

Aplicação Lista de Compras *Offline* em um Comércio Varejista de Pequeno Porte

Gilberto E. Tavares¹, Mário E. Augusto¹

¹Departamento de Sistemas de Informação
Universidade do Estado de Santa Catarina (UDESC) – São Bento do Sul, SC – Brazil

mario.augusto@udesc.br, gilbertoetavares@gmail.com

Abstract. *Even small companies that opt for some kind of system, many times just for legal obligations, don't realize the potential in automate processes. This article analyzes one this processes and propose, through the software development method, a mobile application accessible in most smartphones. This process is a simple shopping list, that beside solving some situations, perhaps come to be the starting point for some automations later. Because the printed list is used during the purchase, the offline access has been proposed and studied the existing technology for it.*

Resumo. *Mesmo pequenas empresas que optam por algum tipo de sistema, muitas vezes apenas por obrigações, não percebem o potencial em automatizar processos. Este artigo analisa um desses processos e propõe, através de método de desenvolvimento de software, uma aplicação móvel acessível pela maioria dos smartphones. Esse processo uma lista de compras simples além de resolver algumas situações, pode vir a ser ponto de partida para automatizações posteriores. Devido ao uso lista impressa no ato da compra, o acesso offline foi proposto e a tecnologia existente para tanto estudada.*

Introdução

Visando produtividade, eficiência e praticidade cada vez mais processos são automatizados, nos mais diversos ramos e portes empresariais variando desde as menores até as grandes companhias. Essas automatizações proporcionam o autosserviço, que possibilita em toda hierarquia da equipe a descentralização de funções e informações.

Níveis de acesso e aprovação podem ser definidos, seguindo a política da empresa. Isso garante controle operacional, de modo autônomo e flexível, que simplifica aos envolvidos, a participação neste processos informatizados. Em acordo com o apoiador de pequenos negócio [Sebrae 2015], alguns benefícios dessa automatização são:

- Redução de custo no treinamento desses processos;
- Possibilita execução com confiança e consistência;
- Torna ágil as atividades, com aperfeiçoamento e mudança gradual;
- Mantém diretrizes, por estar definido na implementação.

Na investigação desses processos pré-estabelecidos na empresa Alecrim Dourado Variedades, foi encontrado de modo oportuno a atividade visual e de registro, de todas mercadorias que devem ser obtidas no mercado atacadista. O processo manual, centralizado e sem controle apropriado, é realizado através da clássica planilha eletrônica. Esta

realidade propicia através do desenvolvimento de aplicação própria uma evolução tangível.

Durante a análise alguns pontos foram verificados como motivacionais à criação do sistema informacional proposto, tendo soluções relativas:

- Acesso centralizado da informação;
- Informação imediata;
- Registro de item com funcionário e data;
- Redução de retrabalho;
- Menor índice de duplicidade de itens;
- Possibilidade de não utilização de impressão em papel.

O objetivo neste artigo é o desenvolvimento de aplicação para gerir itens em lista de compras, também o acompanhamento durante a compra dos mesmos. Devendo este aplicativo ser acessível em dispositivos móveis, com devido tratamento em casos de indisponibilidade de conectividade.

Modelo Inicial

A empresa do estudo é classificada como varejista, comercializa produtos variados: papelaria, utilidades, ferramentas, vestuário, brinquedos, info-eletrônicos, decoração, doces e outros.

Utiliza-se um arquivo ‘pasta de planilha eletrônica’ do Microsoft Excel. A planilha utilizada é bem simples, como pôde ser visto na Figura 1, não há fórmulas ou macros, são apenas três colunas nas quais os itens são preenchidos conforme sua classificação. Essa classificação diz respeito ao local em que será comprada a mercadoria, pois alguns tipos são melhor encontrados em lojas especializadas. As categorias utilizadas são: Atacado, Info-eletrônicos e Bijuterias/Acessórios.

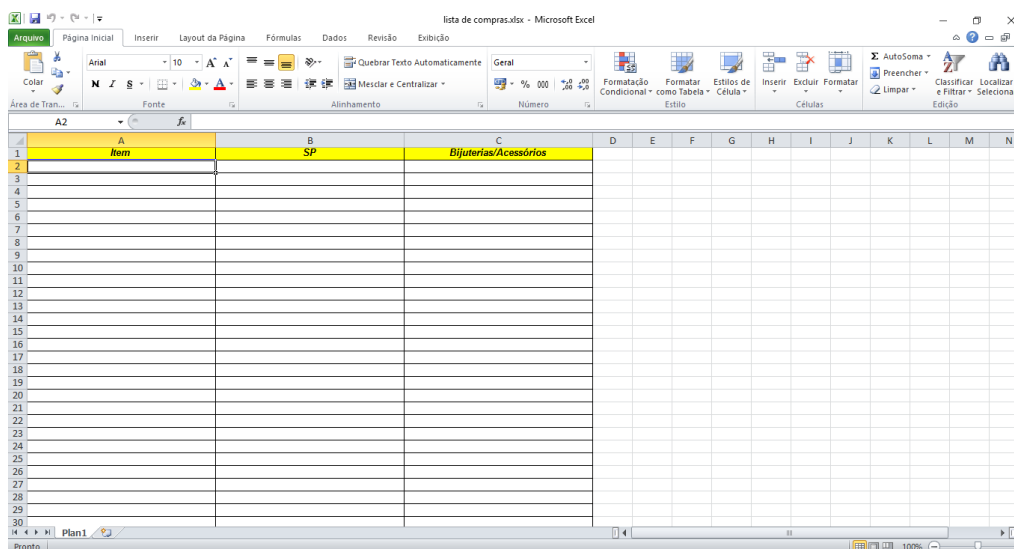


Figura 1. Planilha utilizada

O costumeiro é que cada funcionária tenha sempre consigo uma caderneta (no bolso do colete fornecido pela loja) e anote itens que são de sua seção assim que seja

Não é regra ou obrigatório anotar somente itens da seção a qual a funcionária é responsável, muito pelo contrário, prefere-se que seja assim pois antes “pecar” pelo excesso que pela falta, será pior um item não ter sido anotado do que estar repetido em listas diferentes. Pois na próxima etapa deve o digitador filtrar para que não permaneça itens duplicados, porém essa ação tem complicação por itens com nomes dúbios (descritos com termos diferentes) ou mesmo quando compostos por várias palavras podem estar em várias ordens ou abreviações. Quando no processo de compras esta lista é levada impressa conforme Figura 2.

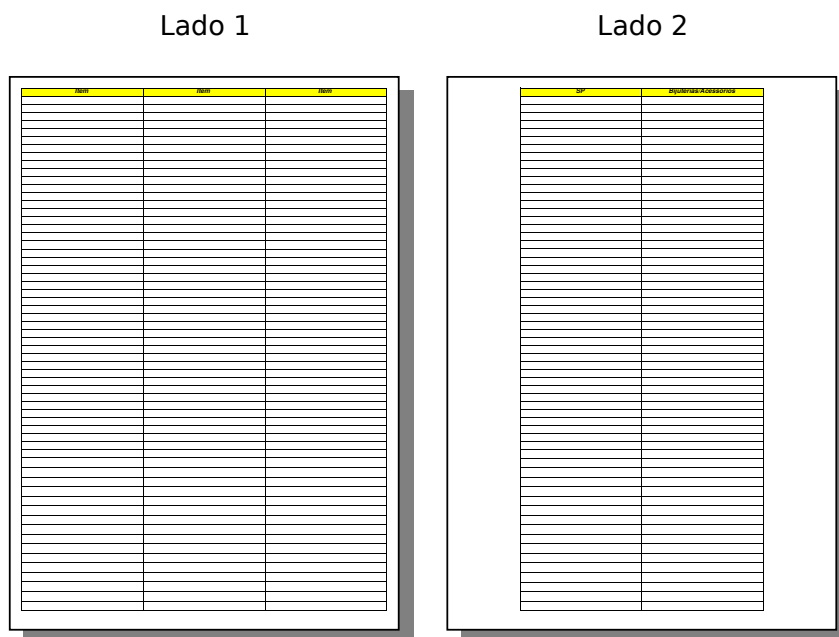


Figura 2. Modelo de Impressão da Planilha

Solução Proposta

O sistema é uma aplicação para gerenciar os itens de compras, onde o usuário pode gerir listas (adicionar, editar ou remover itens), além de ser possível por usuários atribuídos marcar itens como comprados ou encerrar as listas. O acesso deverá ter algum processo de autenticação de usuário

Como explicado por [Sommerville 2011], a **Engenharia de Requisitos** é o processo em que são acordados e especificados os detalhes que satisfazem os envolvidos. Há dois níveis de detalhamento dessa especificação, um para clientes e usuários outro aos desenvolvedores, alto nível e abrangente respectivamente.

Os **Requisitos Funcionais** descrevem o que o sistema deve fazer, normalmente são descritos de modo abstrato para a fácil compreensão por seus usuários [Sommerville 2011]. Puderam para o *software* ser levantados os requisitos funcionais que seguem:

1. [RF001] Realizar a adição de itens;
2. [RF002] Manter as listas de itens;
3. [RF003] Alternar o *status* de compra dos itens;
4. [RF004] Realizar encerramento das listas (manter itens não comprados);
5. [RF005] Realizar autenticação para permitir o acesso.

Já os **Requisitos Não funcionais** segundo [Sommerville 2011], são os relacionados não diretamente as funcionalidades oferecidas aos usuários pelo sistema. Descrevem esse requisitos como: confiabilidade, desempenho, proteção ou disponibilidade. Foram definidos para o *software* os requisitos não funcionais à seguir:

1. [RNF001] Acessível por navegadores, inclusive *mobile*;
2. [RNF002] Interface intuitiva e minimalista;
3. [RNF003] Tratamento em uso desconectado.

Foram desenvolvidos os principais Diagramas UML, inicialmente o **Diagrama de Casos de Uso** (ver Figura 3). Esse diagrama de acordo com [Guedes 2011] este é o mais geral da UML, seu modo informal faz ser um dos primeiros a ser utilizado no levantamento e definição de requisitos. Costuma ser consultado também durante o processo, inclusive base para outros diagramas. Colabora ainda para que os usuários tenham ideia geral do funcionamento proposto, por sua simplicidade e fácil compreensão. Nesse diagrama são identificados atores (pessoas, sistemas, hardware) e funcionalidades, nele denominadas casos de uso.

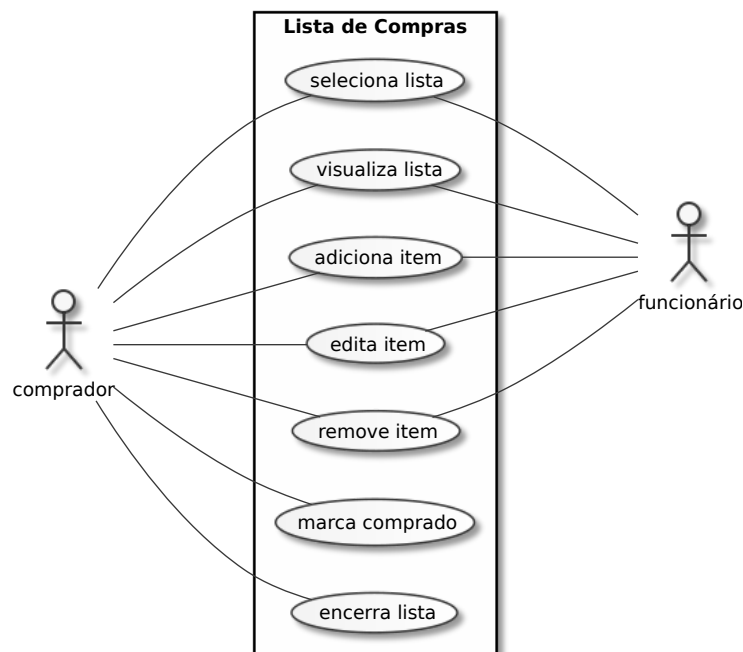


Figura 3. Diagrama de Casos de Uso

Também foi criado o **Diagrama de Classes** (ver Figura 4) que é possivelmente o mais utilizado, além de importante é apoio para alguns dos demais diagramas [Guedes 2011]. Este diagrama apresenta atributos, métodos e estrutura das classes, além da comunicação e relação entre elas.

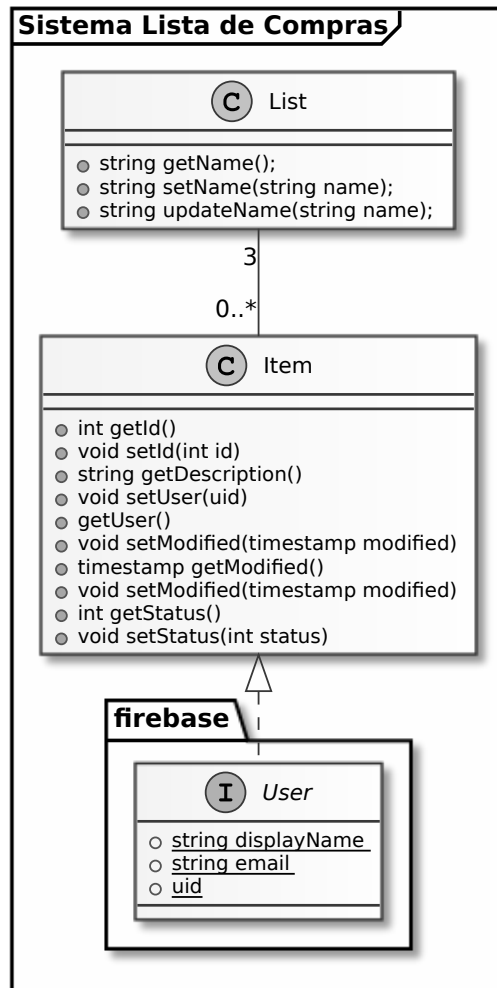


Figura 4. Diagrama de Classes

A modelagem conceitual de banco de dados é descrita por [Heuser 2009] como primeira etapa do projeto de banco de dados, seu objetivo é obter descrição abstrata dos dados, independente da implementação computacional que for realizada. A técnica mais difundida utiliza o modelo entidade-relacionamento (modelo ER), sua representação gráfica utiliza diagrama próprio (ver Figura 5).

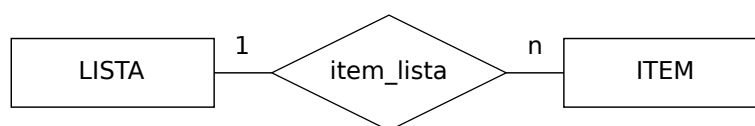


Figura 5. Diagrama Entidade-Relacionamento

Acesso Offline

Com auxílio de *Service Workers* (SW) é possível resolver a navegação *offline*. Está sendo desenvolvido, como explica [Archibald 2014], por um esforço colaborativo entre grandes empresas como Google, Mozilla e outras, sua implementação tem sido constante principalmente nos navegadores Chrome e Firefox. Realmente excitante para os interessados na competição entre web e aplicações nativas nos sistemas operacionais.

Sua implementação no modo mais básico em uma aplicação de página única pode ser considerada simples. Porém sua versatilidade e aplicabilidade é incrivelmente mais extensa. Inicialmente é necessário informar ao navegador o arquivo JavaScript que será o controlador, se comportando como se fosse um proxy local. Para isso utiliza-se o comando de registro (conforme Figura 6), após testar se o navegador é compatível.

```
<!DOCTYPE html>
<html>
  <head>
    (...)
  <body>
    (...)a
    <script>
      if( navigator.serviceWorker !== undefined){
        navigator.serviceWorker.register('sw.js');
      }
    </script>
  </html>
```

Figura 6. HTML destacando o registro do SW

Dentro do arquivo `sw.js`, notável em Figura 7, constam dois eventos: um de instalação para quando o navegador instala (*install*) de acordo com o registro e outra para captura de ação de rede (*fetch*), onde iria naturalmente requisitar determinado recurso. No evento de instalação lista os arquivos a serem armazenados e no de rede indica a utilizar os arquivos armazenados.

```
self.oninstall = function (event) {
  event.waitUntil(
    caches.open('bizual-static-v1').then(function (cache) {
      return cache.addAll([
        './',
        'css/material.teal-lime.min.css',
        'js/material.min.js',
        'js/script.js',
      ]);
    })
  );
};

self.onfetch = function (event) {
  event.respondWith(
    caches.match(event.request)
  );
};
```

Figura 7. Conteúdo do arquivo SW

Durante o artigo foi utilizada a denominação “navegador moderno” neste estudo são navegadores que apostam em novas tecnologia que estejam surgindo. Normalmente podem ser considerados o Google Chrome, Mozilla Firefox e Opera. A Microsoft vêm flexibilizando essas adoções nas últimas versões e do seu substituto o Edge, porém quando tecnologias ainda em rascunho ela opta por sua definição e maturidade. A Tabela 1 apresenta como está, mesmo que parcial, essa compatibilidade. Ressaltando o quão recente é essa adoção, as versões lançadas somente neste ano como estáveis entre 9 de setembro e 23 de outubro, são nesses navegadores a primeira implementação ocorrida. Apenas o Google Chrome para desktop possui compatibilidade nas versões anteriores, desde 2 de maio também deste ano.

Edge	Firefox	Chrome	Safari	Opera	iOS	Android
	49+	49+		41+		5.0+

Tabela 1. Suporte à SW pelos navegadores

Considerações Finais

A resistência em automatização em pequenos comércios é uma realidade. Nem em situações em que o custo é mínimo e as vantagens são evidentemente mensuradas há o apoio necessário. Foi documentada uma proposta para definição de um domínio na internet e contratação do serviço de hospedagem, que devido a outras prioridades não foi devidamente apreciado. Devido a esta falta de aprovação foi aqui considerado como negado, pelo menos para o período de início e fim do presente estágio.

Foi atingido o essencial que era o estudo do caso e o estudo de viabilidade, bem como levantamento, requisitos, diagramas, banco de dados. Houve atraso tecnológico devido ao investimento necessário em serviço de hospedagem do servidor. Ficando a própria implementação da solução como trabalho futuro.

Referências

- [Archibald 2014] Archibald, J. (2014). Service worker - first draft published. <https://jakearchibald.com/2014/service-worker-first-draft/>. Acesso em: 4 novembro 2016. 5
- [Guedes 2011] Guedes, G. T. A. (2011). *UML 2 : uma abordagem prática*. Novatec, São Paulo, 2 edition. 4
- [Heuser 2009] Heuser, C. A. (2009). *Projeto de banco de dados*. Novatec, Porto Alegre, 6 edition. 5
- [Sebrae 2015] Sebrae (2015). Automatizar os processos de uma empresa é uma boa prática. <http://www.sebrae.com.br/sites/PortalSebrae/artigos/automatizar-os-processos-de-uma-empresa-e-uma-boa-pratica>. Acesso em: 2 novembro 2016. 1
- [Sommerville 2011] Sommerville, I. (2011). *Engenharia de Software*. Pearson, São Paulo, 9 edition. 3, 4