o usuário não pode se cadastrar com uma senha em branca ou menor do que definimos como padrão ou até mesmo tentar acessar o aplicativo sem digitar seu nome de usuário e senha.

Para exemplificar vamos usar a tela de cadastro de usuário onde temos o campo nome de usuário, senha, confirmar senha, nome e email sendo que todos são obrigatórios, então a primeira validação que precisamos realizar é para vermos se os mesmo estão sendo preenchidos ou não e podemos realizar essa validação comparando os campos com dados vazios, outra validação que devemos realizar, é quando criamos a interface colocamos na propriedade de cada caixa de texto um tamanho máximo para ele, porém não colocamos um tamanho mínimo para os mesmo, ou seja, essa deve ser a segunda validação que devemos realizar contando cada letra digitada e comparando com a quantidade mínima estabelecida por nós e uma última validação dessa página deve ser uma comparação entre dois campos para ver se os mesmos estão iguais que são a senha e confirmar senha e para realizar essas validações temos o seguinte código.

```
private void button1_Click(object sender, RoutedEventArgs e)
{
    if (verificarCampos())
    {
        if (verificarLogin())
        {
            if (verificarSenha())
            {
                try
                {
                    MessageBox.Show("Cadastrado com sucesso!");
                }
                catch (Exception lex)
                {
                    MessageBox.Show(lex.ToString());
                }
            }
        }
    }
}

private bool verificarSenha()
{
    if (txtSenha.Password == txtSenha2.Password)
    {
```

```

        if (txtSenha.Password.Length >= 6)
        {
            return true;
        }
        else
        {
            MessageBox.Show("A senha deve conter entre 6 a 12 caracteres");
            return false;
        }
    }
    else
    {
        MessageBox.Show("As senhas estão incompatíveis.");
        return false;
    }
}

private bool verificarCampos()
{
    if (txtLogin.Text == string.Empty || txtNome.Text == string.Empty ||
txtSenha.Password == string.Empty || txtSenha2.Password == string.Empty ||
txtEmail.Text == string.Empty)
    {
        MessageBox.Show("Existem campos obrigatórios que devem ser
preenchidos");
        return false;
    }
    else
    {
        return true;
    }
}

private bool verificarLogin()
{
    string login = txtLogin.Text;

    if (login.Length >= 6)
    {
        return true;
    }
    else
    {
        MessageBox.Show("O login deve conter entre 6 a 20 caracteres.");
        return false;
    }
}
}

```

Agora na tela inicial precisamos somente realizar duas validações, uma no sentido de comparar se os campos nome de usuário e senha estão preenchidos ou não e a outra é o requisito principal do sistema que é a verificação da conexão por internet, ou seja, nós temos que ver se o celular possui ou não sinal da internet e podemos realizar essa validação através do seguinte código.

```
if (NetworkInterface.GetIsNetworkAvailable())
{
    //possui conexão
}
else
{
    //não possui conexão
}
```

5.5 SERVIÇOS DA WEB

No começo desse capítulo falamos que enviaremos através de serviços da internet algumas informações para serem processadas e armazenadas no servidor e agora veremos como são feitas as chamadas desses serviços, a princípio teremos três chamadas em nosso aplicativo, o primeiro serviço será depois de todas as validações na tela de cadastro, que realizara a gravação no banco de dados das informações do celular e do usuário, o segundo serviço será o de autenticação do sistema, ou seja, será para validar a entrada do usuário no software e o terceiro e último será o de envio da posição geográfica para o banco de dados. Como falado anteriormente o *web service* será um sistema a parte do nosso projeto, sendo assim nós teremos que referencia – lo no nosso projeto, este processo é bem simples e podemos realizar da seguinte maneira, abrimos o menu do projeto na pasta serviços de referências clicamos com o botão direito do mouse, depois em adicionar serviço de referência e assim abrirá uma caixa de dialogo onde podemos digitar a url do serviço e posteriormente clicamos no botão ir, após checar se o seu serviço apareceu você poderá clicar em ok e assim teremos a referencia em nosso projeto. Após realizado esse processo você terá somente que realizar a chamada do método que está em seu serviço passando os parâmetros necessários para o mesmo. Abaixo temos o exemplo da chamada do serviço de internet de cadastro do usuário com os dados digitados pelo mesmo, nesse ponto também mostrarei como se coleta informações do *smartphone* como o *deviceuniqueid* que é um código único de cada celular e também o modelo do mesmo para gravação no banco de dados, ou seja, nessa tela de cadastro realizaremos o registro tanto do usuário quanto do celular no banco de dados da seguinte forma.

```
try
{
    string DeviceName = DeviceStatus.DeviceName;
```

```

        byte[] myDeviceID =
(byte[])Microsoft.Phone.Info.DeviceExtendedProperties.GetValue("DeviceUniqueId");
        string DeviceIDAsString =
Convert.ToBase64String(myDeviceID);

        ServiceReference1.WSSecurityPhoneSoapClient
lServiceReference1 = new ServiceReference1.WSSecurityPhoneSoapClient();

lServiceReference1.addUserDeviceAsync(txtLogin.Text.ToUpper(),
txtSenha.Password.ToUpper(), txtNome.Text.ToUpper(), txtEmail.Text.ToUpper(),
DeviceIDAsString.ToUpper(), DeviceName.ToUpper());

        lServiceReference1.addUserDeviceCompleted +=new
EventHandler<ServiceReference1.addUserDeviceCompletedEventArgs>(lServiceReference1_add
UserDeviceCompleted);
    }
    catch (Exception lex)
    {
        MessageBox.Show(lex.ToString());
    }

    private void lServiceReference1_addUserDeviceCompleted(object obj,
ServiceReference1.addUserDeviceCompletedEventArgs e)
    {
        string lResult = e.Result.ToString();

        if (lResult == "1")
            MessageBox.Show("Usuário cadastrado com sucesso.");
        else if(lResult == "2")
            MessageBox.Show("Login de usuário já existente, tente novamente.");
        else if (lResult == "3")
            MessageBox.Show("Celular já cadastrado.");
        else
            MessageBox.Show(lResult);
    }
}

```

Desse modo nós temos a chamada do serviço da internet que nosso caso tem o papel de gravar as informações no banco de dados, um último ponto importante se tratando desses serviços é que sempre quando alteramos o mesmo e ele faz parte do nosso projeto temos que atualizar ele nosso sistema, e podemos realizar da seguinte forma clicamos com o botão direito do mouse em cima do serviço referenciado em nosso projeto depois em atualizar serviço de referência.

5.6 POSICIONAMENTO GEOGRÁFICO

Agora a parte principal do nosso projeto, o posicionamento geográfico do celular, esta localização atual do dispositivo é encontrado por uma classe já definida na biblioteca do *Windows Phone* que é a *GeoCoordinateWatcher*, esta classe padrão já existente é responsável por obter a latitude e longitude do dispositivo, ela possui

atributos e métodos específicos que utilizaremos em nosso código como por exemplo o atributo `MovementThreshold` define a quantos metros da distancia anterior ele vai obter a localização atual do dispositivo, um método que utilizaremos e que é padrão também dessa classe é `watcher_PositionChanged` que é acionado a cada atualização da posição geográfica, ou seja, a cada mudança de latitude e longitude esse evento é disparado, assim o mapa que será mostrando na tela se atualizará, a seguir temos o código completo da página para um melhor entedimento.

```
using System;
using System.Data;
using System.Collections.Generic;
using System.Linq;
using System.Xml;
using System.Net;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Animation;
using System.Windows.Shapes;
using Microsoft.Phone.Controls;
using System.Windows.Navigation;
using System.Globalization;
using System.Device;
using System.Device.Location;
using Microsoft.Phone.Tasks;
using Microsoft.Phone.Net.NetworkInformation;
using Microsoft.Phone.Info;

namespace SecurityPhone
{
    public partial class Mapas : PhoneApplicationPage
    {
        GeoCoordinateWatcher watcher;
        //inicializamos o mapa nesse instante
        BingMapsTask bing;

        public Mapas()
        {
            InitializeComponent();

            bing = new BingMapsTask();

            watcher = new GeoCoordinateWatcher(GeoPositionAccuracy.Default);

            watcher.MovementThreshold = 2;

            //Nesse ponto chamamos a o método de mudança de posição
            watcher.PositionChanged += new
            EventHandler<GeoPositionChangedEventArgs<GeoCoordinate>>(watcher_PositionChanged);

            //Inicializamos a classe watcher aqui.
            watcher.Start();
        }
    }
}
```

```

    }

    void watcher_PositionChanged(object sender,
    GeoPositionChangedEventArgs<GeoCoordinate> e)
    {
        Deployment.Current.Dispatcher.BeginInvoke(() => ChangeUI(e));
    }

    void ChangeUI(GeoPositionChangedEventArgs<GeoCoordinate> e)
    {
        //aqui estabelecemos a posição de acordo com a latitude e longitude
        bing.Center = new GeoCoordinate(e.Position.Location.Latitude,
    e.Position.Location.Longitude);
        //aqui colocamos o nível de zoom do mapa
        bing.ZoomLevel = 16;
        bing.Show();
    }
}
}
}

```

Portanto assim nós conseguimos obter uma nova posição a cada dois metros de distancia que o mesmo se alterar, sempre atualizando a posição do mapa.

5.7 APLICAÇÃO EM SEGUNDO PLANO

Agora umas das partes principais do nosso sistema, é criar um agente que segundo o PORTAL DA MICROSOFT, MSDN (2013) permite que um aplicativo execute código em segundo plano, mesmo quando o aplicativo não está sendo executado em primeiro plano, ou seja, para o nosso trabalho é fundamental usarmos esse agente, para criarmos esse serviço temos que clicar com o botão direito do mouse em cima do nosso projeto, depois em adicionar e novo projeto, depois escolhemos o projeto “*Windows phone Scheduled Task agent*” e colocamos o seguinte nome “ScheduledTaskAgentLocation”, após criarmos esse projeto o mesmo cria uma classe onde será executado o algoritmo que ficara em segundo plano. O primeiro passo é realizar a chamada desse agente de segundo plano em nosso aplicativo inicial, chamaremos esse serviço após o usuário se logar no sistema mas antes temos que verificar se o mesmo já esta funcionando, se já estiver vamos excluir o mesmo e iniciá-lo novamente a fim de sempre atualizarmos o serviço. Um ponto importante da chamada é que teremos que estabelecer um tempo determinado para chamar esse serviço e para fins de testes colocarei um tempo curto para vermos na prática. Para exemplificar essa chamada, a remoção e tempo temos o código abaixo comentado para ajudar no desenvolvimento, lembrando

sempre que no portal MSND citado anteriormente temos exemplos e explicações do agente de segundo plano e muitos outros assuntos.

```
#define DEBUG_AGENT
using System;
using System.Collections.Generic;
using System.Linq;
using System.Xml;
using System.Net;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Animation;
using System.Windows.Shapes;
using Microsoft.Phone.Controls;
using Microsoft.Phone.Info;
using System.Net.NetworkInformation;
using Microsoft.Phone.Scheduler;

namespace SecurityPhone
{
    public partial class MainPage : PhoneApplicationPage
    {
        public MainPage()
        {
            InitializeComponent();
        }

        private void StartPeriodicAgent(string periodicTaskName)
        {
            // Nesse ponto verificamos o serviço existente, no nosso caso quando
            chamamos esse método passamos para ele na variável periodicTaskName, então o mesmo
            encontrará o serviço existente.
            PeriodicTask periodicTask = ScheduledActionService.Find(periodicTaskName)
as PeriodicTask;

            //Se a tarefa já existe e os agentes de segundo plano estão habilitados
            para a aplicação,
            //você deve remover a tarefa e, em seguida, adicioná-lo novamente para
            atualizar o serviço
            if (periodicTask != null)
            {
                //chama o método de remoção de serviço
                RemoveAgent(periodicTaskName);
            }

            //instancia um novo serviço
            periodicTask = new PeriodicTask(periodicTaskName);

            // Descrição do serviço
            periodicTask.Description = "Serviço de demonstração de localização.";

            try
            {
                //adiciona o serviço
                ScheduledActionService.Add(periodicTask);
            }
        }
    }
}
```



```

        //inicia o agente em um determinado tempo.
        #if(DEBUG_AGENT)
            ScheduledActionService.LaunchForTest(periodicTaskName,
TimeSpan.FromSeconds(60));
        #endif
    }
    catch (InvalidOperationException lInvalidOperationException)
    {
        MessageBox.Show(lInvalidOperationException.Message);
    }
    catch (SchedulerServiceException lSchedulerServiceException)
    {
        MessageBox.Show(lSchedulerServiceException.Message);
    }
}

//método que remove o serviço
private void RemoveAgent(string name)
{
    try
    {
        ScheduledActionService.Remove(name);
    }
    catch (Exception lex)
    {
        MessageBox.Show(lex.Message);
    }
}

private void btnEnter_Click(object sender, RoutedEventArgs e)
{
    //verificar se o celular possui sinal da internet
    if (NetworkInterface.GetIsNetworkAvailable())
    {
        //chama o método que valida os campos
        if (ValidarCampos())
        {
            try
            {
                //obtem o nome do celular. Ex: Lumia 710
                string DeviceName = DeviceStatus.DeviceName;

                //obtem o id unico do celular
                byte[] myDeviceID =
(byte[])Microsoft.Phone.Info.DeviceExtendedProperties.GetValue("DeviceUniqueId");
                string DeviceIDAsString = Convert.ToBase64String(myDeviceID);

                //cria uma variavel do servico de internet
                ServiceReference1.WSSecurityPhoneSoapClient lServiceReference1
= new ServiceReference1.WSSecurityPhoneSoapClient();

                //chama o método do servico de internet
                lServiceReference1.LogarSistemaAsync(txtName.Text.ToUpper(),
txtSenha.Password.ToUpper(), DeviceIDAsString.ToUpper(), DeviceName.ToUpper());

                //chama o método de retorno do serviço
                lServiceReference1.LogarSistemaCompleted += new
EventHandler<ServiceReference1.LogarSistemaCompletedEventArgs>(lServiceReference1_Logar
SistemaCompleted);
            }
            //É chamado se ocorrer algum erro durante o algoritimo
            catch (Exception lex)

```

```

        {
            //aparece o erro na tela do usuário
            MessageBox.Show(lex.Message);
        }
    }
    else
    {
        MessageBox.Show("Existem campos obrigatórios que precisam ser
preenchidos.");
    }
}
else
{
    MessageBox.Show("A aplicação necessita de conexão com a internet,
tente novamente mais tarde.");
}
}

//método que é chamado após o serviço da internet "web service"
private void lServiceReference1_LogarSistemaCompleted(object obj,
ServiceReference1.LogarSistemaCompletedEventArgs e)
{
    string lResult = e.Result.ToString();

    //se o método retornar o valor 1 o usuário entrará no sistema
    if (lResult == "1")
    {
        //chamada do serviço de referencia
        StartPeriodicAgent("TaskLocation");
        //chamada da página de mapas
        NavigationService.Navigate(new Uri("/Mapas.xaml", UriKind.Relative));
    }
    else
    {
        MessageBox.Show(lResult);
    }
}

//Método que verifica se os campos estão vazios
private bool ValidarCampos()
{
    //compara se os campos estão vazios
    if (txtName.Text == string.Empty || txtSenha.Password == string.Empty)
    {
        return false;
    }
    else
    {
        return true;
    }
}
}
}
}

```

Como falamos anteriormente quando criamos o agente de segundo plano é criada uma classe padrão automaticamente chamada "ScheduledAgent.cs" onde será executado todo o código, se nós abrirmos ela veremos que essa classe já possui conteúdo padrão e também existe um método *invoke* onde será executada

toda ação , e no nosso caso inicializaremos a classe padrão de geolocalização, sendo obteremos a latitude e longitude e posteriormente enviaremos por serviço da internet para o servidor local. Vale destacar que para fins de testes coloquei um tempo curto para iniciar esse serviço. Abaixo temos o código dessa classe comentado em alguns momentos.

```
#define DEBUG_AGENT
using System.Windows;
using Microsoft.Phone.Scheduler;
using System.Device.Location;
using System;

namespace ScheduledTaskAgentLocation
{
    public class ScheduledAgent : ScheduledTaskAgent
    {
        string lTask = string.Empty;

        GeoCoordinateWatcher lWatcher;
        string lResult = "1";

        private static volatile bool _classInitialized;

        /// <remarks>
        /// ScheduledAgent constructor, initializes the UnhandledException handler
        /// </remarks>
        public ScheduledAgent()
        {
            if (!_classInitialized)
            {
                _classInitialized = true;
                // Subscribe to the managed exception handler
                Deployment.Current.Dispatcher.BeginInvoke(delegate
                {
                    Application.Current.UnhandledException +=
ScheduledAgent_UnhandledException;
                });
            }
        }

        /// Code to execute on Unhandled Exceptions
        private void ScheduledAgent_UnhandledException(object sender,
ApplicationUnhandledExceptionEventArgs e)
        {
            if (System.Diagnostics.Debugger.IsAttached)
            {
                // An unhandled exception has occurred; break into the debugger
                System.Diagnostics.Debugger.Break();
            }
        }

        /// <summary>
        /// Agent that runs a scheduled task
        /// </summary>
        /// <param name="task">
        /// The invoked task
        /// </param>
        /// <remarks>
```


Desse modo acabamos aqui o nosso agente de serviço de segundo plano, lembrando que o mesmo é sempre chamado depois de sessenta segundos.

5.8 PORTAL DA INTERNET

Não entraremos muito em detalhes sobre o portal da internet, pois como já dito não é o foco do nosso trabalho, porém podemos destacar alguns pontos importantes, o primeiro deles é que o portal terá um tela inicial onde o usuário deverá se logar no sistema com o nome de usuário e senha cadastrados no celular assim como mostra a figura 17 a seguir.



Figura 17: Portal da internet
Fonte: Blog windowsphonedsv

Após o acesso ao sistema o usuário terá um menu a esquerda com algumas opções como Monitoramento onde será quebrado em mais dois itens, o primeiro última localização onde o usuário poderá escolher entre os seus celulares cadastrados e depois poderá clicar em pesquisar, então será exibido através de um mapa a última localização do seu celular, o outro item é o localizar por data onde o usuário informará uma data de início e uma data de fim para ser rastreado as posições do seu celular, então será exibido através de uma tabela e o usuário poderá clicar em uma localização para ser exibida no mapa que ficara ao lado. O outro item do menu é o gerenciamento onde o usuário poderá alterar as informações

de cadastro como email, nome e senha. Para exemplificar melhor temos abaixo a figura 18 que mostra a tela de localização pela última data informada.

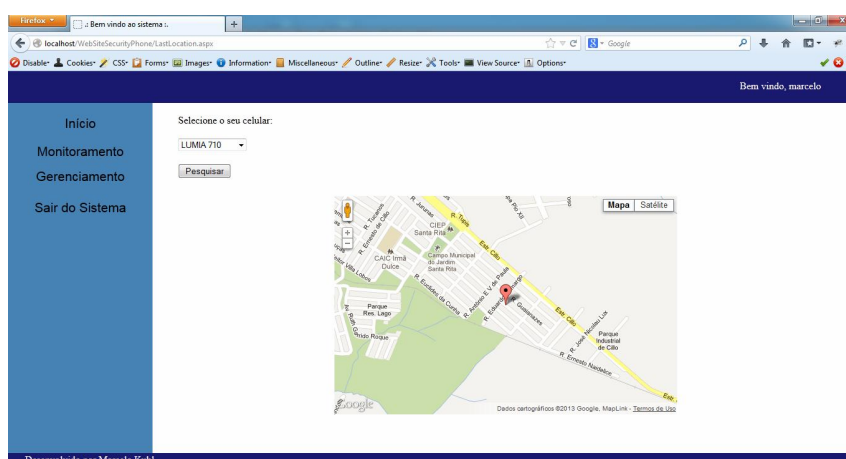


Figura 18: Tela de última localização
Fonte: Blog windowsphonedsv

Como citado anteriormente teremos também a página de pesquisas de posicionamento geográfico a partir de uma data de início e uma data final, onde será apresentado todas as localizações que foram encontradas nesse período, a figura 19 mostra essa página.

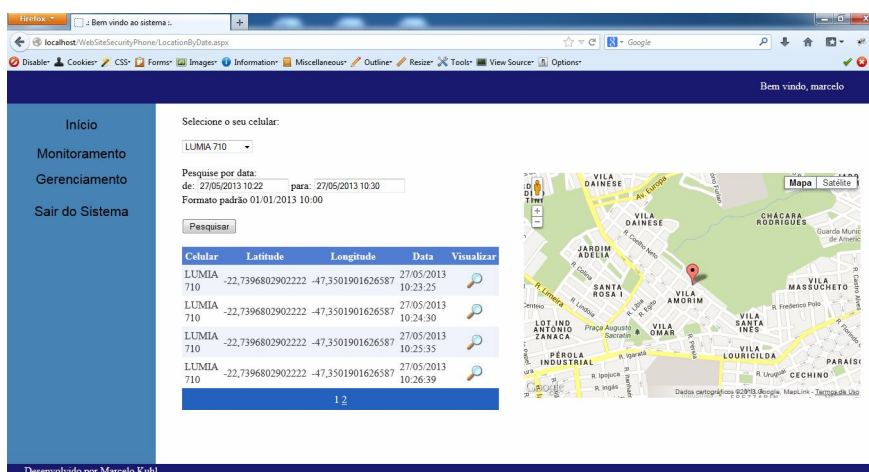


Figura 19: Tela de localização por data
Fonte: Blog windowsphonedsv

Para finalizarmos a parte do portal da internet segue abaixo na figura 20 uma ilustração da página onde o usuário poderá alterar as suas informações pessoais como, por exemplo, o e-mail.

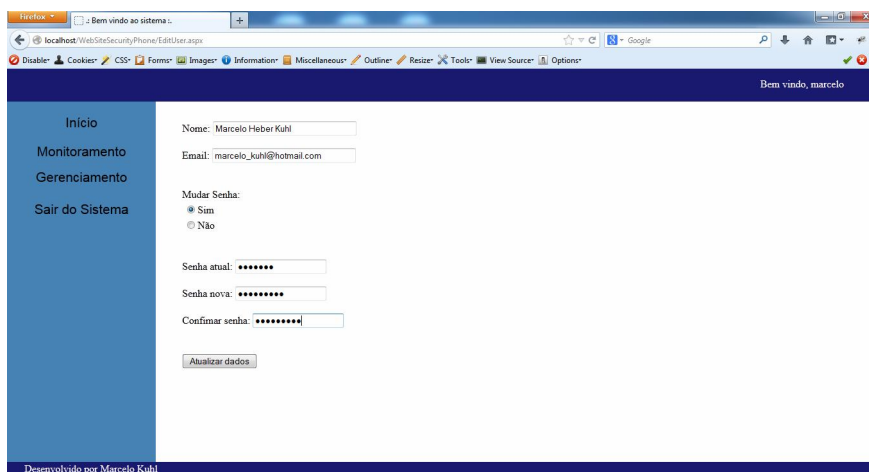


Figura 20: Tela de edição de dados do usuário
Fonte: Blog windowsphonedsv

Assim concluímos o desenvolvimento do aplicativo e do portal da internet, para maiores informações sobre o desenvolvimento deste sistema e de desenvolvimento para Windows phone encontramos no portal da internet [HTTP://windowsphonedsv.blogspot.com.br](http://windowsphonedsv.blogspot.com.br).

CONCLUSÃO

De acordo com todos os dados e informações que tivemos nesse trabalho podemos tirar uma conclusão bem relevante. Em primeiro lugar nós vimos que somente o nome do sistema operacional Windows pode chamar muita a atenção do usuário, outro ponto que podemos citar é que hoje existe uma parcela muito grande de pessoas que utilizam celulares tanto para fins pessoais e profissionais e muitas dessas pessoas se tivessem oportunidade comprariam um celular com o sistema *Windows Phone*, outro ponto é que se você for estudante você não gastará um centavo para começar o desenvolvimento de aplicativos e poderá ganhar dinheiro através da venda do seu produto e um último ponto que colocamos é que a Microsoft trouxe inúmeras inovações aos seus celulares, dando uma possibilidade imensa de personalizações para o desenvolvedor utilizar em seu aplicativo. Além disso, através do PORTAL DA MICROSOFT, o MSDN, citado várias vezes ao longo do trabalho, encontramos diversos exemplos e explicações de classes e códigos para o desenvolvimento. Portanto, conclui-se que há diversas vantagens para quem quer começar a desenvolver e a única desvantagem é que o sistema não é muito aceito pelo usuário atualmente pelo número de pessoas que utilizam o sistema, porém isso pode mudar em um futuro próximo. Enfim, desenvolver aplicativos para *Windows phone* pode ser uma área onde o profissional poderá crescer aprendendo conceitos importantes de programação, como vimos no aplicativo de exemplo, e poderá ganhar dinheiro como um incentivo maior para o desenvolvedor estudar assuntos relacionados a dispositivos móveis e estar atento as novidades do mercado atual no mundo da tecnologia.

GLOSSÁRIO DE TERMOS

SOFTWARE: Em computador ou sistema de computação, os elementos não físicos de processamento de dados, como programas, sistemas operacionais etc.

WP7: Windows Phone versão 7.

.NET: Linguagem de programação criada pela Microsoft.

LATITUDE: Distância angular entre um ponto qualquer da esfera terrestre e o equador, contada sobre o meridiano que passa por esse ponto.

LONGITUDE: Distância angular entre o meridiano 0o (Greenwich) e qualquer ponto da Terra.

LINK: Trecho, palavra ou ícone que conecta um ponto a outro em documentos e sites.

REFERÊNCIAS

ABRIL, Revista. **Brasil terminou agosto com 257,9 mi de celulares.** Disponível em: <http://veja.abril.com.br/noticia/economia/brasil-terminou-agosto-com-257-9-mi-de-celulares>. Acesso em 19/01/2013.

AMOROSO, Danilo. **O que é Computação em Nuvens?**. Disponível em: <http://www.tecmundo.com.br/computacao-em-nuvem/738-o-que-e-computacao-em-nuvens-.htm>. Acesso em 19 Jan. 2013.

ARAUJO, Aluizio. **O mercado e o desenvolvimento para dispositivos móveis.** Disponível em: <http://www.tiespecialistas.com.br/2011/09/o-mercado-e-o-desenvolvimento-para-dispositivos-moveis/#.URyoa6Wkq6G>. Acesso em 19 Jan. 2013.

ARAUJO, Márcio. **Controle Pivot no windows phone.** Disponível em: <http://devbrasil.net/profiles/blogs/controle-pivot-no-windows-phone>. Acesso em: 28 mar. 2013.

AULETE, Dicionário. Editora: Lexikon Editora Digital. Disponível em: <http://aulete.uol.com.br/emulador>. Acesso em: 20 out. 2012.

BLOG WINDOWSPHONEDSV (Marcelo Héber Kuhl). Disponível em: [HTTP://windowsphonedsv.blogspot.com.br](http://windowsphonedsv.blogspot.com.br). Acesso em: 01 jun. 2012.

COMUNIDADE WINDOWS MOBILE BRASIL. **TUTORIAL: Panorama Control para Windows Phone.** Disponível em: <http://winmobr.org/index.php/tutorial-panorama-control-para-windows-phone/>. Acesso em: 28. Mar. 2013.

CUNHA, Afonso. **Confira a história do celular no Brasil.** Disponível em: <http://estreladomunimafcunha.blogspot.com.br/2013/02/confira-historia-do-celular-no-brasil.html>. Acesso em: 10 Abr. 2013.

JORDÃO, Fábio. **História: a evolução do celular.** Disponível em: <http://www.tecmundo.com.br/celular/2140-historia-a-evolucao-do-celular.htm>. Acesso em: 10 Abr. 2013.

LIMA, Alexandre. **[Dicas] Você realmente sabe o que é e para que serve o Bing Vision do seu Windows Phone?**. Disponível em: <http://windowsphonebrasil.com.br/dicas-voce-realmente-sabe-o-que-e-e-para-que-serve-o-bing-vision-do-seu-windows-phone/>. Acesso em: 17 Jan. 2013.

MARCONI, Marina de Andrade e LAKATOS, Eva Maria. Fundamentos de metodologia científica. 5ª edição. São Paulo. Atlas 2003

MACORATTI, José Carlos. **Windows Phone 7 – Introdução.** Disponível em: http://www.macoratti.net/11/01/wp7_int1.htm. Acesso em 07 Fev. 2013.

MÔNACO, Thiago e CARMO, Rodolpho Marques do. **Desenvolvimento de aplicativos para Windows Phone**. 1ª edição. Rio de Janeiro. Brasport. 2013

MOREIRA, Eduardo. **O que é Windows Phone 7 e por que ter um celular com ele?**. Disponível em: <http://www.techtudo.com.br/artigos/noticia/2011/10/o-que-e-windows-phone-7-e-por-que-ter-um-celular-com-ele.html>. Acesso em: 05 dez. 2012.

MARIOTTI, Flávio Secchieri. **Desenvolvendo para Windows Phone 7 Mango**. Disponível em: <http://www.linhadecodigo.com.br/artigo/3284/desenvolvendo-para-windows-phone-7-mango.aspx>. Acesso em: 02 Mai. 2013.

PACIEVITCH, Thais. **História do Telefone**. Disponível em: <http://www.infoescola.com/curiosidades/historia-do-telefone/>. Acesso em: 10 Abr. 2013.

POGUE, David. **Windows Phone traz inovação e empolga, mas ainda não é um sistema maduro**. Disponível em: <http://tecnologia.uol.com.br/noticias/nyt/2011/10/27/o-windows-phone-esta-de-volta-cheio-de-otimos-truques.htm>. Acesso em: 28 nov. 2012.

PORTAL DA MICROSOFT, MSDN. **O que é o desenvolvimento para telefones? (visão geral)**. Disponível em: <http://msdn.microsoft.com/pt-br/ff380145>. Acesso em: 15 out. 2012.

_____. **Selecionando o tipo de conta correto (Windows)**. Disponível em: <http://msdn.microsoft.com/pt-br/library/windows/apps/jj863494.aspx>. Acesso em: 13 fev. 2013.

_____. **How to register your phone for development**. Disponível em: [http://msdn.microsoft.com/en-us/library/windowsphone/dev/elo/ff769508\(v=vs.105\).aspx](http://msdn.microsoft.com/en-us/library/windowsphone/dev/elo/ff769508(v=vs.105).aspx). Acesso em: 28 mar. 2013.

_____. **Apresentando o Visual Studio IDE (C++)**. Disponível em: <http://msdn.microsoft.com/pt-br/library/bb384847%28v=vs.90%29.aspx>. Acesso em: 28 Mar. 2013.

_____. **Background agents for Windows Phone**. Disponível em: [http://msdn.microsoft.com/en-us/library/windowsphone/develop/hh202942\(v=vs.105\).aspx](http://msdn.microsoft.com/en-us/library/windowsphone/develop/hh202942(v=vs.105).aspx). Acesso em: 25 Mai. 2013.

PORTAL DE DESENVOLVIMENTO NETBEANS. **Introdução aos Web Services**. Disponível em: https://netbeans.org/kb/docs/websvc/intro-ws_pt_BR.html?print=Yes. Acesso em 05 Jan. 2013

PORTAL DREAMSPARK. **Cód. do DreamSpark**. Disponível em: <http://e5.onthehub.com/WebStore/OfferDetails.aspx?ws=e3cc15fe-949b-e011-969d-0030487d8897&vsro=8&o=0d78d2b1-5bf0-e111-a77d-f04da23e67f6&JSEnable=1>. Acesso em: 05 Jan 2013.

PORTAL OLHAR DIGITAL. **Indústria de apps deve faturar US\$ 25 bi em 2013.** Disponível em: http://olhardigital.uol.com.br/negocios/digital_news/noticias/industria-de-apps-deve-faturar-us-25-bi-em-2013. Acesso em 11 Mar. 2013.

PORTAL SÃO FRANCISCO. **O Telefone no Brasil.** Disponível em: <http://www.portal.saofrancisco.com.br/alfa/historia-do-telefone/historia-do-telefone.php>. Acesso em: 10 Abr. 2013.

RAVI, Nitya. **Silverlight and Windows Phone SDK.** Disponível em: http://blogs.msdn.com/b/silverlight_sdk/archive/2011/05/25/new-addition-to-windows-phone-quickstarts-pivot-and-panorama-controls.aspx. Acesso em 10 Abr. 2013.

SATO, Alexandre Tadashi. **Controles Pivot e Panorama no Windows Phone 7 - Revista .net Magazine 88.** Disponível em: <http://www.devmedia.com.br/controles-pivot-e-panorama-no-windows-phone-7-revista-net-magazine-88/21983>. Acesso em: 11 dez. 2012.

TADASHI, Alexandre. **Introdução ao Windows Phone 7.** Disponível em: <http://www.inhadecodigo.com.br/artigo/3168/introducao-ao-windows-phone-7.aspx>. Acesso em 09 dez. 2012.

TECHSPOT. **Windows XP usage finally falls below 50% Mark.** Disponível em: <http://www.techspot.com/news/44902-windows-xp-usage-finally-falls-below-50-mark.html>. Acesso em 10 Abr. 2013.

TELECO. **Estatísticas de Celulares no Brasil.** Disponível em: <http://www.teleco.com.br/ncel.asp#>. Acesso em 11 Abr. 2013.

TEZINI, Gilberto. **Desenvolvimento Windows Phone – Primeiros Passos.** Disponível em: <http://windowsphonebrasil.com.br/desenvolvimento-windows-phone-primeiros-passos/>. Acesso em 17 Jan. 2013.

WINDOWS PHONE. **Fique mais perto das pessoas e das coisas importantes para você.** Disponível em: <http://www.windowsphone.com/pt-br/features>. Acesso em 10 Jan. 2013.

WINDOWS PHONE. **Get Sdk.** Disponível em: <http://developer.windowsphone.com/en-us>. Acesso em: 10 Jan. 2013.