

Lucas Macedo Astério De Sousa

Lemys Michel Souza Nascimento

**PROTÓTIPO DE APLICAÇÃO ANDROID PARA PEDIDOS DE PIZZAS**

Trabalho de conclusão de curso apresentado a Faculdade de Tecnologia da Zona Sul como parte da obtenção do titulo de Tecnólogo de Analise e Desenvolvimento de Sistemas

SÃO PAULO

2016



Lucas Macedo Astério De Sousa

Lemys Michel Souza Nascimento

**PROTÓTIPO DE APLICAÇÃO ANDROID PARA PEDIDOS DE PIZZAS**

Trabalho de conclusão de curso apresentado a Faculdade de Tecnologia da Zona Sul como parte da obtenção do titulo de Tecnólogo de Analise e Desenvolvimento de Sistemas

**Orientador:**

Prof. Esp. Denise Lemes Fernandes Neves

SÃO PAULO

2016

**AGRADECIMENTOS**

Lucas Sousa:

Agradeço primeiramente a Deus por ter me dado saúde e força para superar as dificuldades diárias. Aos meus pais Soneide e Augusto, pelo amor, incentivo e apoio incondicional nos momentos difíceis.

A esta faculdade, seu corpo docente, direção e administração que oportunizaram a janela que hoje vislumbro um horizonte superior. Ao professor Alex Macedo de Araújo pela ajuda, e principalmente a professora Denise Lemes Fernandes Neves que sem ela e nosso esforço não seria possível à conclusão desse trabalho. A todos que direta ou indiretamente fizeram parte da minha formação, muito obrigado.

Lemys Michel:

Primeiramente agradeço а Deus que permitiu que tudo isso acontecesse, ao longo de minha vida, е não somente nestes anos como universitário, mas que em todos os momentos é o maior mestre que alguém pode conhecer, aos meus avós e irmãos pelo incentivo, e apoio incondicional.

*"É melhor lançar-se à luta em busca do triunfo mesmo expondo-se ao insucesso, que formar fila com os pobres de espírito, que nem gozam muito nem sofrem muito; E vivem nessa penumbra cinzenta sem conhecer nem vitória nem derrota.”*

*(Franklin Roosevelt)*

**RESUMO**

Analisando o mercado de pizzas, nota-se um setor em expansão contínua no Brasil. Baseado em uma pesquisa quantitativa, foi constado que 60% dos pedidos são realizados através de chamadas telefônicas. Portanto este trabalho visa o desenvolvimento e utilização de aplicativos móveis no setor de pizzarias, possibilitando a praticidade e rapidez no processo de solicitação de pedidos via mobile. Foi desenvolvido um protótipo para dispositivos móveis, com sistema operacional Android, com o objetivo de automatizar o processo de pedidos em pizzarias e consequentemente potencializar a venda nos estabelecimento.

*Palavras chave: Aplicativo Mobile, Android, Pedido,* Pizzaria.

**ABSTRACT**

*Analyzing at the market pizzas, there is a sector in continuous expansion in Brazil. Based on a quantitative survey, it was noted that 60% of applications are made through telephone calls. Therefore this work aims at the development and use of mobile applications in the pizzerias sector, enabling the convenience and speed in the application process requests via mobile. A prototype was developed for mobile devices with Android operating system, in order to automate the process of order in pizzerias and consequently speed up the sale in the establishment.*

*Keywords: Mobile Application, Android, Application, Pizzeria.*

**LISTA DE FIGURAS**

Figura 1 : Grafico de Viabilidade........................................................................28

Figura 2: Caso de Uso.......................................................................................30

[Figura 3: Diagrama de Classe 38](#_Toc444419492)

[Figura 4: Diagrama de Atividades 36](#_Toc444419494)

[Figura 5: Modelo de Diagrama de Entidade Relacional 38](#_Toc444419495)

[Figura 6: Modelo Protótipo Mobile – Tela de Login 41](#_Toc444419499)

[Figura 7: Modelo Protótipo Mobile - Tela de cadastro 42](#_Toc444419500)

[Figura 8: Modelo Protótipo Mobile - Tela de Tipo de Pizza 43](#_Toc444419501)

[Figura 9: Modelo Protótipo Mobile - Tela de pedidos realizados 44](#_Toc444419502)

[Figura 10: Modelo Protótipo Mobile - Tela de Confirmação de compra 45](#_Toc444419503)

[Figura 11: Modelo Protótipo Mobile - Tela de confirmação de endereço 46](#_Toc444419504)

[Figura 12: Modelo Protótipo Mobile - Tela de alteração de endereço 47](#_Toc444419505)

[Figura 13: Modelo Protótipo Mobile - Tela de Confirmação de Compra 48](#_Toc444419506)

**LISTA DE TABELAS**

Tabela 1: Custo do projeto.................................................................................27

Tabela 2: Pesquisa da viabilidade......................................................................28

Tabela 3: Dicionario de dados............................................................................40

Tabela 4: Matriz de responsabilidade completa.................................................53

**LISTA DE QUADROS**

Quadro 1: Matriz de responsabilidade resumida................................................26

Quadro 3: Cenarios ...........................................................................................31

Quadro 4: Criação de Cadastro......................................................................... 42

Quadro 5: Escolha o tamanho Pizza .................................................................43

Quadro 6: Escolha de sabores...........................................................................44

Quadro 7: Manter pedidos..................................................................................45

Quadro 8: confirmação de endereço..................................................................46

Quadro 9: Tela de confirmação de Pedido.........................................................47

Quadro 10: Confirmação de compra...................................................................48

SUMÁRIO

[1 O PROJETO 12](#_Toc445670454)

[1.1 REFERENCIAL TEÓRICO 13](#_Toc445670455)

[2 FUNDAMENTAÇÕES TEÓRICAS 16](#_Toc445670456)

[2.1 DESENVOLVIMENTO 16](#_Toc445670457)

[2.2. REQUISITOS 19](#_Toc445670458)

[2.2 LEVANTAMENTO DE REQUÍSITOS 19](#_Toc445670459)

[2.3 VALIDAÇÃO DE REQUISITOS 20](#_Toc445670460)

[2.4 MODELAGEM DE SISTEMAS 21](#_Toc445670461)

[2.5 DIAGRAMAÇÃO UML 21](#_Toc445670462)

[2.6 MODELO DE RELACIONAMENTO DAS TABELAS 23](#_Toc445670463)

[2.7 DICIONÁRIO DE DADOS 24](#_Toc445670464)

[2.8 FERRAMENTAS DE DESENVOLVIMENTO 25](#_Toc445670465)

[3 ESTUDO DE CASO 26](#_Toc445670466)

[3.1 MATRIZ DE PAPEIS E RESPONSABILIDADES E CRONOGRAMA 26](#_Toc445670468)

[3.2 CUSTOS DO PROJETO 28](#_Toc445670469)

[3.3 ANÁLISE DA VIABILIDADE 29](#_Toc445670470)

[3.4 REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS 30](#_Toc445670471)

[3.5 DETERMINAÇÃO DAS FUNCIONALIDADES DO SISTEMA 30](#_Toc445670472)

[3.6 DIAGRAMA DE CASO DE USO 31](#_Toc445670473)

[3.7 CENÁRIOS DO CASO DE USO 32](#_Toc445670474)

[3.8 DIAGRAMA DE CLASSES 38](#_Toc445670475)

[3.9 DIAGRAMA DE ATIVIDADES 39](#_Toc445670476)

[3.10 CRIAÇÃO DE DIAGRAMAS DE RELACIONAMENTO DAS TABELAS 40](#_Toc445670477)

[3.11 DICIONÁRIO DE DADOS 41](#_Toc445670478)

[3.12 LAYOUT DAS TELAS E RELATÓRIOS 42](#_Toc445670479)

[4. CONCLUSÃO 50](#_Toc445670480)

[5. REFERÊNCIA BIBLIOGRÁFICA 51](#_Toc445670481)

[6. ANEXOS 52](#_Toc445670482)

**INTRODUÇÃO**

Em um mundo globalizado o avanço tecnológico como internet, Maiorino (2003) relata que o desenvolvimento de software e aplicativos trouxe consigo benefícios que se traduzem na informatização e velocidade de tarefas diárias, métodos de localização através de *GPS* e obtenção de informação de forma instantânea por redes sociais e sites, mas principalmente as tecnologias móveis trazem praticidade aos usuários oferecendo agilidade, facilidade, mas como foco principal de solucionar problemas.

Se comparar a situação à do século anterior há mudanças em todas as áreas conforme afirma Toffler (1995, p.142) essa nova civilização traz consigo novos estilos de família; maneiras diferentes de trabalhar, amar e viver; uma nova economia; novos conflitos políticos; e acima de tudo uma consciência modificada.

O avanço da tecnologia e da internet tem desempenhado um importante papel, transformando nossos hábitos de compra.

Um relatório do eCommercenews¹ intitulado *“Global Powers of Retailing* in 2020” afirma que o acesso à rede móvel teve o maior impacto sobre a indústria de varejo até o momento, mas que há também um conjunto mais amplo de tecnologias em jogo, que estão impactando as tendências sociais e econômicas no varejo.

Segundo Carneiro (2002) no ramo empresarial, empresas de grande, médio e pequeno porte, fazem o uso dessa tecnologia móvel para melhorar as suas relações comerciais, disponibilizam prestações de serviços aos clientes de forma variada entre a oferta tradicional sendo ela comercialmente nos locais físicos ou via web e móvel.

[[1]](#footnote-1)

Conforme Fischmann (2015) e os avanços já mencionados, aplicativos como *Hellofood*, *iFood* entre outros nos permitem efetuar compras em apenas alguns *clicks* sem a necessidade do consumidor se deslocar ao estabelecimento e essa facilidade tem se melhorando a cada dia e mais usuários tendo confiança em utiliza-la; percebendo esse mercado que se renova a cada ano este projeto tem como objetivo desenvolver um protótipo de toda parte documental de um aplicativo que gerencie as entregas em *delivery* de uma especifica pizzaria.

O mercado atual de pizzarias não apenas em São Paulo, mas no Brasil em geral mesmo com a crise atual teve um crescimento de 8% e movimenta cerca de R$ 8,5 bilhões ao ano, segundo o empresário José Antônio Baggio, sócio proprietário da rede de pizzarias Baggio explicou no site da jornalista Mirian Gasparin2 que com menos dinheiro no bolso, os consumidores ao invés de irem a um restaurante à noite, estão preferindo comprar uma pizza e comer em casa. Com isso, quatro pessoas podem comer bem, gastando em média R$ 50,00.[[2]](#footnote-2)

Este trabalho será dividido em três capítulos onde o primeiro capítulo irá abordar especificações do negócio do projeto; o segundo capítulo aborda as analises feitas para o desenvolvimento do projeto, o terceiro capítulo demonstra de como o aplicativo será desenvolvido, representação do banco de dados que irá ser utilizado e os protótipos das telas e os relatórios respectivos de cada tela.

### O PROJETO

Atualmente o Brasil é o segundo maior consumidor de pizza do mundo, perdendo apenas para os Estados Unidos. Tem aproximadamente 36 mil pizzarias, dentre essas, 6 mil delas se concentram no estado de São Paulo e consumem diariamente 1,5 milhão de pizzas, sendo 800 mil delas somente no estado de São Paulo, colocando o estado como um dos cinco maiores consumidores per capita do mundo de acordo com a congresso nacional de Pizza o CONPIZZA.

Segundo informações do site Grupo HelloFood³ o setor de pizza no Brasil movimenta cerca de R$ 8,5 bilhões ao ano, com uma estimativa de crescimento de 8% ao ano, porém com esse crescimento muitas dessas empresas no ramo alimentício não conseguem se desenvolverem e ficam estagnadas mediante a sua concorrência.

Segundo Tauhata (2010) os celulares vão se tornar o principal canal de acesso à internet nos próximos anos e seu principal canal são os aplicativos. De acordo com a consultoria Gartner, especializada em tecnologia, o faturamento das lojas de programas ou aplicativos para smartphones e tablets em todo o mundo deve saltar de US$ 6,2 bilhões em 2010 para US$ 29,5 bilhões em 2013, ou seja, um crescimento de quase cinco vezes. A empresa estima que até 80% dos aplicativos baixados em celulares sejam de jogos.[[3]](#footnote-3)

Dados do instituto mostram ainda que os celulares com acesso à internet somam 19% dos 325 milhões de aparelhos vendidos no planeta no terceiro trimestre deste ano. É um crescimento em ritmo vertiginoso: trata-se de um aumento de 50% em relação ao mesmo período de 2009.

Com todo esse avanço e o grande lucro previsto para as áreas de pizzarias e aplicativos foi pensado numa solução para reunir os dois setores em avanço com um aplicativo que gerencie as compras feitas pelos usuários pelos smartphones.

* 1. **REFERENCIAL TEÓRICO**

Conforme Tranfield, Denyer e Smart (2003) esta etapa do projeto é importante, porque apresenta uma breve discussão teórica do problema que esta relatando o projeto, na perspectiva de fundamentá-lo nas teorias existentes. As teorias e teóricos utilizados na pesquisa devem estar alinhados com a linha de pensamento e de investigação da pesquisa. A fundamentação teórica apresentada deve, ainda, servir de base para a análise e interpretação dos dados coletados.

1. **A evolução dos dispositivos móveis**

De acordo com Santana Costa R (2015) a evolução da computação móvel passou e passa por muitas etapas e mudanças podem citar Oersted que em 1820 descobre experimentalmente que a corrente elétrica produz um campo magnético e dali em diante a computação móvel foi evoluindo através dos anos, como o telégrafo, na metade do século XIX possibilitava transferências de palavras a longas distâncias pelo código Morse.

Os dispositivos móveis permitem registrar momentos, obter e disseminar informações em apenas segundos de qualquer lugar do mundo e também vem se tornando uma ferramenta indispensável para o trabalho devido ao grande arsenal de aplicativos, que estão à disposição para usar e servir de apoio a tarefas operacionais.

Podemos citar alguns modelos de dispositivos móveis que comprovam essa evolução nos últimos anos:

1. **Motorola DynaTac**

O website Terra4 aponta como o primeiro celular criado pela *Motorola* no ano de 1974 tinha 13 polegadas, pesava 794 gramas e custava US$ 4 mil.

Obteve um grande sucesso a partir de 1984, quando começou a ser comercializado, após 10 anos de pesquisa e obteve o apelido de "The Brick" (O Tijolo).[[4]](#footnote-4)

1. **J-Sho4**

Segundo o website Terra a *Sharp* também produziu celular, mas foi em 2001 que a *Sharp* apresentou com novidade o primeiro celular com câmera que havia chegado ao mercado. A câmera possui 110 mil *pixels* e tela colorida com 256 cores em seu aparelho.

1. ***iPhone 5***

Segundo o website Terra o *iPhone* 5 é sexta geração do smartphone da *Apple* lançado em 2012e segundo o website Terra ele foi descontinuado em setembro de 2013, após o lançamento do *iPhone* 5c em 2013, mas com a marca de celular mais vendido da historia em sua estreia.

É nítida a evolução dos dispositivos móveis em vários perímetros como em tamanho, espessura, cor, velocidade é indiscutível as melhorias.

Segundo Santana Costa R (2015, pg.7) atualmente, no Japão, foram realizadas as primeiras provas com sucesso, usando a tecnologia 4G (alcançando 100 Mbps a 200km/h). Estima-se lançar comercialmente os primeiros serviços de 4G no ano 2010, com velocidades entre 100 Mbps em movimento e 5 Gbps em repouso.

Com toda evolução tecnológica que a humanidade alcançou, há diversos projetos bem elaborados, com novas ideias e novos paradigmas. Conforme houver barateamento dos dispositivos e surgimentos de novas soluções para problemas reais, a computação móvel continuará se desenvolvendo numa velocidade extraordinária.[[5]](#footnote-5)

1. **Plataforma *Android***

Confome o website UFMG4 sistema Android tem como principal patrocinador a empresa Google. Contudo, ele também possui a participação de várias outras companhias de TI, tais como Intel, Nvidia, eBay, Qualcomm, Sony e Motorola. Essas empresas formaram, em 2007, uma associação para o desenvolvimento de alternativas para a computação móvel, denominada Open Handset Alliance.

Além de contar com a participação dessas empresas, o código que forma o sistema operacional Android também tem recebido contribuições de programadores espalhados por todo o mundo, uma vez que ele é software aberto.

De fato, Android foi construído em torno do *kernel* de Linux, o sistema operacional que provavelmente melhor representa o empenho e a capacidade da comunidade de software livre.[[6]](#footnote-6)

Aplicações em Android executam sobre uma máquina virtual chamada *Dalvik* e máquinas virtuais emulam, via uma camada de software, as funcionalidades normalmente encontradas em *hardware*, garantindo assim a portabilidade e a segurança das aplicações. De um ponto de vista conceitual, *Dalvik* assemelha-se à JVM, a máquina virtual usada para executar programas escritos na linguagem de programação Java.

As aplicações para Android, aliás, são programadas em Java. Em outras palavras, o desenvolvedor escreve aplicações em Java, e essas são, posteriormente, traduzidas para um formato binário que *Dalvik* é capaz de emular.

A escolha de Java como a linguagem base para o desenvolvimento no mundo Android deve-se a diversos fatores, como a sintaxe voltada para a orientação a objetos, as facilidades para controle de memória, e, sobretudo a grande popularidade dessa linguagem de programação. Java tem sido nos últimos dez anos a linguagem mais popular de acordo com o indicador Tiobe6.

### FUNDAMENTAÇÕES TEÓRICAS

**2.1 DESENVOLVIMENTO**

Segundo Sommerville (2011), o mundo moderno não poderia existir sem o software. Infraestruturas e serviços nacionais são controlados por sistemas computacionais, e a maioria dos produtos elétricos inclui um computador e um software que o controla. A manufatura e a distribuição industriais são totalmente informatizadas, assim como o sistema financeiro. A área de entretenimento, incluindo a indústria da música, jogos de computador, cinema e televisão, faz uso intensivo de software. Portanto, a engenharia de software é essencial para o funcionamento de sociedades nacionais e internacionais. Os sistemas de software são abstratos e intangíveis. Eles não são restringidos pelas propriedades dos materiais, nem governados pelas leis da física ou pelos processos de manufatura.

Inúmeras pessoas escrevem programas. Pessoas envolvidas com negócios escrevem programas em planilhas para simplificar seu trabalho; cientistas e engenheiros escrevem programas para processar seus dados experimentais; e há aqueles que escrevem programas como hobby, para seu próprio interesse e diversão.

No entanto, a maior parte do desenvolvimento de software é uma atividade profissional, em que o software é desenvolvido para um propósito específico de negócio, para inclusão em outros dispositivos ou como produtos de software como sistemas de informação, sistemas CAD etc. O software profissional, o que é usado por alguém além do seu desenvolvedor, é normalmente criado por equipes, em vez de indivíduos. Ele é mantido e alterado durante sua vida.

A engenharia de software tem por objetivo apoiar o desenvolvimento profissional de software, mais do que a programação individual. Ela inclui técnicas que apoiam especificação, projeto e evolução de programas, que normalmente não são relevantes para o desenvolvimento de software pessoal.

Muitas pessoas pensam que software é simplesmente outra palavra para programas de computador.

No entanto, quando falamos de engenharia de software, não se trata apenas do programa em si, mas de toda a documentação associada e dados de configurações necessários para fazer esse programa operar corretamente.

Um sistema de software desenvolvido profissionalmente é, com frequência, mais do que apenas um programa; ele normalmente consiste em uma série de programas separados e arquivos de configuração que são usados para configurar esses programas. Isso pode incluir documentação do sistema, que descreve a sua estrutura; documentação do usuário, que explica como usar o sistema; e sites, para usuários baixarem a informação recente do produto.

1. **METODOLOGIA USADA**

Segundo Beck (2004) *Extreme Programming* (XP) é uma metodologia ágil para pequenas e medias equipes que desenvolvem softwares baseado em requisitos vagos e que se modificam rapidamente. Dentre as principais diferenças da XP em relação às outras metodologias estão:

1. *Feedback* constante;
2. Abordagem incremental;
3. Encorajamento de comunicação face a face.

O XP é uma abordagem deliberada e disciplinada para o desenvolvimento de *software*, que vem sendo bastante utilizada e vem tomando espaço que antes pertencia a metodologias tradicionais, como RUP – *Rational Unified Process*.

Você codifica porque se você não codificar você não terá nada. Você testa porque se você não testar você não saberá quando você terminou de codificar. Você ouve porque se você não ouvir você não saberá o que codificar ou o que testar. E você projeta para que você possa codificar testar e ouvir indefinitivamente (KENT BECK, 2004, p. 61).

O XP (*Extreme Programming*) possuí quatro atividades básicas:

Codificar; testar; ouvir; projetar.

E estas são trabalhadas em práticas:

1. Entregas frequentes: idealizando o prazo de entrega de aproximadamente um a dois meses para cada versão do software e trabalhando juntamente com o constante *feedback* do cliente para evitar surpresas na entrega do *software* final.
2. Metáfora: crie uma historia simples sobre como o sistema funciona com a finalidade de passar para o projeto o que realmente o cliente espera.
3. Projeto simples: código mais simples possível contemplando somente os requisitos atuais sem se preocupar com os requisitos futuros. Estes somente serão adicionados ao sistema quando realmente for necessário.
4. Testes: a XP focaliza a validação do projeto durante todo o processo de desenvolvimento. Os programadores desenvolvem o software criando primeiramente os testes.
5. Refatoração: sempre que houver a oportunidade de simplificar o código, remover duplicidade e acrescentar flexibilidade ao sistema os programadores devem reestruturá-lo, mas mantendo sempre seu comportamento inicial.
6. Cliente presente: considere o cliente um membro da equipe disponível a qualquer hora, para que este tire duvidas em relação aos requisitos a serem implantados.
7. Programação em pares: a codificação em feita por duplas. Desta forma o código é sempre revisto por dois desenvolvedores diminuindo erros sintáticos e focando na melhoria do código. Estes dois devem sempre alternar suas funções.
8. Propriedade coletiva: qualquer membro da equipe pode modificar qualquer parte do código, desde que este passe pelos testes necessários.
9. Integração contínua: sempre que uma tarefa for finalizada, integre e atualiza a versão do sistema.
10. Padrões de codificação: os códigos deverão respeitar padrões definidos pela equipe.
    1. **REQUISITOS**

Segundo Sommerville (2011) os requisitos funcionais de um sistema descrevem o que ele deve fazer e dependem do tipo de software a ser desenvolvido. Quando expressos como requisitos de usuário, os requisitos funcionais são normalmente descritos de forma abstrata, para serem compreendidos pelos usuários do sistema. No entanto, requisitos de sistema funcionais mais específicos descrevem em detalhes as funções do sistema, suas entradas e saídas, exceções etc.

Os requisitos não funcionais, como o nome sugere, são requisitos que não estão diretamente relacionados com os serviços específicos oferecidos pelo sistema a seus usuários.

Na realidade, a distinção entre diferentes tipos de requisitos não é tão clara como sugerem essas definições simples. Um requisito de usuário relacionado com a proteção, tal como uma declaração de limitação de acesso a usuários autorizados, pode parecer um requisito não funcional. No entanto, quando desenvolvido em mais detalhes, esse requisito pode gerar outros requisitos, claramente funcionais, como a necessidade de incluir recursos de autenticação de usuário no sistema.

Isso mostra que os requisitos não são independentes e que muitas vezes geram ou restringem outros requisitos. Portanto, os requisitos de sistema não apenas especificam os serviços ou as características necessárias ao sistema, mas também a funcionalidade necessária para garantir que esses serviços/características sejam entregues corretamente.

* 1. **LEVANTAMENTO DE REQUÍSITOS**

Segundo Kendall (1992) o levantamento de requisitos consiste em uma solicitação do usuário ou cliente para que algo seja mudado ou executado no projeto. Para apoiar este trabalho, diversas técnicas podem ser utilizadas como: Brainstorming, workshops, entrevistas entre outras.

Neste projeto foi utilizada a técnica de entrevista, uma entrevista de levantamento de informações é uma conversa direcionada com um propósito específico, que utiliza um formato “pergunta-resposta”. Os objetivos de uma entrevista incluem:

1. Obter as opiniões do entrevistado, o que ajuda na descoberta dos problemas-chave a serem tratados;
2. Conhecer os sentimentos do entrevistado sobre o estado corrente do sistema;
3. Obter metas organizacionais e pessoais;
4. Levantar procedimentos informais.

Fatos obtidos em uma investigação podem explicar o desempenho passado.

Uma entrevista envolve as seguintes etapas como principais: planejamento, condução e elaboração de um relatório da entrevista e são importantes para se determinar metas onde o engenheiro de software está, provavelmente, estabelecendo um relacionamento com uma pessoa estranha a ele. Assim, é importante que ele:

1. Construa, rapidamente, uma base de confiança e entendimento;
2. Mantenha o controle da entrevista;
3. Venda a “ideia do sistema”, provendo ao entrevistado as informações necessárias.
   1. **VALIDAÇÃO DE REQUISITOS**

Sommerville (2011) relata que a validação de requisitos é o processo pelo qual se verifica se os requisitos definem o sistema que o cliente realmente deseja. Ela se sobrepõe à análise, uma vez que está preocupada em encontrar problemas com os requisitos. A validação de requisitos é importante porque erros em um documento de requisitos podem gerar altos custos de retrabalho quando descobertos durante o desenvolvimento ou após o sistema já estar em serviço.

O custo para consertar um problema de requisitos por meio de uma mudança no sistema é geralmente muito maior do que o custo de consertar erros de projeto ou de codificação. A razão para isso é que a ocorrência de mudança dos requisitos normalmente significa que o projeto e a implantação do sistema também devem ser alterados. Além disso, o sistema deve, posteriormente, ser retestado.

* 1. **MODELAGEM DE SISTEMAS**

De acordo com Sommerville (2011) a modelagem de sistema é o processo de desenvolvimento de modelos abstratos de um sistema, em que cada modelo apresenta uma visão ou perspectiva, diferente do sistema.

A modelagem de sistema geralmente representa o sistema com algum tipo de notação gráfica, que, atualmente, quase sempre é baseada em notações de UML (linguagem de modelagem unificada, do inglês *Unified Modeling Language*). No entanto, também é possível desenvolver modelos (matemáticos) formais de um sistema, normalmente como uma especificação detalhada do sistema.

Os modelos são usados durante o processo de engenharia de requisitos para ajudar a extrair os requisitos do sistema; durante o processo de projeto, são usados para descrever o sistema para os engenheiros que o implantam; e, após isso, são usados para documentar a estrutura e a operação do sistema.

Podem-se desenvolver modelos do sistema existente e do sistema a ser desenvolvido: [[7]](#footnote-7)

1. Modelos do sistema existente são usados durante a engenharia de requisitos. Eles ajudam a esclarecer o que o sistema existente faz e podem ser usados como ponto de partida para discutir seus pontos fortes e fracos. Levam, então, os requisitos para o novo sistema.
2. Modelos do novo sistema são usados durante a engenharia de requisitos para ajudar a explicar os requisitos propostos para outros *Stakeholders* do sistema. Os engenheiros usam esses modelos para discutir propostas de projeto e documentar o sistema para a implantação. Em um processo de engenharia dirigida a modelos, é possível gerar uma implantação completa ou parcial do sistema a partir do modelo de sistema.
   1. **DIAGRAMAÇÃO UML**

De acordo com o website *Docs Kde7* (2016) a *Unified Modelling Language* (UML) é uma linguagem de diagramas para especificar, visualizar e documentar modelos de 'software' orientados por objetos. O UML não é um método de desenvolvimento, o que significa que não lhe diz o que fazer primeiro ou o que fazer depois ou como desenhar o seu sistema, mas ajuda-o a visualizar o seu desenho e a comunicar com os outros. O UML é controlado pelo *Object* *Management* *Group* (OMG) e é a norma da indústria para descrever graficamente o software.

Conforme Sommerville (2003) avalia a UML como sendo um padrão para modelagem orientada a objetos, e aponta a visão dos casos de uso como sendo uma fonte de obtenção de requisitos importantíssima.

A UML não inclui uma notação específica para a modelagem de banco de dados, pois supõe um processo de desenvolvimento orientado a objetos e modelam dados usando objetos e seus relacionamentos. No entanto, pode-se usar a UML para representar um modelo semântico de dados, sob um modelo semântico de dados, podemos pensar em entidades como classes de objetos simplificados (não possuem operações), em atributos como atributos da classe de objetos e em relações como as associações nomeadas entre classes de objetos.

1. **DIAGRAMA DE CASO DE USO**

De acordo Pressman (2011, pg. 731) o diagrama de caso de uso ajuda a determinar as funcionalidades e características do software sob o ponto de vista do usuário.

Um caso de uso descreve como um usuário interage com o sistema definindo passos necessários para atingir um objetivo especifico

[...] Casos de uso descrevem as coisas que os atores querem que o sistema faça [...] e deve ser uma tarefa completa, segundo a perspectiva do ator. [...] O conjunto de todos os casos de uso irá descrever a funcionalidade completa do sistema sob o ponto de vista dos usuários.

(SCHNEIDER; WINTERS, 2001, p. 14).

1. **DIAGRAMA DE CLASSE**

Conforme Sommerville (2003) os diagramas de classe são usados no desenvolvimento de um modelo de sistema orientado a objetos para mostrar as classes de um sistema e as associações entre essas classes, em poucas palavras, uma classe de objeto pode ser pensada como uma definição geral de um tipo de objeto do sistema. Uma associação é um link entre classes que indica algum relacionamento entre essas classes.

Os diagramas de classe em UML podem ser expressos em diferentes níveis de detalhamento. Quando você está desenvolvendo um modelo, o primeiro estágio geralmente é o de olhar para o mundo, identificar os objetos essenciais e representá-los como classes. A maneira mais simples de fazer isso é escrever o nome da classe em uma caixa ou pode simplesmente observar a existência de uma associação, traçando uma linha entre as classes.

1. **DIAGRAMA DE ATIVIDADES**

Segundo Pressman (2011) o diagrama de atividades mostra o comportamento dinâmico de um sistema ou parte do sistema através de controle entre ações que o sistema executa, é similar a um fluxograma exceto que pode mostrar fluxos concorrentes.

O componente principal de um diagrama de atividades é um nó ação, representado por um retângulo arredondado, que corresponde a uma tarefa executada pelo sistema ou software. Setas que vão de um nó de ação a outro indicam o fluxo de controle, isto é, uma seta entre dois nós de ação significa que depois que a primeira ação é efetuada a segunda pode começar. Um ponto preto sólido forma o nó inicial que representa o ponto inicial de atividade. Um ponto preto envolvido por um circulo preto é o nó final indicando o fim da atividade.

* 1. **MODELO DE RELACIONAMENTO DAS TABELAS**

Segundo Heuser (1998) técnica de modelagem de dados mais utilizada é a abordagem Modelo Entidade-Relacionamento (MER) devido a sua simplicidade e eficiência. Consiste em uma coleção de objetos básicos, chamados entidades e relacionamentos entre esses objetos. É um modelo de dados de alto-nível criado com o objetivo de representar a semântica associada aos dados do minimundo.

O MER é utilizado para na fase de projeto conceitual, onde o esquema conceitual do banco de dados da aplicação é concebido. Seus conceitos são intuitivos, permitindo que projetistas de banco de dado capturem os conceitos associados aos dados da aplicação, sem a interferência da tecnologia específica de incrementarão do banco de dados.

* 1. **DICIONÁRIO DE DADOS**

De acordo com o website luis.br8 no processo de análise de sistemas um dos pontos fortes é o MER – Modelo de Entidade e Relacionamento, onde são definidas as entidades que irão compor o sistema e como elas irão relacionar-se.

Junto com o modelo de entidade e relacionamento (MER), é necessário que se mantenha um documento com a explicação de todos os elementos nele criados. Este documento, que pode ser chamado de dicionário de dados, permite que os analistas obtenham informações sobre todos os objetos do modelo de forma textual, contendo explicações por vezes difíceis de incluir no diagrama. [[8]](#footnote-8)

Onde teremos os seguintes tipos de dados.

1. Atributo: Os atributos são os nomes e as características da entidade
2. Classe: as classes podem ser: simples, composto, multivalorado e determinante. Simples indica um atributo normal. Composto indica que ele poderá ser dividido em outros atributos, como por exemplo, o endereço. Multivalorado é quando o valor do atributo poderá não ser único e determinante é um atributo que será usado como chave, como CPF, Código do cliente, etc.
3. Domínio: podem ser numérico, texto, data e booleano. Podemos chamar também de tipo do valor que o atributo irá receber. A definição desses tipos deve seguir um processo lógico, exemplo: nome é texto, salário é numérico, data de nascimento é data e assim por diante.
4. Tamanho: define a quantidade de caracteres que serão necessários para armazenar o seu conteúdo.
5. Descrição: é opcional e pode ser usado para descrever o que é aquele atributo ou dar informações adicionais que possam ser usadas futuramente pelo analista ou programador do sistema.
   1. **FERRAMENTAS DE DESENVOLVIMENTO**

A aplicação executada utiliza dispositivos móveis para plataforma *Google Android*. O software, disponibilizado em *Tablet* ou smartphone, desenvolvido na linguagem de programação *HTML5, CSS3, JavaScript* e *PHP, utiliza* as seguintes tecnologias:

1. ***PHP*** *(Hypertext Preprocesso)* O site oficial PHP descreve que é uma linguagem de programação de ampla utilização, interpretada, que é especialmente interessante para desenvolvimento para a web e pode ser mesclada dentro do código HTML. O objetivo principal da linguagem é permitir a desenvolvedores escreverem páginas que serão geradas dinamicamente rapidamente, mas você pode fazer muito mais do que isso com *PHP*.
2. **Intel XDK** é um programa cuja proposta é facilitar a criação de aplicativos em *HTML5*, oferecendo uma ferramenta simples para ajudar os desenvolvedores em sua árdua tarefa. Depois de instaladas as ferramentas básicas do Intel XDK, ele abre uma janela do seu navegador-padrão, a partir da qual você interage com a interface para criar aplicativos. Como criar os aplicativos usando *HTML5* (cinco), qualquer plataforma que suporte o formato é capaz de rodá-los. Pensando nisso, você tem a alternativa de substituir o quadro exibido ao redor das opções do aplicativo, podendo escolher entre molduras de sistemas com *iOS*, *Android* e computadores (Tecmundo, 2015)
3. [**XAMPP**](http://www.techtudo.com.br/downloads/xampp) segundo o website Techtudo é um pacote com os principais servidores de código aberto do mercado, incluindo FTP, banco de dados MySQL e Apache com suporte as linguagens PHP e Perl que são interligados e configurados automaticamente

### 3 ESTUDO DE CASO

O objetivo deste software é proporcionar ao cliente uma facilidade de fazer *download* do aplicativo em seu *smartphone*, onde deverá:

1. Efetuar um cadastro de dados pessoais para gerar seu *login.*
2. Opção de cardápios de pizzas.
3. Poderá fazer as solicitações dos tipos variados de pizzas pequenas, médias e grandes, logo após finalizar o pedido.
4. O funcionário poderá emitir um relatório dos pedidos realizados, e pizzas mais vendidos, com isso terá um gerenciamento maior dos seus clientes, tamanhos e sabores de cada pizza solicitada.
5. Consequentemente a empresa se tornará mais competitiva, organizada e com um maior controle de suas informações no sistema.
6. Aumentar os lucros do estabelecimento.
7. Propor uma automação das tarefas do estabelecimento.

**[[9]](#footnote-9)**

* 1. **MATRIZ DE PAPEIS E RESPONSABILIDADES E CRONOGRAMA**

Segundo o site do PMO o Escritoriodeprojetos9 a Matriz de Responsabilidades, ou matriz de alocação de responsabilidades (do inglês - *Responsibility Assignment* *Matrix* - RAM), é uma matriz que aloca a responsabilidade de cada integrante da equipe do projeto (recursos humanos) sobre cada entrega e seus respectivos pacotes de trabalho.

A Matriz de Responsabilidades também pode ser conhecida como RACI onde as letras representam a responsabilidade do integrante da tarefa.

Onde:

**R**: Responsável pela execução / **A**: Responsável pela aprovação

**C**: Recurso deve ser consultado / **I**: Recurso deve ser informado

Quadro 1: Matriz de Responsabilidades resumida com as tarefas e seus respectivos responsáveis

|  |  |
| --- | --- |
| **Tarefa** | **Responsáveis** |
| Análise e planejamento de Projeto | Lucas Macedo Astério de Sousa;  Lemys Michel Souza Nascimento |
| Verso Inicial do Projeto, Levantamento de Requisitos funcionais e não funcionais. | Lemys Michel Souza Nascimento;  Lucas Macedo Astério de Sousa |
| Criação do Caso de Uso. | Lucas Macedo Astério de Sousa |
| Correção do Caso de Uso. | Lemys Michel Souza Nascimento |
| Criação do diagrama de classes | Lucas Macedo Astério de Sousa |
| Finalização dos Fluxos Principais, Alternativos e dos casos de uso | Lucas Macedo Astério de Sousa |
| Correção e analise do projeto | Lemys Michel Souza Nascimento |
| Criação do diagrama de classes | Lucas Macedo Astério de Sousa |
| Correção diagrama de classes | Lemys Michel Souza Nascimento |
| Criação diagrama de objetos | Lucas Macedo Astério de Sousa |
| Correção diagrama de objetos | Lemys Michel Souza Nascimento |
| Criação diagramas de Atividade | Lucas Macedo Astério de Sousa |
| Correção diagramas de Atividade | Lemys Michel Souza Nascimento |
| Melhorias e documentação do projeto | Lucas Macedo Astério de Sousa;  Lemys Michel Souza Nascimento |
| Criação da Matriz de Responsabilidade | Lucas Macedo Astério de Sousa |
| Ajustes finais do projeto e da criação dos protótipos. | Lucas Macedo Astério de Sousa;  Lemys Michel Souza Nascimento |
| Conclusão da documentação | Lemys Michel Souza Nascimento |

Pode observar o cronograma do projeto com as datas de inicio e fim de cada tarefa o cronograma do projeto documenta as atividades do projeto, suas respectivas datas de início e de término, além dos recursos usados. Observa-se abaixo a Matriz de Responsabilidades com cronograma onde teremos os responsáveis de cada tarefa e datas (Matriz de Responsabilidades resumida, matriz completa no anexo 1).

* 1. **CUSTOS DO PROJETO**

Segundo o site do PMO o Escritoriodeprojetos9 o total dos custos de um projeto é a somatória de todos os recursos necessários para executar as atividades previstas no projeto expressos em unidade monetária. Eles formam as saídas do fluxo de caixa e são usadas juntas com os ganhos obtidos pelo projeto (que são as entradas do fluxo de caixa) para formar o fluxo de caixa e determinar a viabilidade do projeto.[[10]](#footnote-10)

Observa-se abaixo a tabela de custos com todos os recursos utilizados no projeto:

**Tabela 1 – Custo total do projeto - Próprios autores, 2015**

|  |  |
| --- | --- |
| **Recursos** | **Custo** |
| Notebook | R$ 1.500,00 |
| Computador (Desktop) | R$ 1.250,00 |
| Lucas Macedo Astério de Sousa | RS 35,00/hr (100) |
| Lemys Michel Souza Nascimento | RS 35,00/hr (100) |
|  |  |
| **Custo total do projeto** | **R$ 25.840,00** |

* 1. **ANÁLISE DA VIABILIDADE**

Foi realizada uma pesquisa qualitativa utilizando a ferramenta do *Google* o *Google* *Forms10* e sua divulgação feita no *Facebook11* obteve resultados satisfatórios, foi feito uma analise de viabilidade do projeto que testava a viabilidade e aceitação dos usuários para aplicativo.[[11]](#footnote-11)

**Tabela 2 - Números totais da pesquisa de viabilidade - Próprios autores, 2015**

|  |  |
| --- | --- |
| **Total** | **647** |
| **Sim** | **496** |
| **Não** | **94** |
| **Talvez** | **31** |
| **Não sei** | **26** |

Foram recebidas 647 respostas e teve uma aceitação de 496 respostas, o publico que não aceitou foi de 94 respostas, com todos esses dados foi criado um gráfico de analise, e constatando que a criação desse aplicativo é aceita pelos usuários.

Fonte: próprios autores (2015)

**Figura 1: Gráfico de Viabilidade**

* 1. **REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS**

1. **Requisitos Funcionais:**

RF01: Permitir cadastro de dados de clientes.

RF02: Permitir cadastro de dados de funcionários (administrador via WEB).

RF03: Permitir cadastro de Pizzas (via *WEB*).

RF04: Escolher tipo de pizza e verificar pedido concluído no carrinho eletrônico.

RF05: O usuário terá um carrinho de compras para poder manter seu pedido antes de efetuar a compra.

###### **Requisitos não funcionais:**

RF01: Sistema será implantado em *HTML5, JAVASCRIPT e PH*P na plataforma Intel XDK para *smartphones* *Android*.

RF02: Sistema estará disponível de acordo com o expediente da Pizzaria.

RF03: Deverá conter um sistema de controle Web para manter (Excluir, Consultar, Inserir, Alterar) as pizzas e os usuários do aplicativo.

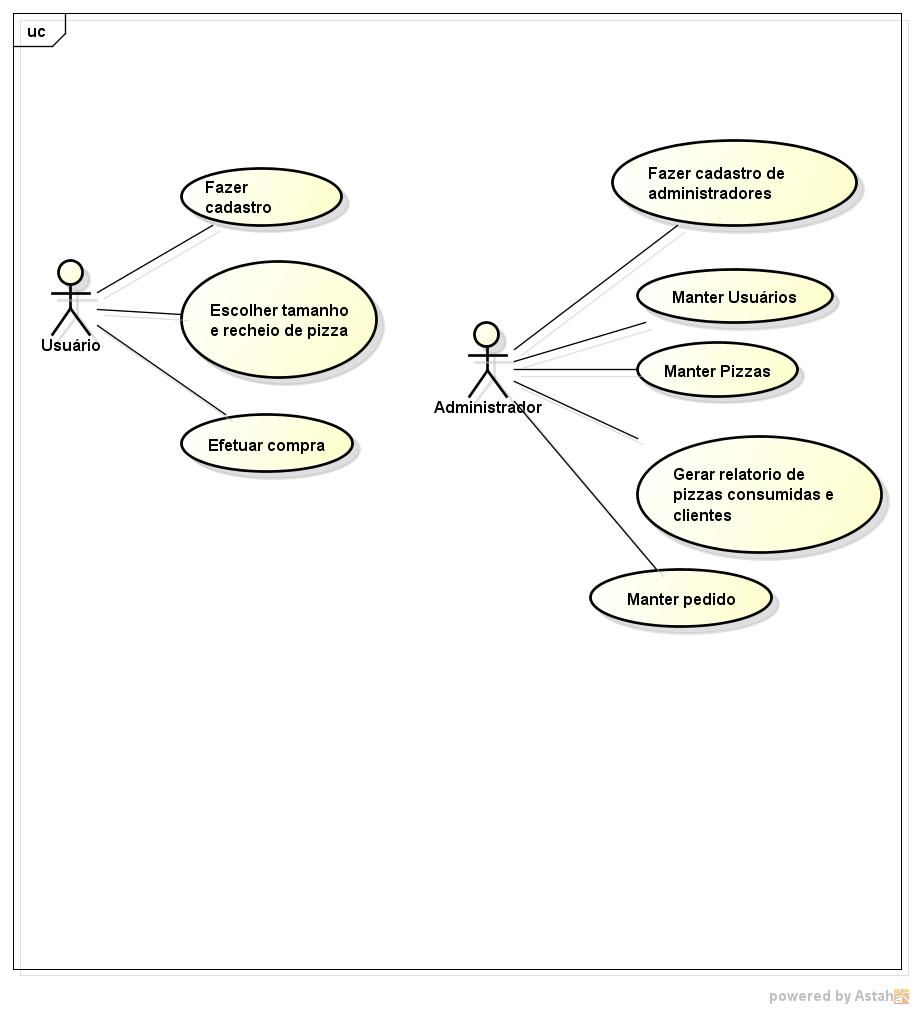
* 1. **DETERMINAÇÃO DAS FUNCIONALIDADES DO SISTEMA**

O aplicativo permite aos seus usuários efetuarem compras de pizzas online, de tamanhos diferentes (Pequeno, Médio e Grande) e sabores diferentes dependendo da pizzaria e suas características.

Para acesso ao aplicativo o usuário deve efetuar o download e logo após efetuar o cadastro e fazer sua compra.

* 1. **DIAGRAMA DE CASO DE USO**

Para a validação com os usuários, será apresentado o diagrama de caso de uso com as funcionalidades do projeto:

****

Fonte: próprios autores (2015)

Figura 2: Caso de Uso

* 1. **CENÁRIOS DO CASO DE USO**

Apresentaremos abaixo a descrição das funcionalidades de cada caso de uso do diagrama de caso de uso.

3.7.1 Descrição do caso de uso escolher tamanho e recheio de pizza:

Quadro 2: Cenário para cada caso de uso.

|  |  |
| --- | --- |
| NOME: | Escolher tamanho e recheio de pizza |
| ATOR: | Usuário |
| DESCRIÇÃO: | Escolha de tamanho de pizzas (Grande, Media e Pequena). |
| PRÉ-CONDIÇÃO: | Ter login de acesso |
| PÓS-CONDIÇÃO: | Escolher o sabor de pizza |
| FLUXO PRINCIPAL: | |  |  | | --- | --- | | Escolher tamanho e recheio de pizza | | | Sistema: | Retorna a tela inicial de escolha de tamanho de pizzas (Grande, Media e Pequena) | | Usuário: | Realiza sua escolha | | Sistema: | Recebi a escolha e o redireciona para pagina de sabores | |
| FLUXO ALTERNATIVO: | |  |  | | --- | --- | | Escolher tamanho e recheio de pizza | | | Sistema: | Retorna a tela inicial de escolha de tamanho de pizzas (Grande, Media e Pequena) | | Usuário: | Não realiza sua escolha e sai do seu login | | Sistema: | Redireciona para pagina de login do Aplicativo | |

3.7.2 Descrição do caso de uso efetuar compra:

|  |  |
| --- | --- |
| NOME: | Efetuar compra |
| ATOR: | Usuário |
| DESCRIÇÃO: | Escolha de sabores e confirmação de pedido |
| PRÉ-CONDIÇÃO: | Ter escolhido o tamanho da pizza |
| PÓS-CONDIÇÃO: | Efetuar pagamento |
| FLUXO PRINCIPAL: | |  |  | | --- | --- | | Efetuar compra | | | Sistema: | Retorna a escolha de sabores de pizzas | | Usuário: | Realiza sua escolha | | Sistema: | E redireciona para pagina de pagamento | |
| FLUXO ALTERNATIVO: | |  |  | | --- | --- | | Efetuar compra | | | Sistema: | Retorna a escolha de sabores de pizzas | | Usuário: | Não realiza sua escolha | | Sistema: | Redireciona para pagina de escolha de tamanho de pizzas | |

3.7.3 Descrição do caso de uso manter pizzas:

|  |  |
| --- | --- |
| NOME: | Manter Pizzas, Cadastros de usuários |
| DESCRIÇÃO: | Interface que consulta, insere, altera e exclui Cadastro, Pizzas do sistema. |
| ATOR: | Administrador |
| PRÉ-CONDIÇÃO: | Executa o controle de acesso. |
| PÓS-CONDIÇÃO: | Manter dados de Cadastro, Pizzas com sucesso. |
| OBSERVAÇÂO: | Apenas será possível fazer tais ações na Web. |
| FLUXO PRINCIPAL: | |  |  | | --- | --- | | Consultar Cadastro, Pizzas | | | Usuário: | Executa menu Cadastro de Pizzas | | Sistema: | Cadastrar Pizzas com os campos padrões | | Usuário: | Preenche os campos necessários e executa o botão consultar, por padrão o campo Código. | |
| FLUXO ALTERNATIVO: | |  |  | | --- | --- | | Inserir Cadastro, Pizzas | | | Usuário: | Executa o botão Inserir. | | Sistema: | Cadastrar Pizzas com os campos de Cadastro e Pizzas | | Usuário: | Preenche os campos necessários e executa o botão Salvar. | | Sistema: | Valida os dados e Fluxo encerrado. |   ––––––––––––––––––––––––––––––––––––––––––––––––––––––––   |  |  | | --- | --- | | Alterar Cadastro, Pizzas | | | Usuário: | Seleciona um item na lista e executa o botão Alterar. | | Sistema: | Cadastrar Pizzas com os campos desejados | | Usuário: | Modifica os campos e executa o botão Salvar. | | Sistema: | Valida os dados e Fluxo encerrado. |   –––––––––––––––––––––––––––––––––––––––––––––––––––––––   |  |  | | --- | --- | | Excluir Cadastro, Pizzas | | | Usuário: | Seleciona um item na lista e executa o botão Excluir. | | Sistema: | Excluir Cadastro, Pizzas com os campos: | | Usuário: | Executa o botão Confirmar. | | Sistema: | Valida o pedido e Fluxo encerrado. | |

3.7.4 Descrição do caso de uso manter pedido:

|  |  |
| --- | --- |
| NOME: | Manter pedidos |
| DESCRIÇÃO: | Tela que consulta, insere, altera e exclui Cadastro, Pizzas do sistema. |
| ATOR: | Administrador |
| PRÉ-CONDIÇÃO: | Executa o controle de acesso de acordo com a necessidade |
| PÓS-CONDIÇÃO: | Manter dados de pedidos com sucesso. |
| OBSERVAÇÂO: | Apenas será possível fazer tais ações na Web. |
| FLUXO PRINCIPAL: | |  |  | | --- | --- | | Consultar pedidos | | | Usuário: | Executa menu Cadastro, Pizzas | | Sistema: | Consultar pedidos com os campos padrões | | Usuário: | Preenche os campos necessários e executa o botão consultar, por padrão o campo Código. | |
| FLUXO ALTERNATIVO: | |  |  | | --- | --- | | Inserir pedidos | | | Usuário: | Executa o botão Inserir | | Sistema: | Inserir pedidos com os campos obrigatórios de pedido | | Usuário: | Preenche os campos necessários e executa o botão Salva | | Sistema: | Valida os dados e Fluxo encerrado |   ––––––––––––––––––––––––––––––––––––––––––––––––––––––––   |  |  | | --- | --- | | Alterar pedidos | | | Usuário: | Seleciona um item na lista e executa o botão Alterar. | | Sistema: | Alterar pedidos com os campos desejados | | Usuário: | Modifica os campos e executa o botão Salvar | | Sistema: | Valida os dados e Fluxo encerrado |  |  |  | | --- | --- | | Excluir pedidos | | | Usuário: | Seleciona um item na lista e executa o botão Excluir | | Sistema: | Excluir pedidos com os campos | | Usuário: | Executa o botão Confirmar | | Sistema: | Valida o pedido e Fluxo encerrado | |

3.7.5 Descrição do caso de uso manter usuários:

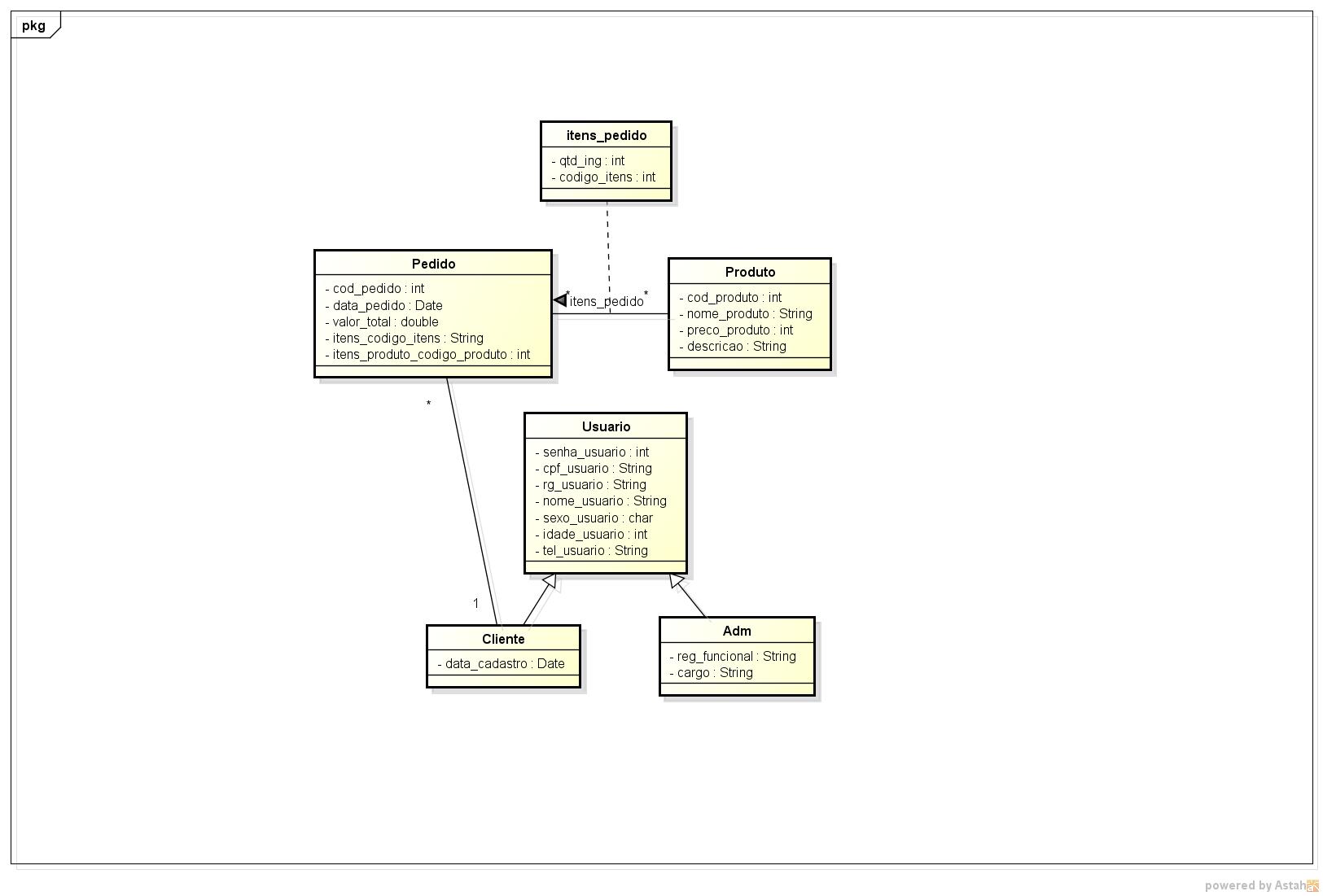
|  |  |
| --- | --- |
| NOME: | Manter usuários |
| DESCRIÇÃO: | Interface que consulta, insere, altera e exclui Cadastro, Pizzas do sistema. |
| ATOR: | Administrador |
| PRÉ-CONDIÇÃO: | Executa o controle de acesso. |
| PÓS-CONDIÇÃO: | Manter dados de Cadastro, Pizzas com sucesso. |
| OBSERVAÇÂO: | Apenas será possível fazer tais ações na Web. |
| FLUXO PRINCIPAL: | |  |  | | --- | --- | | Consultar Cadastro de usuários | | | Usuário: | Executa menu Cadastro de usuários | | Sistema: | Cadastrar usuários com os campos padrões | | Usuário: | Preenche os campos e executa o botão consultar, por padrão o campo Código. | |
| FLUXO ALTERNATIVO: | |  |  | | --- | --- | | Inserir Cadastro de usuários | | | Usuário: | Executa o botão Inserir. | | Sistema: | Cadastrar usuários com os campos de Cadastro | | Usuário: | Inseri os todos os campos e executa o botão Salvar. | | Sistema: | Valida os dados e Fluxo encerrado. |   ––––––––––––––––––––––––––––––––––––––––––––––––––––––––   |  |  | | --- | --- | | Alterar Cadastro de usuários | | | Usuário: | Seleciona um item na lista e executa o botão Alterar | | Sistema: | Cadastrar Pizzas com os campos desejados | | Usuário: | Modifica os campos e executa o botão Salvar | | Sistema: | Valida os dados e Fluxo encerrado |   –––––––––––––––––––––––––––––––––––––––––––––––––––––––   |  |  | | --- | --- | | Excluir Cadastro de usuários | | | Usuário: | Seleciona um item na lista e executa o botão Excluir. | | Sistema: | Excluir usuários | | Usuário: | Executa o botão Confirmar. | | Sistema: | Valida o pedido e Fluxo encerrado. | |

3.7.5 Descrição do caso de uso gerar relatório de pizzas consumidas e clientes:

|  |  |
| --- | --- |
| NOME: | Gerar relatórios de pizzas consumidas e clientes |
| DESCRIÇÃO: | Consulta, insere, altera e exclui produtos do sistema. |
| PRÉ-CONDIÇÃO: | O administrador realizara relatórios (Usuários, Compras) |
| PÓS-CONDIÇÃO: | Manter dados de produtos com sucesso. |
| FLUXO PRINCIPAL: | |  |  | | --- | --- | | Gerar relatórios de pizzas consumidas e clientes | | | Usuário: | Executa o botão de pizzas consumidas | | Sistema: | Relatório de pizzas consumidas é exibido | | Sistema: | Volta para tela de menu de ações gerencias | |
| FLUXO(S) ALTERNATIVO(S): | |  |  | | --- | --- | | Gerar relatórios de pizzas consumidas e clientes | | | Usuário: | Executa o botão de clientes | | Sistema: | Relatório de Usuários é exibido | | Usuário: | Volta para tela de menu de ações gerencias | |

* 1. **DIAGRAMA DE CLASSES**

Apresentaremos abaixo a descrição das relações de cada tabela do diagrama de caso de uso.

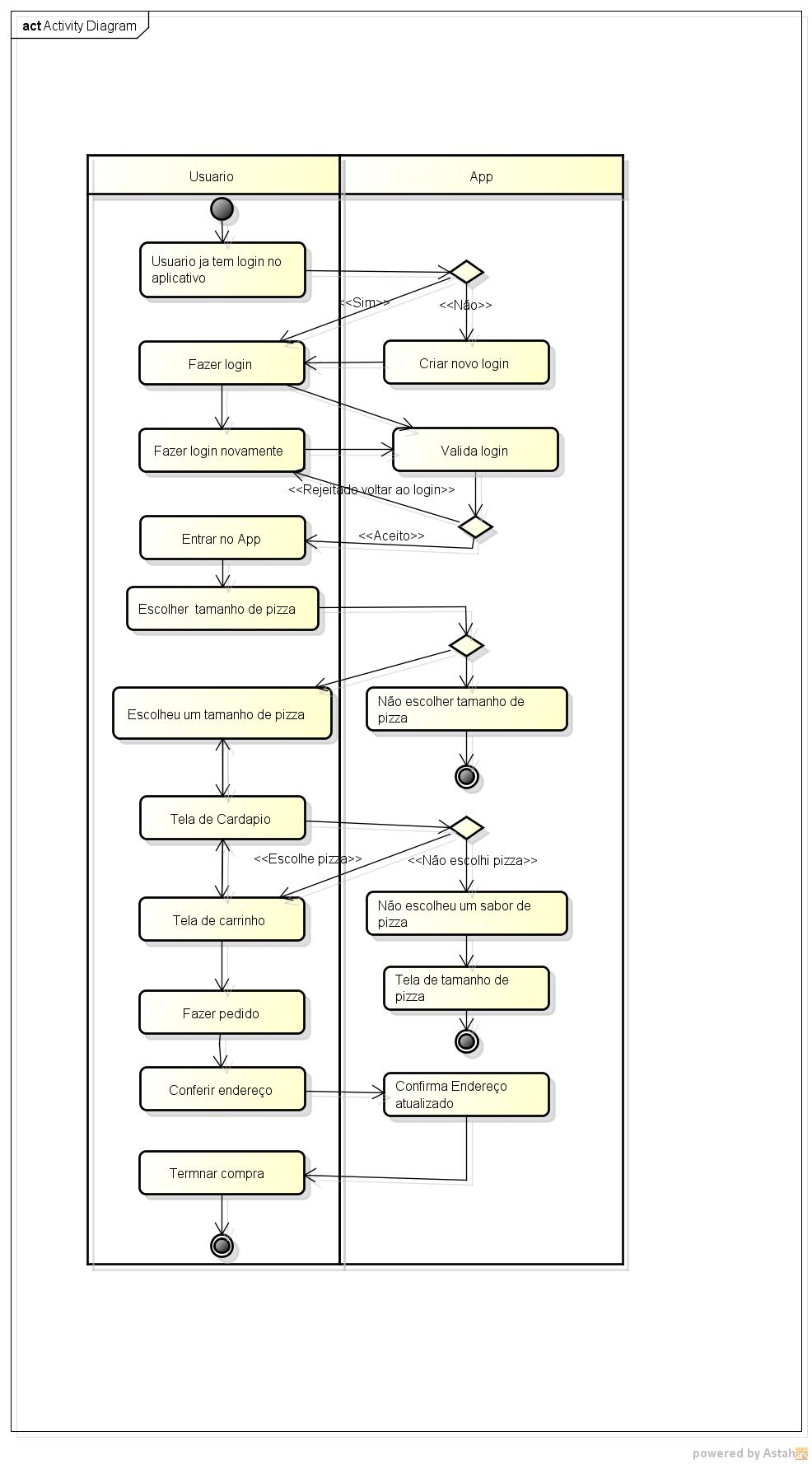


Fonte: próprios autores (2015)

Figura 3: Diagrama de Classe

* 1. **DIAGRAMA DE ATIVIDADES**

Apresentaremos abaixo a descrição das atividades do aplicativo no diagrama de atividades.

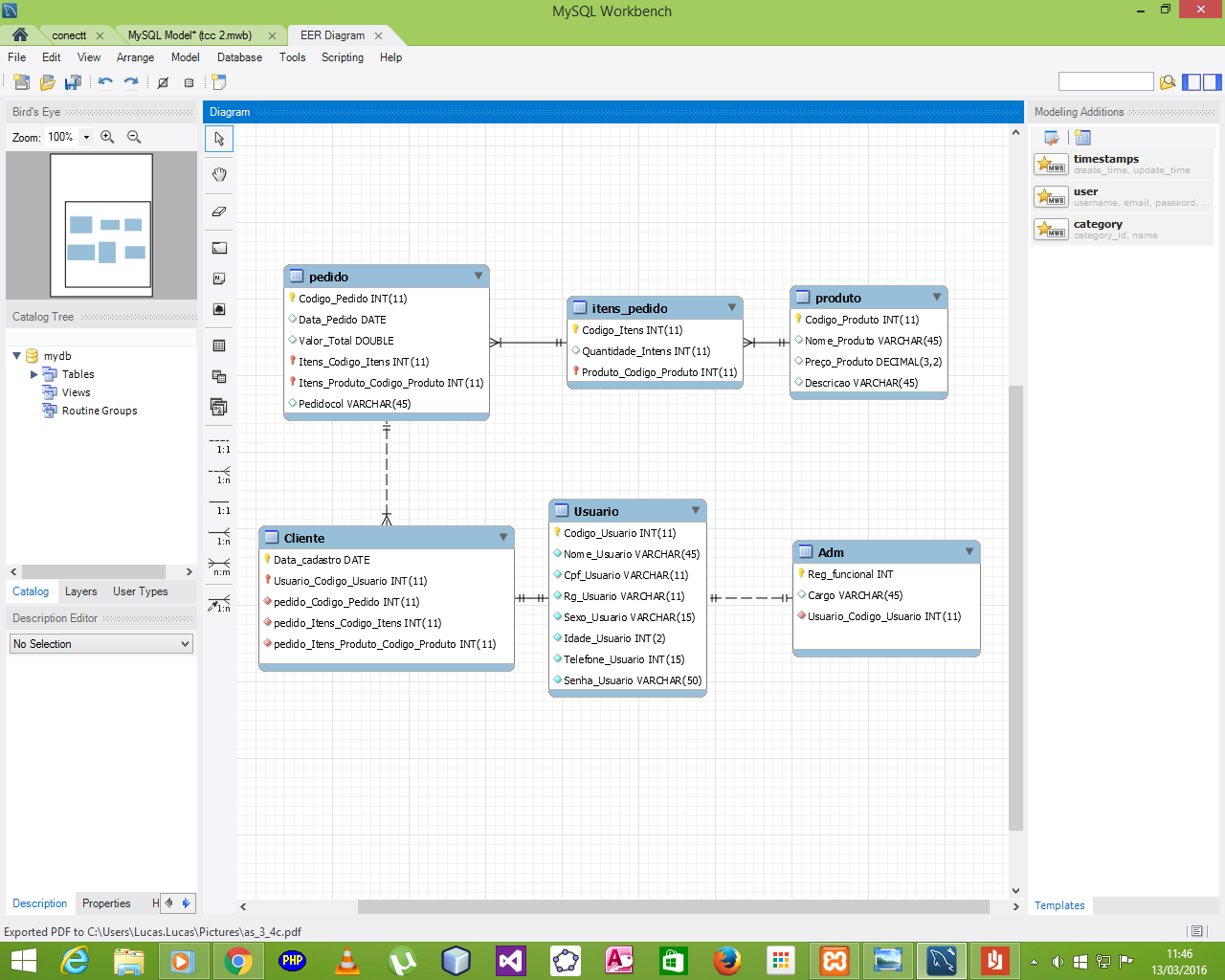


Fonte: próprios autores (2015)

Figura 4: Diagrama de Atividades

* 1. **CRIAÇÃO DE DIAGRAMAS DE RELACIONAMENTO DAS TABELAS**

Apresentaremos abaixo a descrição das relações de cada tabela do diagrama de relacionamento.



Fonte: próprios autores (2015)

Figura 5: Modelo de Diagrama de Entidade Relaciona

* 1. **DICIONÁRIO DE DADOS**

Segundo o website UCB12 (2016) dicionário de dados consiste numa lista organizada de todos os elementos de dados que são pertinentes para o sistema e apresentaremos os seguintes elementos do projeto.[[12]](#footnote-12)

**Tabela 3 – Dicionário de Dados - Próprios autores, 2015**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atributo | Classe | Domínio | Tamanho | Descrição |
| nome\_usuário | Simples | Texto | 50 | Até 100 caracteres |
| rg\_usuário | Determinante | Numérico |  | Valor sem traço |
| cpf\_usuário | Determinante | Numérico |  | Valor sem traço |
| sexo\_usuário | Simples | Texto |  | Homem ou Mulher |
| idade\_usuário | Simples | Numérico |  |  |
| endereco\_usuário | Composto | Texto | 100 | Até 100 caracteres |
| tel\_usuário | Multivalorado | Numérico |  | Exemplo (11 97955 3279) |
| senha\_usuário | Determinante | Texto |  | Sem caracteres especiais |
| nome\_adm | Simples | Texto | 50 | Até 100 caracteres |
| rg\_adm | Determinante | Numérico |  | Valor sem traço |
| cpf\_adm | Determinante | Numérico |  | Valor sem traço |
| sexo\_adm | Simples | Texto |  | Homem ou Mulher |
| idade\_adm | Simples | Numérico |  |  |
| endereco\_adm | Composto | Texto | 100 | Até 100 caracteres |
| tel\_adm | Multivalorado | Numérico |  | Exemplo (11 97955 3279) |
| senha\_adm | Determinante | Texto |  | Sem caracteres especiais |
| cod\_pedido | Determinante | Numérico |  | Código de pedido |
| data\_pedido | Simples | Data |  | Dd/mm/aa |
| valor\_total | Simples | Numérico |  |  |
| produto\_codigo | Simples | Numérico |  |  |
| codigo\_itens | Determinante | Numérico |  |  |
| cod\_produto | Determinante | Numérico |  | Código de produto |
| nome\_produto | Simples | Texto | 50 |  |
| preco\_produto | Simples | Numérico |  |  |
| descricao\_produto | Simples | Texto | 100 |  |
| Reg\_funcional | Determinante | Numérico |  | Código do relatório |
| cargo | Simples | Texto | 50 |  |
| data\_cadastro | Simples | Data |  | Dd/mm/aa |

**3.12 LAYOUT DAS TELAS E RELATÓRIOS**

Os layouts a seguir foram desenvolvidos no software Intel XDK utilizado para interfaces com HTML5.

**Tela inicial do aplicativo:** Tela onde os usuários iram inserir o CPF e Senha para logarem no aplicativo, caso não tenha login clique em “Cadastro” (os dados são obrigatorios).



Fonte: próprios autores (2016)

Figura 6: Modelo Protótipo Mobile

Abaixo é descrito as principais funcionalidades e os botões da tela inicial do aplicativo.

**Quadro 3: Quadro de funcionalidades.**

|  |  |
| --- | --- |
| **Ação** | **Botões** |
| 1. Usuário ira digitar seu login ou se cadastrar | 1. Ok – Usuário ira efetur login 2. Cadastro – Clicando aqui ira fazer um novo cadastro |

**Tela de cadastro:** Tela onde são inseridos os dados para cadastro para logo após poderem efetuar pedidos e visualizar o cardápio de pizzas disponíveis. Todos os dados são obrigatórios.



Fonte: próprios autores (2016)

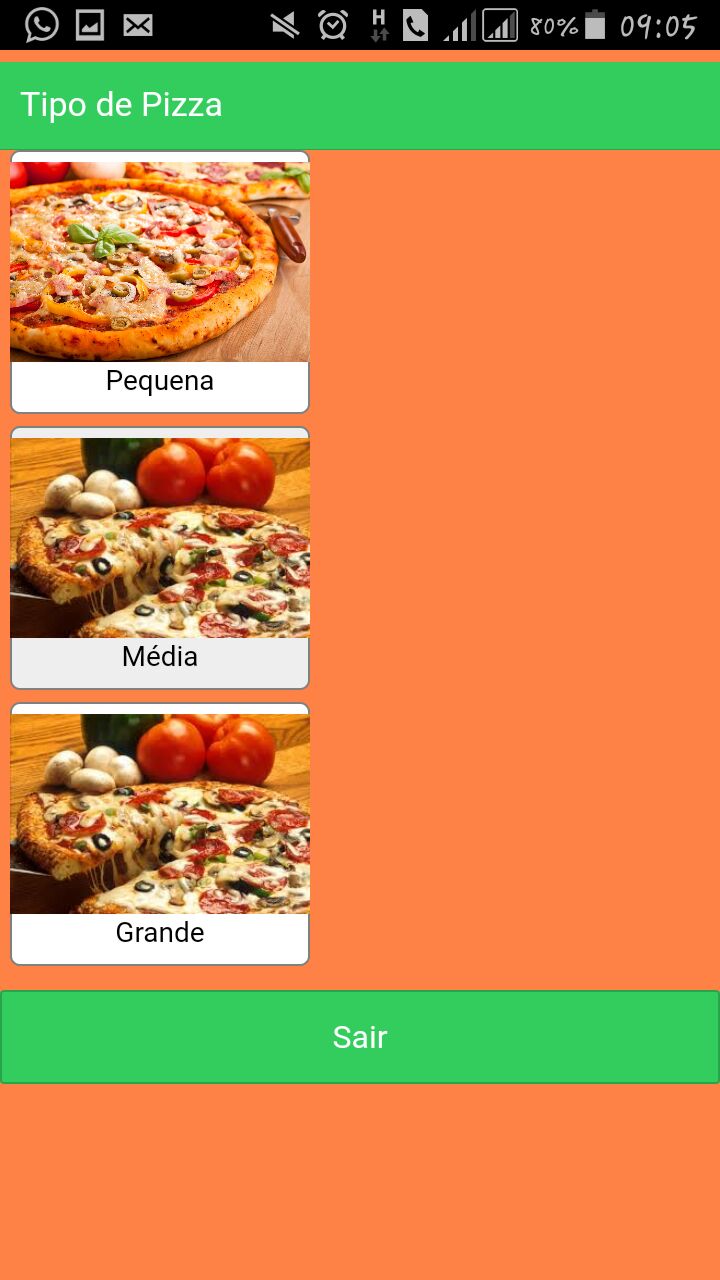
Figura 7: Modelo Protótipo Mobile- Tela de cadastro

Abaixo é descrito as principais funcionalidades e os botões da tela de cadastro aplicativo.

**Quadro 4: Quadro de funcionalidades.**

|  |  |
| --- | --- |
| **Ação** | **Botões** |
| 1. Usuário ira digitar seus dados para ser efetuado o cadastro | 1. Ok – Para confirmar os dados e efetuar cadastro |

**Tela de escolha de tamanho de pizza:** Tela onde os usuários escolheram o tamanho de sua pizza entre Pequena, Média e Grande.

****

Fonte: próprios autores (2015)

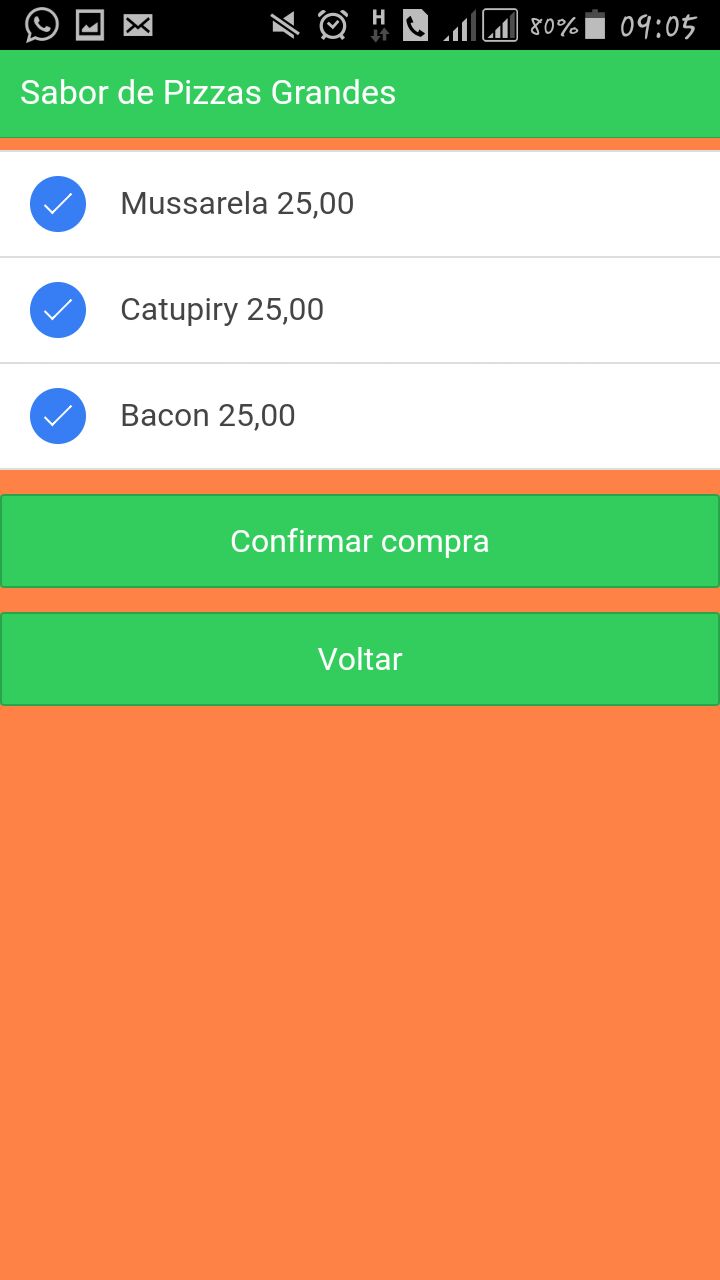
Figura 8: Modelo Protótipo Mobile- Tela de Tipo de Pizza

Abaixo é descrito as principais funcionalidades e os botões da tela de escolha de tamanho de pizza aplicativo.

**Quadro 5: Quadro de funcionalidades.**

|  |  |
| --- | --- |
| **Ação** | **Botões** |
| 1. Usuário ira escolher o tamanho de sua pizza | 1. Sair – Efetuar Logoff do aplicativo |

**Tela de cardápio de compra:** Nesta tela serão exibidos todos os sabores relativos a cada tamanho que foi escolhido anteriormente e com seus valores correspondentes, podendo escolher quantas pizzas e quais tamanhos quiserem.

****

Fonte: próprios autores (2015)

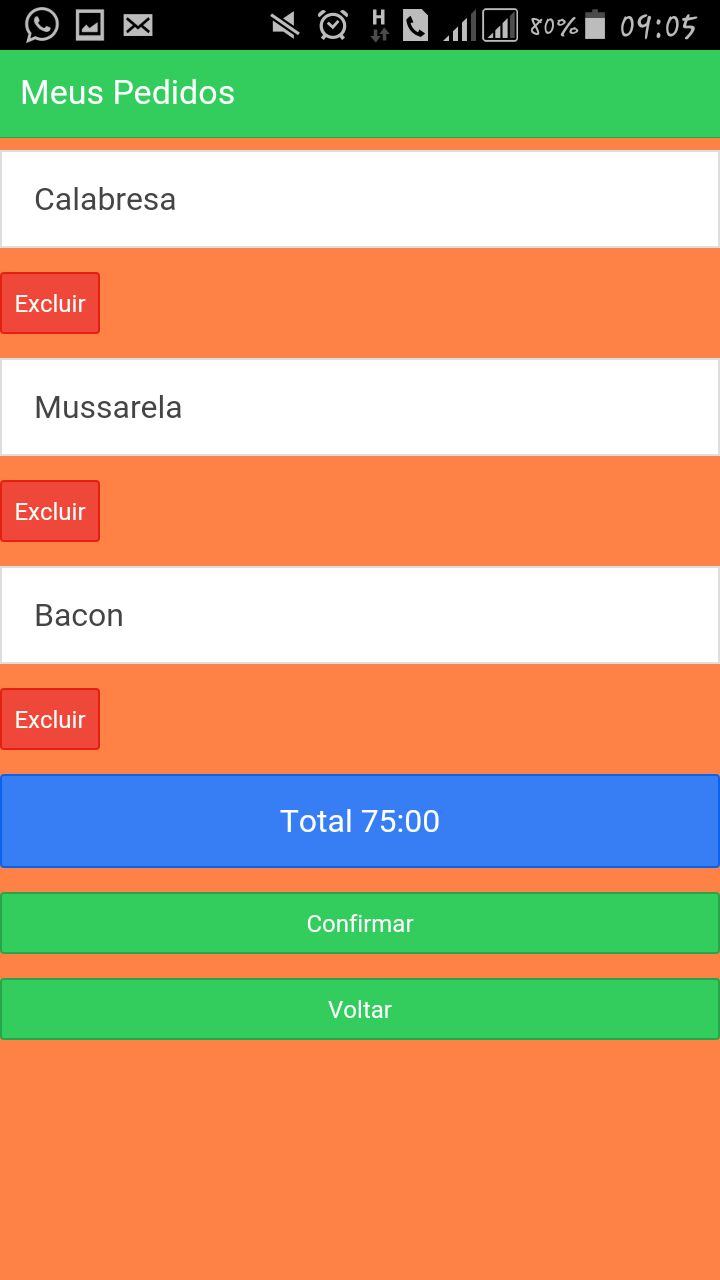
Figura 9: Modelo Protótipo Mobile- Tela de pedidos realizados

Abaixo é descrito as principais funcionalidades e os botões da tela de cardápio de compra do aplicativo.

**Quadro 6: Quadro de funcionalidades.**

|  |  |
| --- | --- |
| **Ação** | **Botões** |
| 1. Usuário ira escolher os sabores para fazer o pedido | 1. Confirmar compra– Confirmar pizzas marcadas e que vão ser inseridas no carrinho 2. Voltar – para pagina de tamanho de pizzas |

**Tela de carrinho de compras:** Nesta etapa o usuário ira manter sua compra, podendo alterar quantidades e tipos de pizza, já contendo sua somatória em baixo.

****

Fonte: próprios autores (2015)

Figura 10: Modelo Protótipo Mobile- Tela de Confirmação de compra

Abaixo é descrito as principais funcionalidades e os botões da tela de carrinho de compra do aplicativo.

**Quadro 7: Quadro de funcionalidades.**

|  |  |
| --- | --- |
| **Ação** | **Botões** |
| 1. Usuário ira manter seus pedidos de acordo com a sua vontade | 1. Excluir – Usuário poderá excluir as pizzas escolhidas anteriormente 2. Confirmar – Usuário ira confirmar pedido 3. Voltar – O usuário poderá voltar e escolher mais pizzas dos tamanho desejado. |

**Tela de confirmação de endereço de entrega:** Para confirmação e maior segurança o aplicativo ira mostrar o endereço inserido pelo usuário em seu cadastro para que ele possa confirmar ou alterar seu endereço de entrega.

****

Fonte: próprios autores (2015)

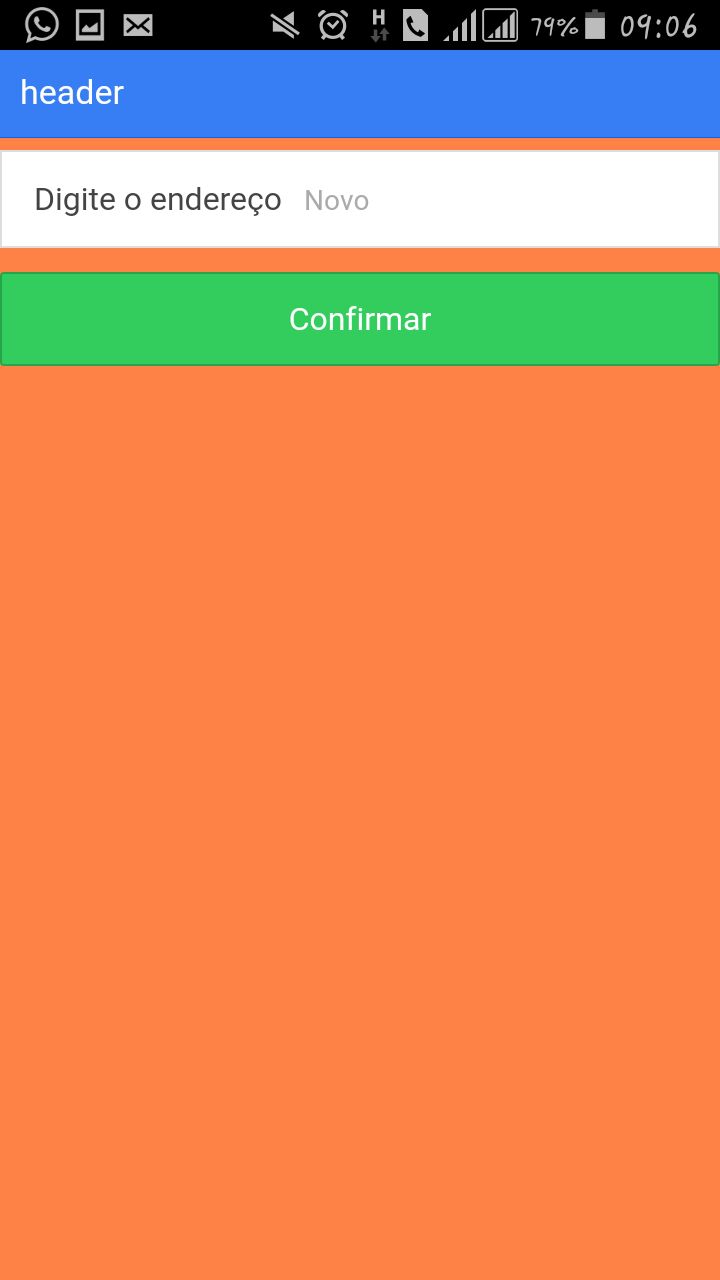
Figura11: Modelo Protótipo Mobile- Tela de confirmação de endereço

Abaixo é descrito as principais funcionalidades e os botões da tela de confirmação de endereço de entrega.

**Quadro 8: Quadro de funcionalidades.**

|  |  |
| --- | --- |
| **Ação** | **Botões** |
| 1. Usuário confirmar ou mudar o endereço de entrega | 1. Confirmar compra – Usuário aceita o endereço atual e será redireciona a tela final do aplicativo confirmando seu pedido 2. Mudar endereço – Usuário não aceita o endereço atual e digitará um novo |

**Tela de mudança de endereço de entrega:** Caso o usuário deseja alterar seu endereço já inserido no momento do cadastro é possível nesta tela.

****

Fonte: próprios autores (2015)

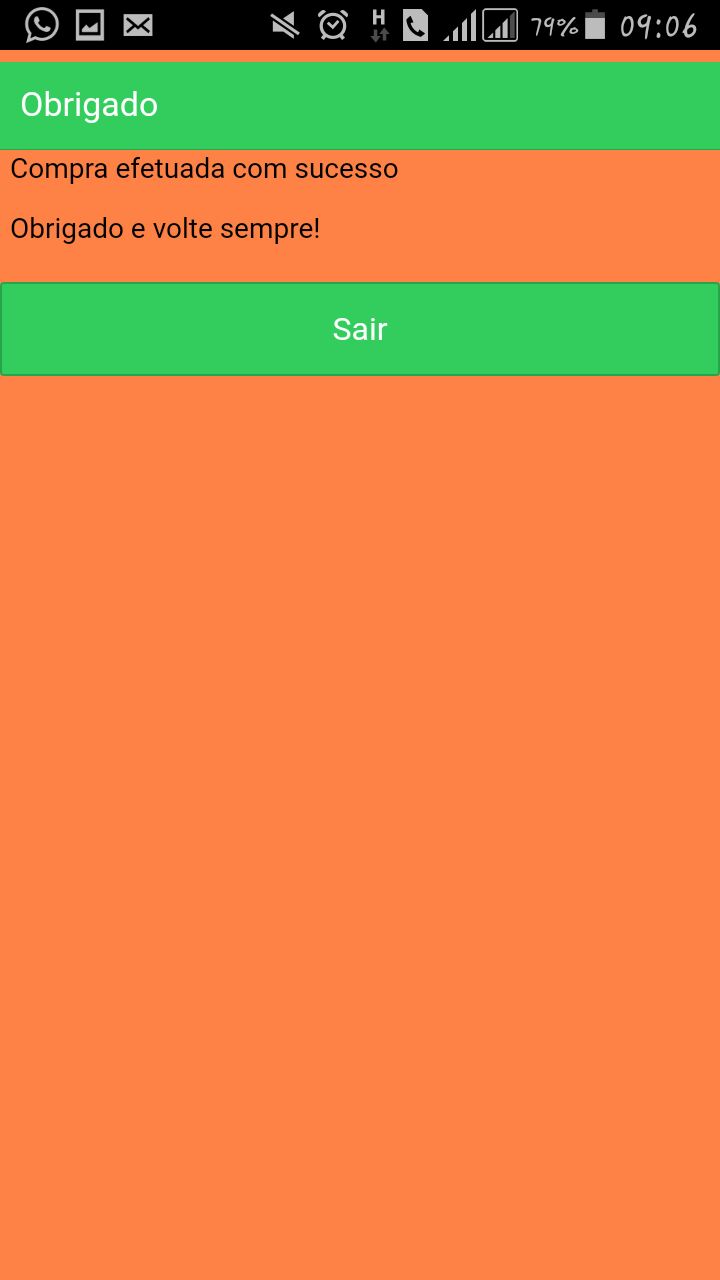
Figura 12: Modelo Protótipo Mobile- Tela de alteração de endereço

Abaixo é descrito as principais funcionalidades e os botões da tela de mudança de endereço de entrega.

**Quadro 9: Quadro de funcionalidades.**

|  |  |
| --- | --- |
| **Ação** | **Botões** |
| 1. Usuário ira digitar seu novo endereço e confirmar seu pedido | 1. Confirmar – Usuário ira confirmar seu endereço e será redireciona a tela final do aplicativo |

**Tela de confirmação de pedido:** Tela final do aplicativo com a confirmação de compra

****

Fonte: próprios autores (2015)

Figura 4: Modelo Protótipo Mobile- Tela de Confirmação de Compra

Abaixo é descrito as principais funcionalidades e os botões da tela de confirmação de pedido.

**Quadro 10: Quadro de funcionalidades.**

|  |  |
| --- | --- |
| **Ação** | **Botões** |
| 1. O aplicativo ira indicar que seu pedido esta pronto | 1. Sair – Redireciona o usuário para tela inicial do aplicativo |

1. **CONCLUSÃO**

Este trabalho de conclusão de curso acrescentou de forma positiva na expansão do nosso conhecimento adquirindo uma visão de negócio diferente e também a complexidade no desenvolvimento de um projeto que vem atender um determinado nicho de mercado. Abriu conhecimento de um mercado promissor para desenvolvimento de tecnologias móveis que venham atender esse tipo de negócio.

Diversos obstáculos foram superados com o levantamento de requisitos para um aplicativo mais intuitivo e simples pensando na comodidade dos usuários. Obstáculos na elaboração de diagramas e pesquisas relacionadas às pizzarias do Brasil, mas com experiências como aulas de projetos, diagramação UML, logica de programação e muitas outras vivenciadas na faculdade que forneceu a base para ultrapassar os obstáculos

Para o leitor e usuário este projeto traz novas possibilidades de informatização no serviço de entrega de pizzas demonstrando como uma compra pode ser realizada através de um aplicativo de uma forma simples e sem “dores de cabeça” ocasionadas nas ligações.

Ao mundo acadêmico traz novas ideias, demonstrando de forma organizada e estruturada todas as informações para o desenvolvimento de um aplicativo mobile.

Para este projeto é pretendido melhorias futuras como inserir funcionalidades de localização por GPS para encontrar o cliente com mais facilidade e assim agilizando a entrega.

1. **REFERÊNCIA BIBLIOGRÁFICA**

BARCAUI, André B., **PMO - Escritórios de Projetos, Programas e Portfólio Na Prática,** São Paulo, Brasport,2012.

BECK, K. **Programação extrema (XP) explicada**: **acolha as mudanças**. Porto Alegre, Bookman, 2004.

DENYER, D; SMART, P.; TRANFIELD, D, **Towards a Methodology for Developing Evidence-Informed Management Knowledge Means of Systematic Review**, Bristol, British Journal of Management, 2003.

HEUSER, Carlos Alberto, **Projeto de banco de dados**, Porto Alegre, Bookman, 1998.

KENDALL, K.E., KENDALL, J.E.; **Systems Analysis and Design** 2ª ed, São Paulo, Prentice Hall, S1992.

MAIORINO, José Emilio, **Introdução aos Métodos da Matemática Aplicada**, São Paulo, UNICAMP, 2003.

PRESSMAN, Roger. **Engenharia de Software**. 6ª ed., São Paulo, Makron Books, 1995.

SANTANA, **Reinaldo Costa. Computação Móvel, Histórico da Evolução**, Porto Alegre, 2008.

SCHNEIDER, Geri; WINTERS, Jason P., **Applying Use Cases: A practical guide**, Boston, Addison-Wesley, 2001.

SOMMERVILLE, Ian. **Engenharia de Software**. 9ª ed. São Paulo, Person, 2011.

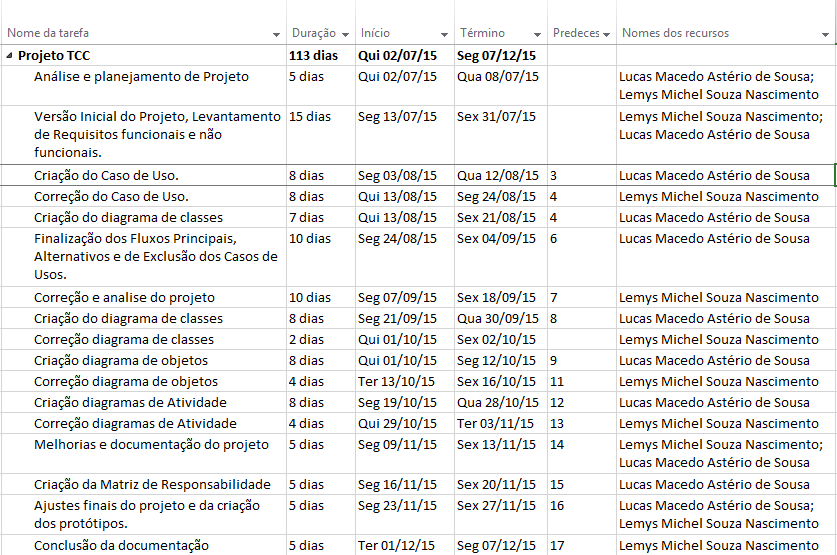
TAUHATA, Sergio, **O mercado de aplicativos para celular cresce sem parar,** São Paulo, Globo, 2010.

TOFFLER, Alvin, **War and Anti-War,** Nova Iorque, Warner Books, 1995.

1. **ANEXOS**

**ANEXO 1:**

**Tabela 4: Matriz de responsabilidade, Próprios autores, 2015.**



1. ¹ http://ecommercenews.com.br/artigos/cases/o-futuro-das-compras/ Acesso em14/01/2016 [↑](#footnote-ref-1)
2. miriangasparin.com.br/ Acesso em14/01/2016 [↑](#footnote-ref-2)
3. 3https://www.hellofood.com/ Acesso em11/01/2016 [↑](#footnote-ref-3)
4. 4 Terra, Celular, a historia do celular, 2016

   Disponível em: http://www.tecnologia.terra.com.br/celular/celulares/ [↑](#footnote-ref-4)
5. 5 http://homepages.dcc.ufmg.br/~fernando/classes/android.html/ Acesso em 16/01/2016 [↑](#footnote-ref-5)
6. 6 www.tiobe.com/ Acesso em 17/01/2016 [↑](#footnote-ref-6)
7. 7 https://docs.kde.org/trunk4/pt\_BR/kdesdk/umbrello/uml-basics.html/ Acesso em .18/01/2016 [↑](#footnote-ref-7)
8. 8 http://www.luis.blog.br/modelo-de-entidade-e-relacionamento-mer.aspx/ Acesso em 10/03/2016 [↑](#footnote-ref-8)
9. 9 http://escritoriodeprojetos.com.br/ Acesso em 02/02/2016 [↑](#footnote-ref-9)
10. 9 http://escritoriodeprojetos.com.br/ Acesso em 02/02/2016 [↑](#footnote-ref-10)
11. 10 https://www.google.com/forms/ Acesso em 07/02/2016

    https://www.facebbok.com/ Acesso em 07/02/2016 [↑](#footnote-ref-11)
12. 12 http://www,cae.ucb.br/programar/.banco2\_recursosAdministracaoBd/ Acesso em 02/02/2016 [↑](#footnote-ref-12)