

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. Abstract.

Resumo. Resumo.

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.

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



Figura 2. Modelo de Impressão da Planilha

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

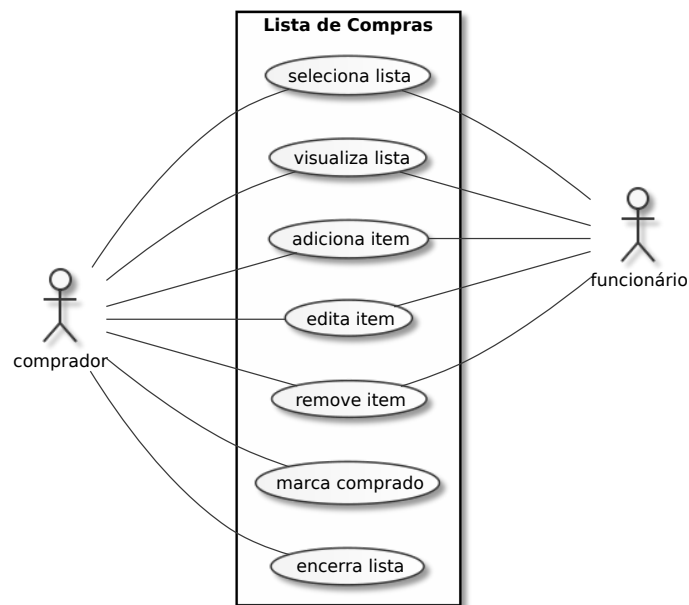


Figura 3. Diagrama de Casos de Uso

diagrama são identificados atores (pessoas, sistemas, hardware) e funcionalidades, nele denominadas 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.

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).

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 navegar 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.

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

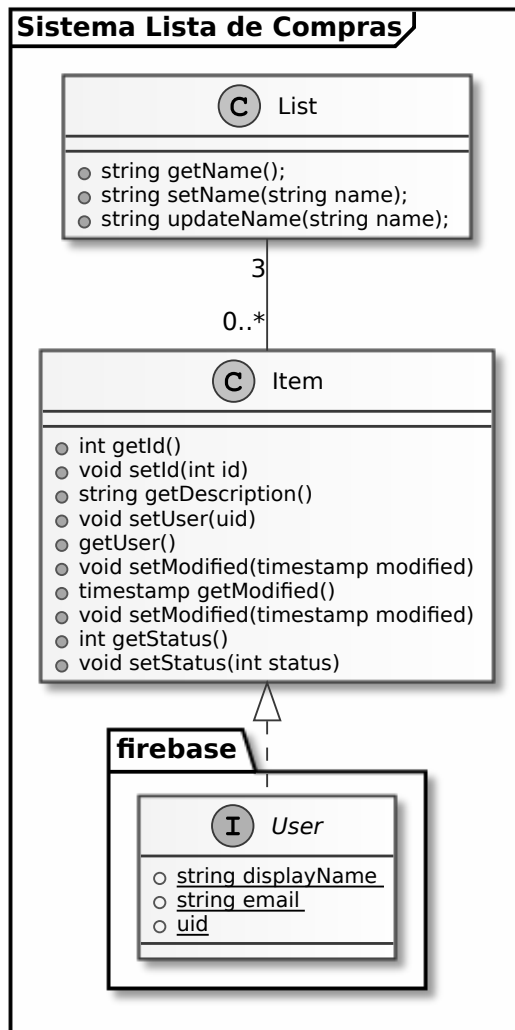


Figura 4. Diagrama de Classes

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.

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 compabilidade nas versões anteriores, desde 2 de maio também deste ano.



Figura 5. Diagrama Entidade-Relacionamento

```

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

```

Figura 6. HTML destacando o registro do SW

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.

O projeto foi inicialmente acordado verbalmente com boa aceitação. Após o listado como solução proposta, continuou o aceite porém com ressalvas. A principal diz respeito a uso de smartphone ser levemente controlado, por solicitação quando possível deixar em seu armário ou com uso muito restrito, apenas para que não haja um abuso. Outra ressalva, em continuidade à anterior, são as poucas funcionárias que por opção não se utilizam de celulares, devendo assim ser possível selecionar qual funcionário através de alguns logins.

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.

Como trabalho futuro fica o desenvolvimento e implementação da solução proposta. O supervisor sugeriu acrescentar a possibilidade de informar o valor da compra para consulta em compras posteriores para comparativo.

Também foi notado pelo supervisor que não há smartphones corporativos e nem todos os funcionários têm aparelhos para o uso proposto, logo deve possibilitar a troca

```

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

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

Tabela 1. Suporte à SW pelos navegadores

após logado de qual funcionário está informando o item. Foi sugerido e é possível que no próximo ano o celular corporativo deixe de ser um feature phone para ser um smartphone.

Referências

- [Archibald 2014] Archibald, J. (2014). Service worker - first draft published. <https://jakearchibald.com/2014/service-worker-first-draft/>. Acesso em: 4 novembro 2016. 4
- [Guedes 2011] Guedes, G. T. A. (2011). *UML 2 : uma abordagem prática*. Novatec, São Paulo, 2 edition. 3, 4
- [Heuser 2009] Heuser, C. A. (2009). *Projeto de banco de dados*. Novatec, Porto Alegre, 6 edition. 4
- [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