

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO DE PÓS-GRADUAÇÃO
ESPECIALIZAÇÃO EM DESENVOLVIMENTO WEB

RODRIGO FERREIRA DE SOUZA

**UTILIZAÇÃO DE PÁGINAS *WEB OFF-LINE* NA REALIZAÇÃO DE
INVENTÁRIOS DE PATRIMÔNIO**

MONOGRAFIA DE ESPECIALIZAÇÃO

LONDRINA
2012

RODRIGO FERREIRA DE SOUZA

**UTILIZAÇÃO DE PÁGINAS *WEB OFF-LINE* NA REALIZAÇÃO DE
INVENTÁRIOS DE PATRIMÔNIO**

Trabalho de Conclusão de Curso apresentado ao Departamento de Pós-Graduação da Universidade Federal do Paraná, como requisito parcial à obtenção do grau de especialista em Desenvolvimento Web.

Orientador: Prof. Thiago Prado de Campos

LONDRINA
2012



TERMO DE APROVAÇÃO

Título da Monografia

UTILIZAÇÃO DE PÁGINAS WEB OFF-LINE NA REALIZAÇÃO DE INVENTÁRIOS DE PATRIMÔNIO

por

RODRIGO FERREIRA DE SOUZA

Esta monografia foi apresentada às 17h00 do dia **23** de **fevereiro** de 2013 como requisito parcial para a obtenção do título de ESPECIALISTA EM DESENVOLVIMENTO WEB. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho

_____.

(aprovado, aprovado com restrições ou reprovado)

Prof. Thiago Prado de Campos
(UTFPR)

Prof. Dr. César Augusto Cusin
(FAIT)

Prof. Esp. Dhiego Augusto dos Santos
Bicudo
(Agência Avadora)

Visto da coordenação:

Prof. Thiago Prado de Campos
Coordenador da esp. em Desenvolvimento Web

Prof. Walmir Eno Pottker
Coordenador de Pós-Graduação Lato Sensu

Dedicatória

Dedico este trabalho a minha querida família e amigos.

AGRADECIMENTOS

Ao meus pais, que embora não tenham estudado muito, sempre me incentivaram no estudo e me mostraram os caminhos corretos a seguir.

A minha esposa, pelas palavras de conforto em situações difíceis, conselhos.

Ao Prof^o Thiago Prado de Campos, pelos conselhos e sábia orientação a respeito deste trabalho, e a todos os professores do curso de desenvolvimento web pelos conhecimentos ministrados.

Aos colegas, pela amizade, e a todas as pessoas que direta ou indiretamente contribuíram para a conclusão do curso.

*A mente que se abre a uma nova ideia jamais
voltará ao seu tamanho original. (Albert Einstein)*

SOUZA, Rodrigo Ferreira. **Utilização De Páginas Web Off-Line Na Realização De Inventários De Patrimônio**. 2012. 61. Trabalho de Conclusão do Curso de Desenvolvimento Web – Departamento de Pós-Graduação, Universidade Federal do Paraná, Londrina, 2012.

RESUMO

Este trabalho tem como objetivo realizar um estudo de tecnologias abertas e padrões *web* aplicadas para a realização de inventários de patrimônio. O objetivo principal desse estudo é criar uma aplicação que funcione *off-line* em dispositivos móveis, tornando possível a coleta de informações dos bens em diversos setores das empresas, além de executar a sincronia dos dados com o servidor quando o aparelho voltar a se conectar na rede interna da empresa. Desta forma, o trabalho descreve os passos tomados para a criação da aplicação, sendo eles o levantamento de requisitos, um estudo sistematizado das tecnologias existentes, projeto de telas e diagramação da aplicação. O uso da aplicação desenvolvida trouxe benefícios antes não existentes no processo de inventário de patrimônio, como o reaproveitamento de informações dos inventários anteriores, a não necessidade de redigitar as informações coletadas, e a possibilidade de coletar a geolocalização e fotos dos bens sendo inventariados. Além da aplicação desenvolvida, este trabalho traz como contribuição uma discussão de técnicas e tecnologias utilizadas para conseguir uma aplicação simples e eficaz aproveitando as possibilidades da tecnologia móvel em aplicações *web*.

Palavras-chave: AppCache, *Web storage*, *Geolocation*, HTML5, *Mobile first*, Restful *web services*, QR Code, Backbone.js, Inventário de patrimônio.

SOUZA, Rodrigo Ferreira. **Utilização De Páginas Web Off-Line Na Realização De Inventários De Patrimônio**. 2012. 61. Trabalho de Conclusão do Curso de Desenvolvimento Web – Departamento de Pós-Graduação, Universidade Federal do Paraná, Londrina, 2012.

ABSTRACT

This assignment has the objective of studying open technologies and web standards applied when making inventories of assets. The main objective of this study is to create an application that works off-line on mobile devices, making it possible to collect assets data in various sectors of the companies, besides executing the synchronization of data with the server when the device reconnects to the company's internal network. Thus, the paper describes the steps taken to create the application, which were technical requirements description, a systematic study of existing technologies, screen design and creation of structure diagrams. Using the developed application brought benefits that didn't exist in the inventory of assets, such as reuse of information from previous inventories, no need to type again the information collected, and the possibility to collect the geolocation and photos of the asset being inventoried. Besides the developed application, this paper brings a discussion of techniques and technologies used to achieve a simple and effective application taking advantage of the possibilities of mobile technology in web applications.

Keywords: AppCache, Web storage, Geolocation, HTML5, Mobile first, Restful web services, QR Code, Backbone.js, Inventory of assets.

ÍNDICE DE FIGURAS

Figura 1: Exemplo de QR Code.....	17
Figura 2: Serviço prestado pela AfixCode de Inventário de Patrimônio (AFIXCODE, 2012).....	22
Figura 3: Mudanças no fluxo de trabalho em um inventário de patrimônio.....	24
Figura 4: Caso de uso de alto nível.....	25
Figura 5: Como foi planejado e como ficou a tela principal da aplicação.....	26
Figura 6: Como foi planejado e como ficou a tela do relatório.....	27
Figura 7: Ao imprimir o relatório, é escondido o cabeçalho e rodapé do sistema.....	27
Figura 8: Como foi planejado e como ficou a tela de ajuda.....	28
Figura 9: A tela de contato deixou de existir, e foi criada a tela mobile para acessar o sistema mobile.....	29
Figura 10: Como foi planejado, e como ficou o menu principal.....	30
Figura 11: Indicativo se a página está operando on-line ou off-line.....	30
Figura 12: Como foi planejado, e como ficou o menu de cadastros.....	31
Figura 13: Como foi planejado, e como ficou a tela de cadastro de bem.....	32
Figura 14: Como foi planejado, e como ficou a tela de adicionar ou editar o bem.....	33
Figura 15: Como foi planejado, e como ficou a tela de cadastro de inventário.....	34
Figura 16: Como foi planejado, e como ficou a tela de adicionar ou editar inventário.....	34
Figura 17: O menu de movimentação deixou de existir, ao invés dele é chamado o lançamento de inventário.....	35
Figura 18: Como foi planejado, e como ficou a tela principal de adicionar ou editar lançamento de inventário.....	36
Figura 19: Demais telas para adicionar um lançamento de inventário.....	37
Figura 20: Demais telas para editar um lançamento de inventário.....	38
Figura 21: Diagrama entidade relacionamento.....	39
Figura 22: Diagrama de classes.....	40
Figura 23: Diagrama de atividades - Checagem e sincronização das informações no sistema mobile.....	41
Figura 24: Diagrama de comunicação - Salvar dados no sistema.....	42
Figura 25: Lançamento de Inventário (Fotos) – 320x480.....	60
Figura 26: Lançamento de Inventário (Fotos) – 480x320.....	61
Figura 27: Lançamento de Inventário (Fotos) – 600x800.....	62
Figura 28: Lançamento de Inventário (Fotos) – 800x600.....	63
Figura 29: Lançamento de Inventário (Fotos) – 768x1024.....	64
Figura 30: Lançamento de Inventário – 1024x768.....	65

ÍNDICE DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

- PDA – Personal Digital Assistants
- QR Code – Quick Response Code
- HTML – Hyper-text Markup Language
- API – Application Programming Interface
- W3C – World Wide Web Consortium
- MVC – Model View Controller
- MVP – Model View Presenter
- ORM – Object Relational Mapper
- SQL – Structured Query Language
- SGBD – Sistema Gerenciador de Banco de Dados
- CSS – Cascade Style Sheets
- SASS – Syntatically Awesome Stylesheets

SUMÁRIO

ÍNDICE DE FIGURAS.....	9
ÍNDICE DE TABELAS.....	10
1INTRODUÇÃO.....	13
2FUNDAMENTAÇÃO TEÓRICA.....	15
2.1Definições Contábeis.....	15
2.1.1Gestão Patrimonial.....	15
2.1.2Balanço Patrimonial.....	15
2.1.3Inventário de Patrimônio.....	16
2.2Tecnologias Disponíveis.....	16
2.2.1QR Code.....	17
2.2.2Novos recursos da HTML.....	18
2.2.3Mobile first.....	19
2.2.4Restful Web Services.....	20
2.3Trabalhos Relacionados.....	22
3DESENVOLVIMENTO.....	23
3.1Descrição.....	23
3.1.1Requisitos do sistema.....	23
3.2Como o sistema foi pensado.....	24
3.2.1Fluxo de trabalho (Workflow).....	24
3.2.2Diagrama de caso de uso.....	25
3.2.3Interface.....	25
3.2.4Diagrama entidade relacionamento.....	39
3.2.5Diagrama de classes.....	39
3.2.6Diagrama de atividades.....	40
3.2.7Diagrama de comunicação.....	41
3.3Como o sistema foi implementado.....	42
3.3.1Python.....	43
3.3.2Pyramid.....	43
3.3.3SQLAlchemy.....	44
3.3.4Coffeescript.....	45
3.3.5Backbone.js.....	45
3.3.6Sass.....	46
3.3.7Webassets.....	47
4CONCLUSÃO.....	49
REFERÊNCIAS.....	50
APÊNDICE I – ENTREVISTA SOBRE INVENTÁRIO DE PATRIMÔNIO.....	55
APÊNDICE II – TELAS DO SISTEMA MOBILE EM FORMATOS RESPOSIVOS.....	59

1 INTRODUÇÃO

O Balanço Patrimonial é um demonstrativo que informa os Bens Ativos e Passivos de uma Entidade. A palavra Balanço remete a igualdade que existe entre o Ativo e o Passivo neste demonstrativo. A palavra Patrimonial, vem de Patrimônio, ou seja, bens e dívidas pertencentes a uma empresa ou instituição.

Segundo a legislação Brasileira, esse demonstrativo deve ser realizado pelo menos uma vez por ano, e é uma ferramenta poderosa que permite acompanhar todo o fluxo de dinheiro dentro da empresa.

Para realizar o Balanço Patrimonial, é necessário executar o Inventário de Patrimônio. O Inventário de Patrimônio consiste na contagem dos bens da empresa e anotação do seu estado de conservação, determinando qual a sua depreciação.

Geralmente esse inventário é realizado manualmente, anotando os dados dos bens sendo inventariados com o auxílio de pranchetas de papel, e posteriormente essas informações são digitadas no sistema principal de gerencia da empresa. Além do processo de contagem ser demorado, a digitação dos valores precisa ser revisada mais de uma vez para ter certeza que os dados informados no sistema estão iguais aos dados anotados em papel.

Alguns sistemas permitem informar todos os valores mais de uma vez, e mostram só a diferença entre as duas últimas digitações para uma conferencia final, o que agiliza um pouco o processo de digitação, pois a pessoa que está digitando os dados pode fazer isso de forma mais rápida.

Há poucos sistemas no mercado que utilizam integração com aparelhos PDA's (*Personal Digital Assistants*) e estes sistemas costumam ter desvantagens como: elevado valor da solução; utilização de aparelhos antigos comercializados para uma classe restrita da população e; maior complexidade de desenvolvimento por não haver um padrão de tecnologia para desenvolvimento de aplicativos.

Com o advento dos *smartphones* e sua popularização é possível criar uma solução mais simples e de custo baixo. Esses dispositivos são produzidos em alta escala, uma vez que a nova geração de aparelhos foi pensada para funcionar com padrões, e é perfeitamente possível criar uma aplicação que funciona tanto em

aparelhos pequenos como *smartphones*, quanto em aparelhos maiores como *tablet's*¹.

¹Deste ponto em diante, esses aparelhos serão denominados somente como Dispositivos Móveis.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 DEFINIÇÕES CONTÁBEIS

A seguir veremos algumas das definições contábeis inseridas no contexto desse trabalho.

2.1.1 Gestão Patrimonial

O Patrimônio é o conjunto de bens, direitos e obrigações pertencentes a uma pessoa física ou jurídica (APÊNDICE I).

Os bens podem ser classificados como tangíveis e intangíveis, sendo que os bens intangíveis são aqueles que não podem ser medidos, como patentes de produtos ou recursos humanos por exemplo. Os bens tangíveis se restringem aos bens móveis e imóveis, que sofrem depreciação, possuem uma vida útil finita e necessitam de inventários e avaliações para projetar sua substituição por bens mais novos (IBRACON, 2012).

A Gestão Patrimonial é o conjunto de procedimentos necessários para determinar o ativo patrimonial de uma empresa. Através do controle apropriado dos bens é possível determinar a valoração correta dos ativos da entidade, representando de forma fidedigna os resultados patrimoniais da empresa (APÊNDICE I).

2.1.2 Balanço Patrimonial

O Balanço Patrimonial é um demonstrativo cuja principal função é mostrar de forma comparativa a real situação da entidade, separando os resultados em ativos e passivos (APÊNDICE I).

Por meio da análise deste documento é possível identificar seus saldos positivos e negativos, e confrontando esses resultados é apurado se houve um resultado superavitário ou deficitário (APÊNDICE I).

Sua principal utilização está ligada a necessidade de um controle formal e padronizado do patrimônio da empresa por parte dos gestores, e pela apuração de impostos realizada pelos órgãos públicos (DIAS, 2011).

Parte das informações contidas no Balanço Patrimonial é levantada através de um Inventário de Patrimônio.

2.1.3 Inventário de Patrimônio

O Inventário de Patrimônio é um documento que evidencia a existência física dos bens, e a qualidade desse documento é medida pela quantidade de detalhes existente no mesmo (APÊNDICE I).

Segundo a NBC T 19.1 (Normas Brasileiras de Contabilidade), o Inventário de Patrimônio deve descrever a época do reconhecimento de ativos, a determinação de seus valores nos registros contábeis e as circunstâncias que podem influenciar o momento em que esses valores são levados às contas de resultado (CONSELHO FEDERAL DE CONTABILIDADE, 2005).

Ou seja, em um inventário de patrimônio, podemos ter informações como a localização física do bem, imagens, medidas, características físicas, valor atual, documento comprobatório do valor atual, depreciação, origem do bem, estado de conservação, pessoa responsável pelo bem, data de aquisição, data do levantamento, enfim, quanto mais informações relacionadas ao bem descritas no Inventário de Patrimônio, melhor (APÊNDICE I).

Os dados do Inventário de Patrimônio aparecem na seção de imobilizado do Balanço Patrimonial (CONSELHO FEDERAL DE CONTABILIDADE, 1990), e em uma auditoria esse documento pode ser utilizado para confrontar a veracidade dos dados informados, tão bem como pode ser utilizado para confrontar as suas informações com os bens existentes na empresa (APÊNDICE I).

2.2 TECNOLOGIAS DISPONÍVEIS

Para o desenvolvimento de um sistema informatizado que otimize a execução de inventários de patrimônio, precisamos de tecnologias que sirvam para identificação de objetos, armazenamento de dados, interface de uso e de

comunicação com sistemas servidores que sejam compatíveis com os dispositivos móveis aos quais queremos suportar. Descrevemos algumas das tecnologias disponíveis a seguir.

2.2.1 QR Code

QR Code (*Quick Response Code*) é um código de barras em formato matricial que utiliza um padrão único de busca localizado em três dos seus cantos com o objetivo de facilitar a mensuração de sua posição, tamanho e inclinação (INTERNATIONAL STANDARD, 2000).

Foi criado pela equipe da Toyota para facilitar a rastreabilidade de seus produtos na linha de produção e atualmente essa tecnologia está ganhando muito espaço na utilização com dispositivos móveis, pois ela facilita a integração de sistemas convencionais com dispositivos móveis (DENSO WAVE INCORPORATED, 2012).



Figura 1: Exemplo de QR Code

Por utilizar um formato matricial ao invés de um formato linear encontrado na maioria dos códigos de barras existentes, ele consegue trazer diversas vantagens aos formatos antigos, enumerados a seguir (DENSO WAVE INCORPORATED, 2012):

1. Enquanto os formatos normais conseguem codificar no máximo 20 caracteres, o QR Code pode codificar um texto de até 7.089 dígitos (DENSO WAVE INCORPORATED, 2012).

2. Por guardar informações tanto no sentido horizontal quanto vertical, o QR *Code*, ele pode guardar a mesma informação que um código de barras comum utilizando um décimo do espaço de impressão (DENSO WAVE INCORPORATED, 2012).
3. É possível guardar textos em caracteres especiais, como Kanji e Kana (DENSO WAVE INCORPORATED, 2012).
4. Utiliza um algoritmo de correção de erros, ou seja, caso o código de barras for danificado por sujeira ou rasuras em até 30%, o código de barras ainda vai poder ser interpretado sem problemas (DENSO WAVE INCORPORATED, 2012).
5. O código de barras pode ser lido em qualquer direção (360°) facilitando a utilização do aparelho na leitura do código de barras (DENSO WAVE INCORPORATED, 2012).
6. Um código de barras pode ser separado em até 16 códigos de barras menores com a mesma informação, permitindo imprimir em um espaço mais estreito (DENSO WAVE INCORPORATED, 2012).

QR *Code* é útil de duas formas no sistema. A primeira é eliminar a necessidade de digitar no dispositivo móvel o endereço web para acessar o sistema. Para tanto, um QR *Code* aparece na página web do sistema para poder sincronizar os dados cadastrados com o dispositivo móvel. A outra forma de utilizar o QR *Code* imprimindo etiquetas com as informações do bem. Nesta etiqueta, existe a informação do endereço do sistema e o código do bem, para facilitar a busca do bem nos dados sincronizados no dispositivo.

2.2.2 Novos recursos da HTML

A versão cinco da linguagem HTML (*Hyper-text Markup Language*) e novas APIs (*Application Programming Interfaces*) trouxeram importantes inovações para sistemas *Web* com diversos recursos que podem ser aplicados em uma enorme gama de soluções digitais.

Um dos recursos utilizado neste projeto é o *AppCache*, que avisa para o navegador quando ele deve guardar uma cópia de alguns dos arquivos necessários para mostrar a página por exemplo, para que ela continue funcionando

quando não existir conexão de internet (BIDELMAN, 2012). Na prática, a página pode passar a funcionar como um aplicativo comum instalado no computador (CHRISTIAN, 2012).

Outro recurso importante é o *Web Storage*, através do qual, é possível guardar informações localmente no navegador sem precisar acessar um servidor de banco de dados (PILGRIM, 2012). Este recurso é utilizado para poder colher as informações do inventário em ambientes onde não existe conexão com a rede de dados da empresa. Assim que a página conseguir acessar novamente o sistema, as informações armazenadas no *Web Storage* são sincronizadas com a aplicação.

Uma das premissas de um inventário é guardar a localização do que está sendo inventariado. Para ajudar nesse ponto existe uma API para Geolocalização, com o qual é possível extrair a atual localização do aparelho utilizado no inventário em coordenadas de latitude e longitude (W3C, 2012). Essa informação pode ser então facilmente coletada e guardada como complemento das informações de localização cadastradas no sistema.

2.2.3 *Mobile first*

Levando em conta as mudanças ocorridas no cenário da computação móvel entre os anos de 2006 e 2012 Luke Wroblewski sugere que seja pensado primeiro em como as aplicações *Web* irão funcionar no ambiente com recursos limitados de um *smartphone* para depois se aproveitar das vantagens adquiridas dessa experiência na criação da solução *Web* principal em geral voltada para desktop (WROBLEWSKI, 2012).

Seguindo esse preceito, o desenvolvimento de aplicativos *Web* ganham foco e simplicidade, pois não há espaço na pequena tela de um *smartphone* para adicionar recursos desnecessários, ou seja, é preciso ter um real conhecimento de como o aplicativo será utilizado pelas pessoas, e quando será utilizado (WROBLEWSKI, 2012).

Outro ganho está na performance, pois tudo que é possível fazer para que um aplicativo se comporte mais rápido nas redes com banda limitada dos aparelhos móveis, deve ser feito. O que difere muito das aplicações *Web*

convencionais que existem hoje, onde recursos são abundantes e essa preocupação é tratada somente como um gasto desnecessário de tempo (WROBLEWSKI, 2012).

Um cuidado essencial no desenvolvimento de aplicativos Web móveis é o contexto. Por ser naturalmente portátil, um dispositivo móvel é utilizado em lugares e momentos diferentes de um aplicativo web comum (WROBLEWSKI, 2012).

Um exemplo claro está na solução proposta nesse trabalho. Como ferramenta para a realização de um inventário de patrimônio, até hoje utilizamos pranchetas de papel por que os computadores não se encaixam nesse contexto, são pesados para serem carregados o dia todo, enquanto dispositivos móveis cabem na palma da mão e carregamos eles conosco naturalmente, precisando ou não.

Sendo assim, a solução apresentada neste trabalho é feita com essas ideias e conceitos em mente, procurando extrair o que há de melhor nelas.

2.2.4 Restful Web Services

O resultado desse trabalho não tem como objetivo substituir sistemas gerenciais existentes, mas sim complementá-los com os recursos possíveis da computação móvel. Então, em algum momento, nossa aplicação terá que conversar com os outros sistemas, daí que entra em cena os *Web Services*, que são pontos de acesso a uma aplicação que dão a possibilidade de serem manipulados por outra aplicação terceira, ou seja, é um contrato de prestação de serviços entre dois programas de finalidades distintas para se complementar em uma tarefa maior (RICHARDSON, 2012).

Restful Web Services seguem uma linha um pouco diferente dos *Web Services* comuns. Estes últimos funcionam em cima da *Web* somente com o objetivo de passar com mais facilidade pelas restrições impostas por barreiras na rede de internet (também conhecidos como *firewalls*). Já *Restful Web Services* por sua vez adotam os ideais existentes na *Web*, trazendo consigo várias vantagens (RICHARDSON, 2012).

Um *Web Service* comum utiliza a *Web* somente como uma ponte entre duas aplicações, a mensagem trafegada entre as aplicações geralmente traz diversas informações que são somente relevantes para a linguagem de

programação a qual foi implementado, o que dificulta a inter-operabilidade entre implementações com linguagens diferentes (RICHARDSON, 2012).

Na prática, se for criado um *Web Service* comum em uma linguagem para conversar com uma outra aplicação na mesma linguagem, tudo ocorre de uma forma simples e transparente, por que a complexidade fica escondida em programas assistentes (também conhecidos como *wizards*) embutidos na ferramenta IDE (*Interface Development Environment*) criada para a linguagem de programação utilizada .

Mas se for criado um *Web Service* comum em uma linguagem para conversar com outra linguagem que não possui a mesma IDE, essa complexidade atua como um fator impeditivo, dificultando muito o simples objetivo maior de fazer dois sistemas conversarem e se complementarem.

Pensando nisso, a principal vantagem de utilizar um *Restful Web Service* é sua simplicidade na mensagem trafegada entre os sistemas, facilitando a comunicação entre qualquer linguagem, e seguindo um dos princípios que torna a *Web* tão importante para todos, a inter-operabilidade (RICHARDSON, 2012).

Esse simples fato vem de encontro com a necessidade imposta pela aplicação de inventário construída, pois ela deve complementar outras aplicações de forma simples, independente de qual técnica utilizada nessa aplicação, sem trazer restrições e dificuldades.

Isto exposto, foi possível criar uma aplicação rica em funcionalidades: para facilitar a utilização do sistema com dispositivos móveis, utilizamos *QR Code*; para fazer a aplicação funcionar *offline* utilizamos *AppCache* e *Web Storage*; para complementar a aquisição de informações do patrimônio utilizamos *Geolocalização*, e; para que o aplicativo consuma pouca largura de banda os arquivos em linguagem *CSS (Cascade Style Sheet)* e *Javascript* são mesclados, minificados e compactados. Outra premissa da aplicação foi manter o foco na simplicidade de sua interface. Finalmente o acesso as informações é exposto em um *Restful Web Service* para poder se integrar com outros sistemas, conseguindo uma solução simples e eficaz para solucionar o problema proposto.

2.3 TRABALHOS RELACIONADOS

A grande maioria das empresas que possuem soluções que contemplam módulo de inventário patrimonial focam a gerencia do patrimônio, como a definição de vida útil e cálculo de depreciação, porém não possuem nenhuma integração com aparelhos coletores de dados, fazendo-se necessário preencher pranchetas de papel com as informações coletadas para posterior digitação nos sistemas.

Esse é o caso dos sistemas das empresas Reliasoft ® (RELIASOFT CORPORATION, 2012), SoftExpert ® (SOFTEXPERT SOFTWARE FOR PERFORMANCE EXCELLENCE, 2012), e WinPatrimonio ® (WINPATRIMONIO, 2012), que possuem todo o controle patrimonial em seus sistemas, mas não trazem a facilidade da utilização de um aparelho para facilitar a coleta de dados durante o inventário físico.

Existem alguns sistemas que possuem uma integração com aparelhos PDA's especiais, com leitores de código de barras embutidos. Mas estas soluções costumam ser muito caras por funcionar somente em um aparelho específico e sua programação costuma ser complexa por precisar utilizar a linguagem específica que este aparelho entende.

Como exemplo, a empresa AfixCode ® oferece o serviço terceirizado de inventário de patrimônio utilizando aparelhos PDA's na coleta dos dados. Este serviço cuida da gerência patrimonial como um todo, começando pela colocação de plaquetas identificadoras com código de barras, execução do inventário com PDA's, e disponibilizando aplicação gerencial que traz e calcula todas as informações de depreciação dos bens (AFIXCODE, 2012).



Figura 2: Serviço prestado pela AfixCode de Inventário de Patrimônio (AFIXCODE, 2012).

3 DESENVOLVIMENTO

3.1 DESCRIÇÃO

A aplicação desenvolvida tem como objetivo principal funcionar como uma ferramenta de suporte ao contador durante o processo de inventário de patrimônio, e para atingir esse objetivo, foram levantados os requisitos do sistema, realizada uma pesquisa das ferramentas, tecnologias e técnicas para atingir os objetivos, gerados diagramas que descrevem a estrutura e comportamentos do sistema, além da criação de protótipos das telas necessárias para o funcionamento da aplicação.

3.1.1 Requisitos do sistema

1. O sistema deve funcionar em ambiente *Web*.
2. O sistema deve funcionar em aparelhos móveis como *smartphones* e *tablets*, que serão utilizados na coleta dos dados de inventário.
3. O sistema deve possuir mecanismo para integração fácil com outros sistemas, dispensando a necessidade de digitar as informações cadastrais necessárias para seu funcionamento.
4. O sistema deve operar em modo *on-line* e *off-line*. O modo *on-line* ocorre quando o sistema estiver acessando diretamente o servidor para carregar as informações cadastrais, ou para consolidar os dados coletados durante o inventário. O modo *off-line* ocorre quando será executada a coleta de dados patrimoniais na empresa.
5. O sistema deve armazenar informações quando estiver operando em modo *off-line*.
6. O sistema deve sincronizar as informações coletadas em modo *off-line* com o servidor, quando voltar ao estado *on-line*.
7. O sistema deve utilizar tecnologias abertas.

3.2 COMO O SISTEMA FOI PENSADO

3.2.1 Fluxo de trabalho (*Workflow*)

Um dos principais objetivos da criação dessa aplicação é melhorar o fluxo de trabalho na realização de um inventário de patrimônio.

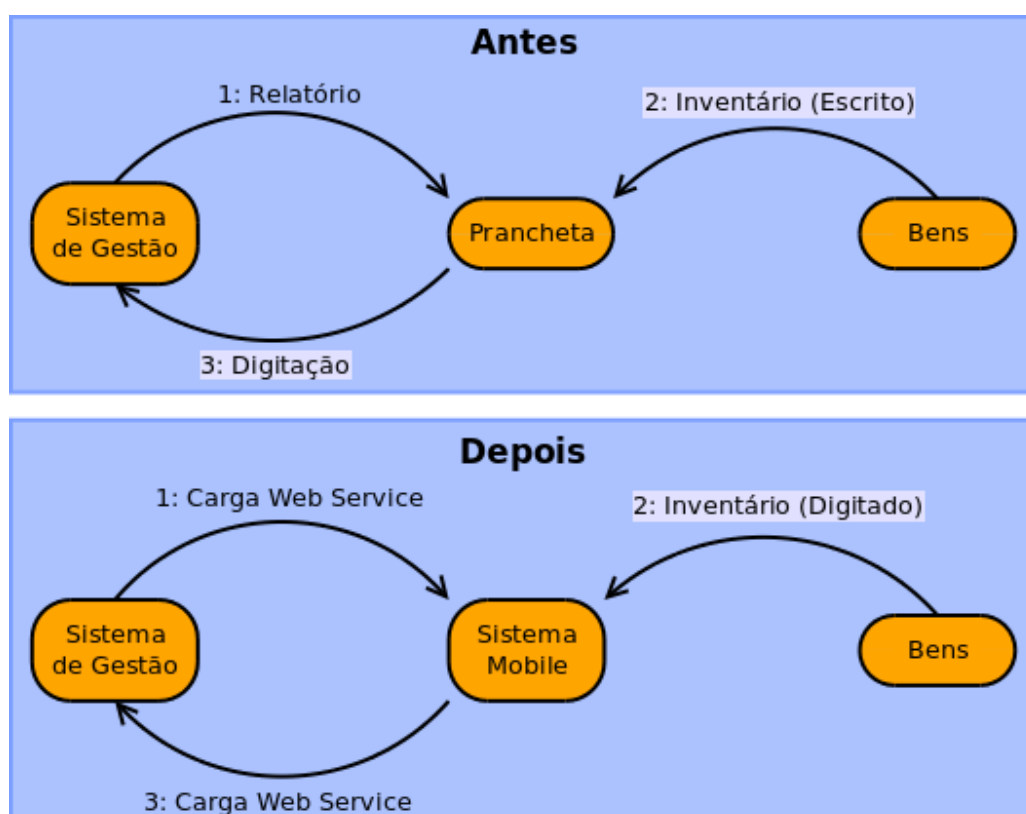


Figura 3: Mudanças no fluxo de trabalho em um inventário de patrimônio

Como podemos observar na Figura 3, na maioria das empresas é utilizada uma prancheta de papel para anotar os dados dos bens que devem ser inventariados, e após a realização dessa anotação é necessário digitar todos os dados colhidos em campo no sistema gerencial da empresa. Note que essa digitação é a maior causa de erros de inventários de patrimônio. Para diminuir esses erros alguns sistemas dão a possibilidade de que seja realizada a digitação diversas vezes, e após isso rodar uma conferência entre as duas últimas digitações, e apontar as divergências para conferência.

O fluxograma proposto pela ferramenta criada coloca o sistema *mobile* no lugar da prancheta, e os dados são colhidos diretamente nele, sem a necessidade de um segundo passo para digitar as informações. Os dados transmitidos para o aparelho para o sistema gerencial e vice versa através de um *Web Service*, o que ocorre de forma automática após a configuração do sistema gerencial para conversar com esse *Web Service*.

3.2.2 Diagrama de caso de uso

O diagrama abaixo (Figura 4) busca mostrar uma visão mais externa do sistema, e em quais partes cada ator acessa.

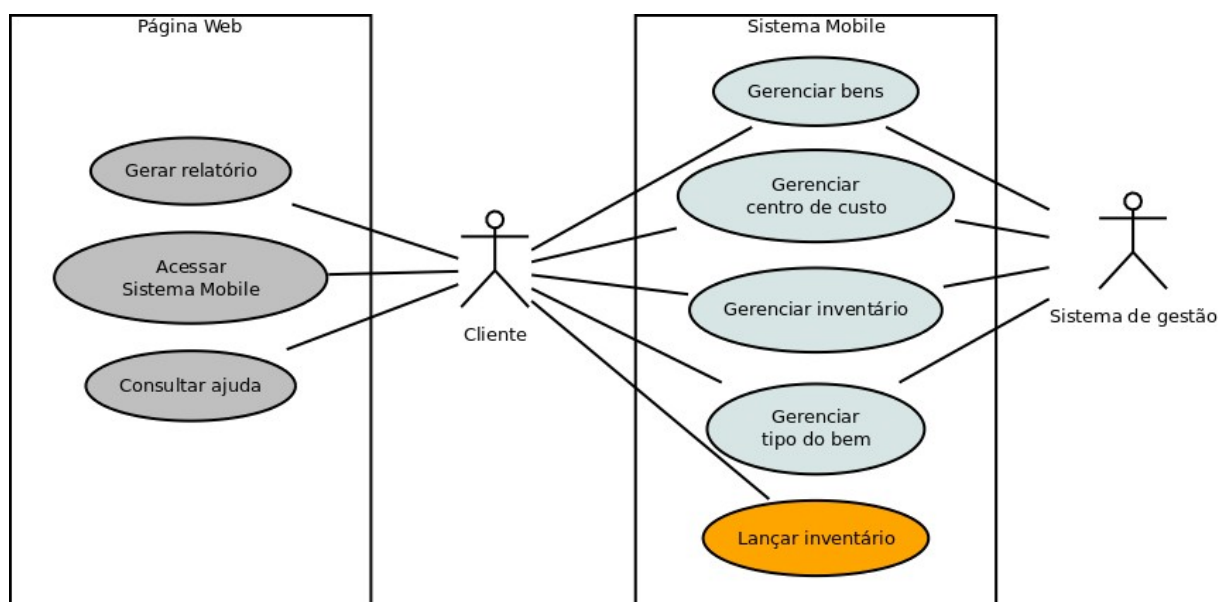


Figura 4: Caso de uso de alto nível

O cliente é o ator principal, que acessa todo o sistema e executa a rotina de lançar inventário no sistema. Além disso, o sistema de gestão da empresa pode carregar o sistema *mobile* com informações através do *Web Service*.

3.2.3 Interface

O sistema foi projetado para ter duas telas principais. A primeira tela é voltada para *desktop*, e sua principal função é ser a porta de entrada para o sistema *mobile*, trazendo: um breve texto de ajuda sobre como utilizar um programa leitor de

códigos de barra *QR Code*; uma listagem de bens com sugestão de etiquetas para impressão e: um *QR Code* apontando para a aplicação principal, utilizado para a primeira carga do sistema no *smartphone*. Opcionalmente, é possível também acessar o sistema *mobile* através da tela de *desktop*, caso seja necessário alguma manutenção nos cadastros sem utilizar o dispositivo móvel.



Figura 5: Como foi planejada e como ficou a tela principal da aplicação.

A Figura 5 mostra como foi planejada e como ficou a página principal do sistema. Entre as duas fases, podemos notar que: O menu principal ganhou mais destaque; A página de informações se tornou uma página de ajuda; E o menu de contato foi substituído pelas informações de contato no rodapé.

O *QR Code* que aparece ao lado do logotipo da aplicação serve para realizar a primeira carga do sistema no *cache* do navegador do dispositivo *mobile*. Esse procedimento deve ser realizado pelo menos uma vez com o aparelho conectado na mesma rede do sistema para que o aplicativo seja guardado no navegador, podendo ser executado quando o dispositivo móvel não possuir acesso a rede interna da empresa.



Figura 6: Como foi planejado e como ficou a tela do relatório.

A Figura 6 mostra como foi planejado e como ficou o relatório de bens do sistema. Este relatório traz uma lista de códigos de barras com um link especialmente formatado para acionar o lançamento de inventário do bem no sistema.

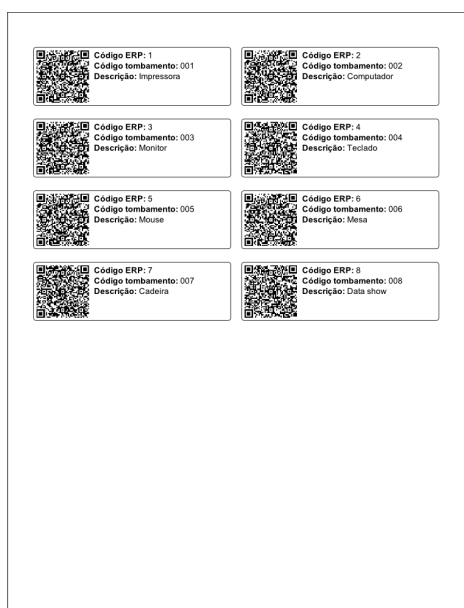


Figura 7: Ao imprimir o relatório, é escondido o cabeçalho e rodapé do sistema.

Na Figura 7, vemos como o relatório aparece quando é impresso. Note que ao imprimir a página o cabeçalho e rodapé são ocultados. É sugerido realizar esta impressão em um papel de etiquetas para que seja colado no bem a ser inventariado.



Figura 8: Como foi planejado e como ficou a tela de ajuda.

A Figura 8 acima mostra como foi planejado e como ficou a página de ajuda. Inicialmente foi adicionado um texto descrevendo como utilizar o leitor de código de barras. Futuramente, pretendo adicionar uma explicação de como integrar o sistema de inventário com o sistema de gestão da empresa, além de noções de como utilizar o sistema em modo *off-line* e como ocorre a sincronização dos dados com o sistema quando ele volta em modo *on-line*.



Figura 9: A tela de contato deixou de existir, e foi criada a tela mobile para acessar o sistema mobile.

Na Figura 9 vemos que foi planejada uma página para informações de contato. Essa página foi removida e a informação de contato aparece no rodapé da página. Por outro lado, foi adicionada uma página para acessar o sistema *mobile*, caso seja necessário realizar algum ajuste no sistema a partir da interface *Web*.

A segunda tela principal do sistema é o aplicativo móvel, diferente da tela anterior, após seu acesso inicial pelo aparelho, o aplicativo continua funcionando sem a necessidade de conexão com o servidor. As etiquetas *QR Code* possuem uma chamada especial desse sistema, que traz a tela de lançamento do inventário do bem pré digitado, com os dados do último lançamento realizado, caso existir, sendo necessário que seja digitado somente a complementação do que mudou do inventário anterior para o atual.

É possível ainda armazenar arquivos de fotos no sistema, porém as fotos são tiradas pelo aplicativo próprio do aparelho, então esse processo de tirar a foto e anexar no sistema pode ser complicado dependendo do navegador utilizado e versão do sistema operacional do aparelho. Na última versão do sistema Android por exemplo é possível chamar a câmera do aparelho diretamente pelo navegador, o que dá uma sensação melhor do que em outras configurações, onde é necessário

tirar a foto, salvar em alguma pasta, abrir novamente o aplicativo e finalmente anexar a foto.



Figura 10: Como foi planejado, e como ficou o menu principal.

Na Figura 10 vemos como foi planejado e como ficou o menu principal do sistema *mobile*. Note que o sub-menu movimentações foi retirado, já que não tem necessidade de ter um sub-menu para uma única operação (Lançamento de inventário).

Uma adição importante em todo o sistema é o indicativo se está trabalhando *on-line* ou *off-line* (Figura 11), posicionado no cabeçalho da aplicação.



Figura 11: Indicativo se a página está operando *on-line* ou *off-line*



Figura 12: Como foi planejado, e como ficou o menu de cadastros.

Na Figura 12 mostramos como foi planejado e como ficou o sub-menu de cadastros. Note que o botão voltar passou a se posicionar no cabeçalho da tela.



Figura 13: Como foi planejado, e como ficou a tela de cadastro de bem.

Na Figura 13 vemos como foi planejada e como ficou a tela de cadastro de bem. Note que foi adicionado um campo para filtrar os itens cadastrados, podendo ser digitado o código do sistema de gestão, ou o código do tombamento, ou ainda a descrição do bem para filtrar o cadastro e facilitar a localização para sua edição. Além disso os botões voltar e adicionar também foram posicionados no cabeçalho da aplicação.

The image shows a mobile application interface for managing assets. It is divided into two main sections: a header and a main content area. The header includes a 'Logo' and 'Cabeçalho' (Header) section on the left, and two buttons: 'Apagar' (Delete) and 'Salvar' (Save). The main content area is split into two modes: 'Adicionar bem' (Add asset) and 'Editar bem' (Edit asset). The 'Adicionar bem' mode shows a form with fields for system code, category code, acquisition date, disposal date, description, acquisition value, monthly depreciation, useful life, and asset type. The 'Editar bem' mode shows the same form with pre-filled values. The 'Voltar' (Back) button is also present in the header area.

Figura 14: Como foi planejado, e como ficou a tela de adicionar ou editar o bem.

Na Figura 14 mostramos como foi planejada, e como ficou a tela para adicionar ou editar o bem. Note que apesar de ser a mesma tela, seu comportamento muda dependendo da ação, mudando o título, e desabilitando o botão apagar. Além disso o botão voltar também foi posicionado no cabeçalho da aplicação.



Figura 15: Como foi planejado, e como ficou a tela de cadastro de inventário.

A tela da Figura 15, de listagem de inventário é similar ao cadastro de bens, e as modificações entre a versão planejada e finalizada são as mesmas.



Figura 16: Como foi planejado, e como ficou a tela de adicionar ou editar inventário.

Esta tela é similar a tela de editar ou adicionar o bem, e as modificações entre a versão planejada e finalizada são as mesmas.

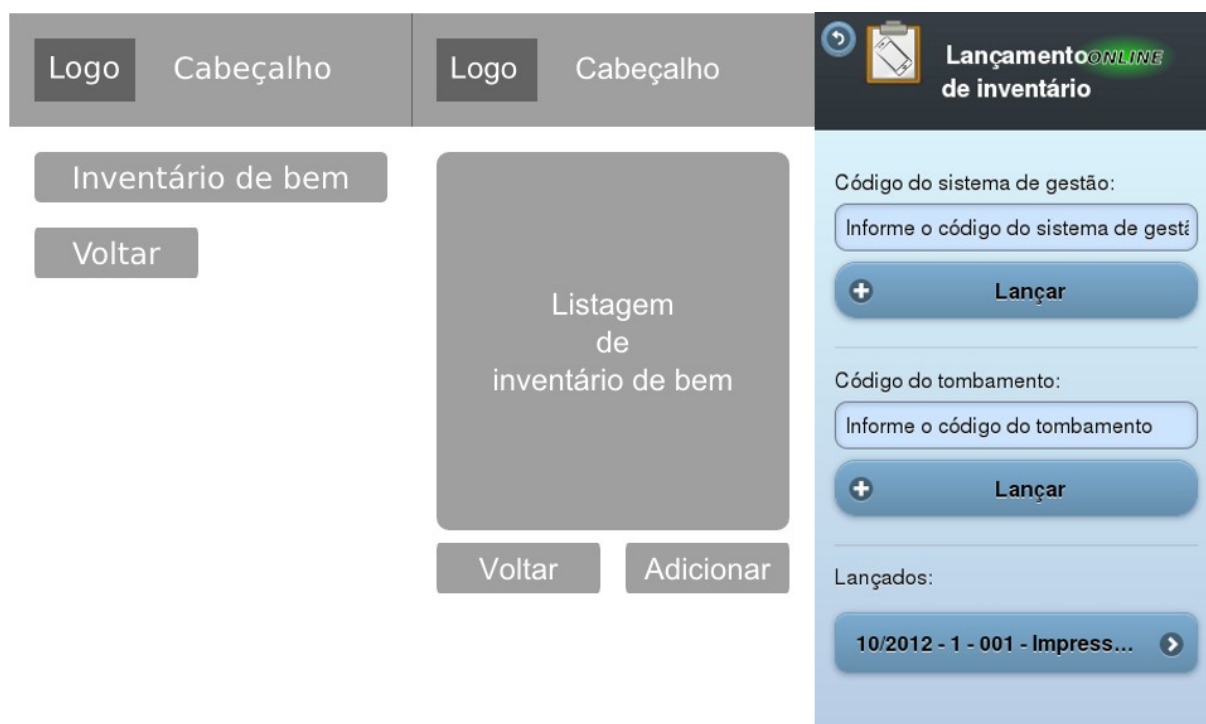


Figura 17: O menu de movimentação deixou de existir, ao invés dele é chamado o lançamento de inventário.

Na Figura 17 vemos a tela de lançamento de inventário. Inicialmente, foi planejada ter um sub-menu de movimentações para depois acessar a tela de lançamento de inventário. Este sub-menu foi retirado, já que o único item dele é o lançamento de inventário.

Outro detalhe é esta tela inicialmente se chamava inventário de bem, e posteriormente foi modificado para lançamento de inventário por fazer mais sentido ao contexto onde a aplicação será utilizada, e para que seu nome se diferencie mais da tela de cadastro de inventário.

Ainda no planejamento não foi notada a necessidade de ser uma tela diferente dos demais cadastros, mas após os testes da aplicação, foi percebida uma necessidade de modificação na interface, trazendo além da listagem dos últimos lançamentos, a possibilidade de digitar o código do sistema de gestão ou o código do tombamento para realizar o inventário, tornando possível utilizar um dispositivo desprovido de câmera ou *scanner* de código de barras, através da digitação do código do bem a ser inventariado.

Logo Cabeçalho

Dados do inventário de bem

Apagar Salvar

Voltar

Adicionar ONLINE lançamento de inventário

Editar ONLINE lançamento de inventário

- Geral

Inventário:
10/2012 - Inventário d... ▼

Bem:
2 - 002 - Computador ▼

Data do levantamento:
2012-11-07

Estado de conservação:
Informe o estado de conservação

Nome do responsável:
Informe o nome do responsável

+ Medidas

+ Valores

+ Localização

+ Fotos

- Geral

Inventário:
10/2012 - Inventário d... ▼

Bem:
1 - 001 - Impressora ▼

Data do levantamento:
2012-11-01

Estado de conservação:
bom

Nome do responsável:
thiago

+ Medidas

+ Valores

+ Localização

+ Fotos

Apagar Salvar

Apagar Salvar

Figura 18: Como foi planejado, e como ficou a tela principal de adicionar ou editar lançamento de inventário.

Na Figura 18 vemos como foi planejada, e como ficou a tela de lançamento de inventário. Note que as informações do lançamento de inventário são divididas em 5 categorias, facilitando o lançamento das informações.

The figure displays three sequential screens for adding an inventory item, each titled "Adicionar lançamento de inventário ONLINE".

- Screen 1 (Left):** Features a "Medidas" section with input fields for "Largura" (Informe a largura), "Altura" (Informe a altura), and "Profundidade" (Informe a profundidade). It also has a "Características físicas" field (Informe as características físicas) and expandable sections for "Valores", "Localização", and "Fotos".
- Screen 2 (Middle):** Features a "Valores" section with input fields for "Valor atual calculado" (Informe o valor atual calculado), "Depreciação acumulada" (Informe a depreciação acumulada), and "Valor atual comprovado" (Informe o valor atual comprovado). It also has a "Documento comprobatório" field with a "Browse..." button and expandable sections for "Localização" and "Fotos".
- Screen 3 (Right):** Features a "Localização" section with a "Descrição" field (Informe a descrição da localização), a "Centro de custo" dropdown menu (Informe o centro de c...), and a "Latitude e longitude" field with an "Atualizar latitude e lo..." button. It also has an "Adicionar foto" button and expandable sections for "Fotos".

Each screen includes a "Salvar" (Save) button and an "Apagar" (Delete) button at the bottom.

Figura 19: Demais telas para adicionar um lançamento de inventário.

Acima (Figura 19) como são apresentadas as demais categorias do lançamento de inventário no momento de criar um novo inventário.

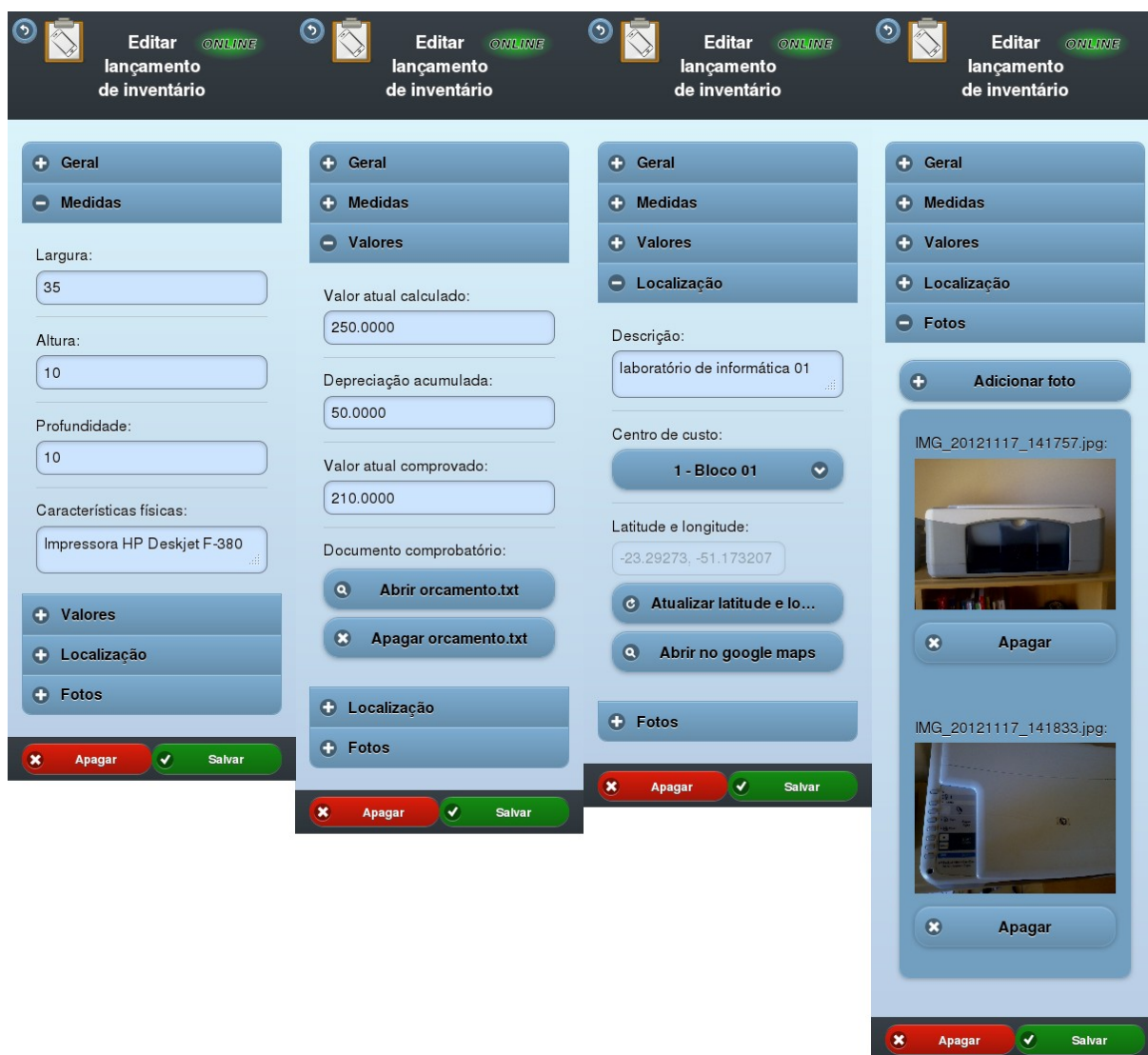


Figura 20: Demais telas para editar um lançamento de inventário.

Na Figura 20 vemos como são apresentadas as categorias do lançamento de inventário no momento de edição dos dados já lançados, para ter uma noção melhor de como a tela aparece ao lançar um inventário.

Como pode ser percebido na imagem, é possível adicionar várias fotos ao invés de uma única, para isso só é necessário clicar no botão adicionar foto, e é aberto um novo campo para inserir a nova foto.

Ao anexar uma imagem na aplicação, é realizado um processo de compactação da imagem, para que ela ocupe menos espaço na memória do dispositivo móvel. Esse processo é necessário para evitar que apareçam mensagens do navegador requisitando permissão para guardar mais informações na memória do navegador, já que existe um limite de 5 *Megabytes* de informações que podem ser guardadas por sessão.

3.2.4 Diagrama entidade relacionamento

O diagrama entidade relacionamento mostra como foi criada a estrutura do sistema no banco de dados, facilitando o entendimento de onde as informações são guardadas, e quais os relacionamentos entre elas.

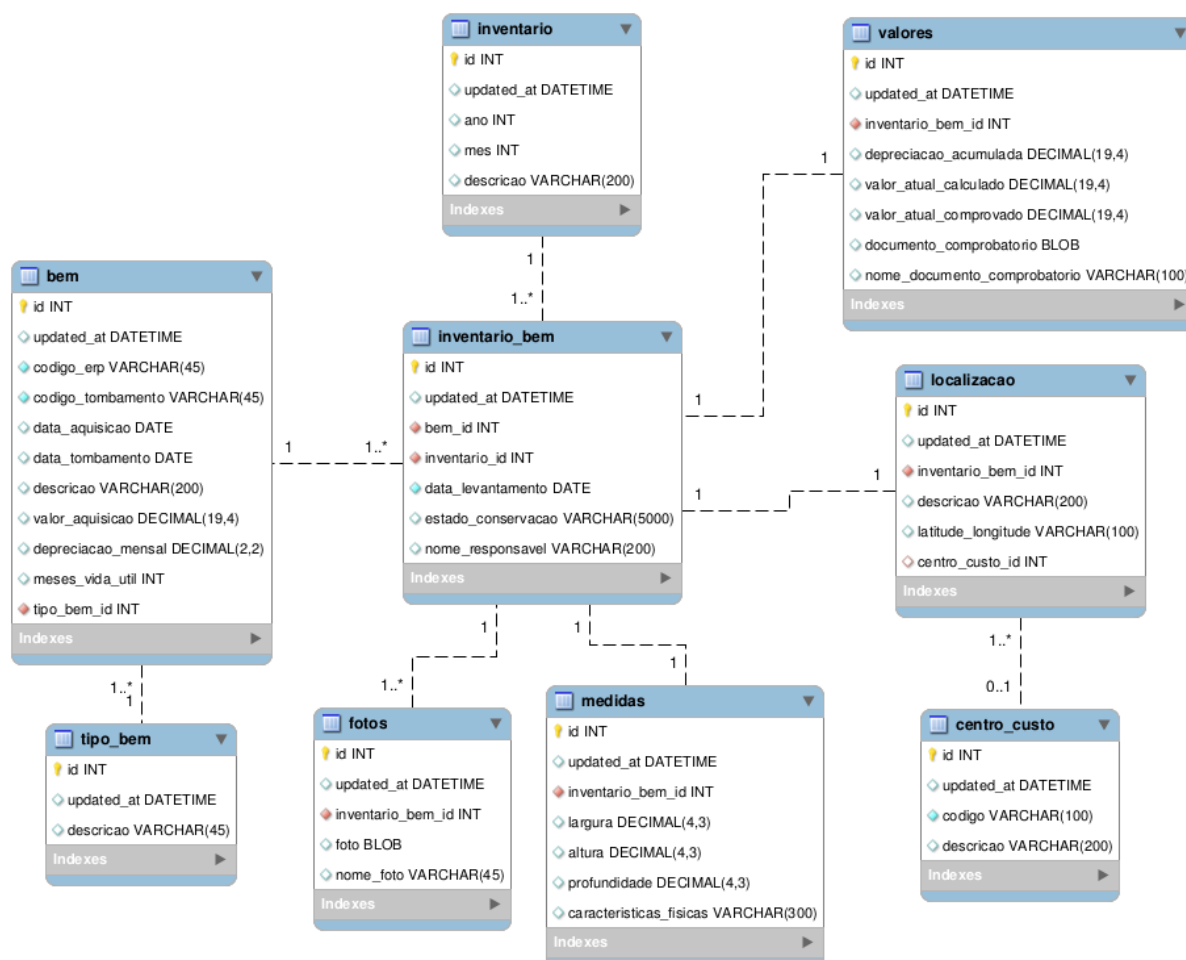


Figura 21: Diagrama entidade relacionamento.

3.2.5 Diagrama de classes

O diagrama de classes é o diagrama UML estrutural do sistema equivalente ao diagrama entidade relacionamento do banco de dados. Nele

podemos notar quais são as classes do sistema, e quais os relacionamentos entre elas. A estrutura desse diagrama reflete a regra de negócio do sistema.

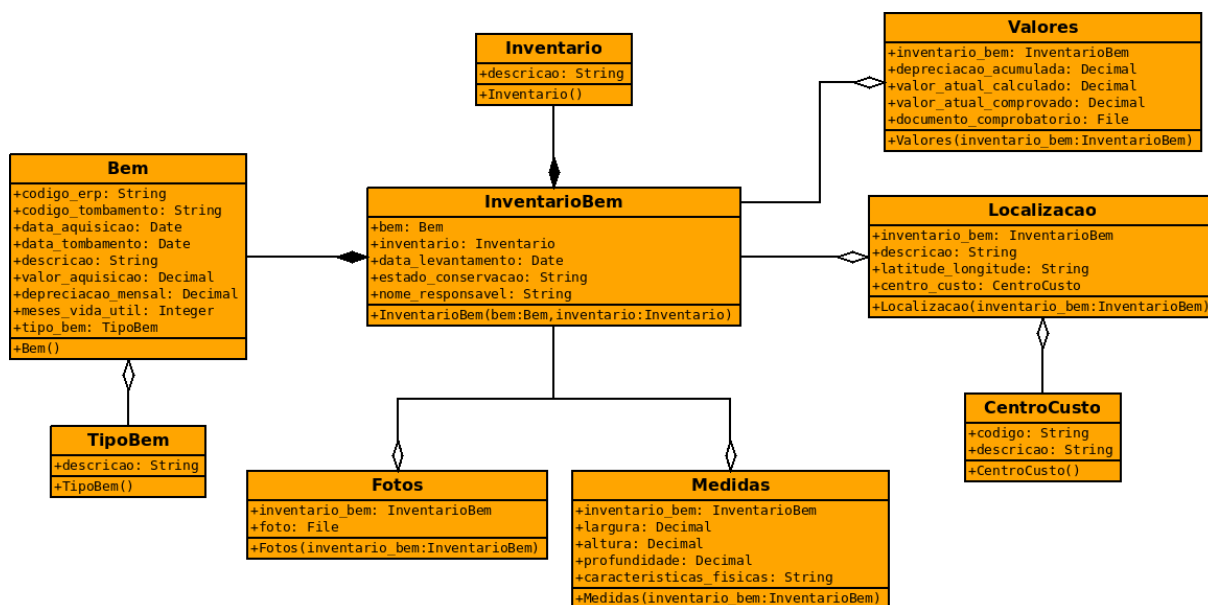


Figura 22: Diagrama de classes.

3.2.6 Diagrama de atividades

O diagrama de atividades serve para mostrar o fluxo de atividades de um processo, e como uma atividade depende da outra.

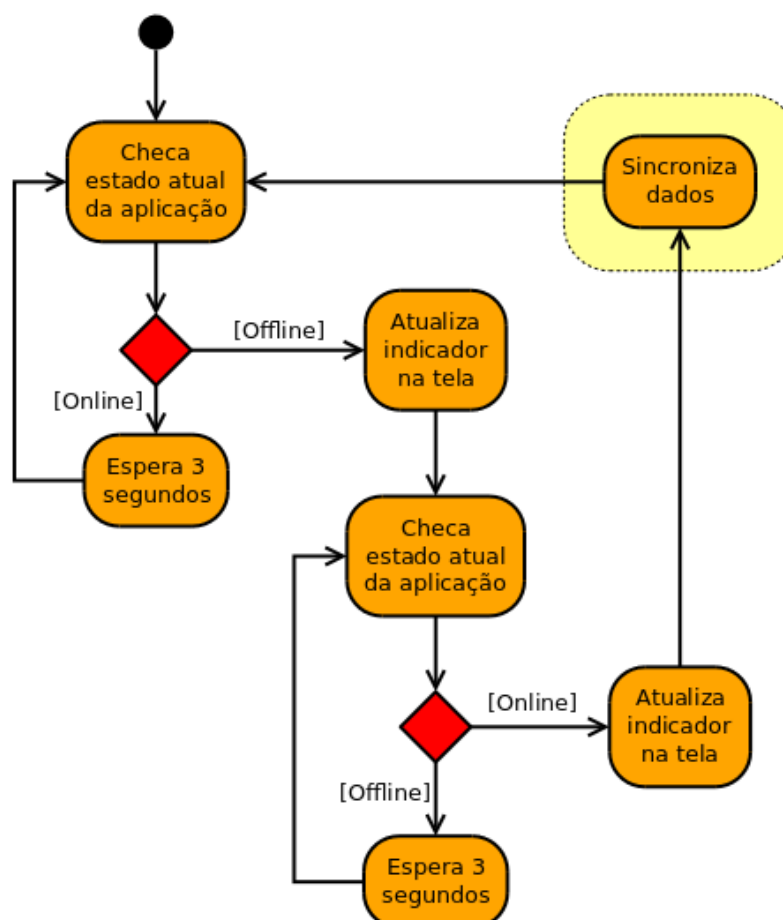


Figura 23: Diagrama de atividades - Checagem e sincronização das informações no sistema mobile.

No diagrama de atividade acima é demonstrado como e em que condição ocorre a sincronia de dados do sistema, sendo que somente na transição entre estar com a aplicação em modo *off-line* para o modo *on-line* é realizada a sincronização de dados com o servidor.

3.2.7 Diagrama de comunicação

Os diagramas de comunicação mostram como os objetos do sistema interagem, descrevendo como ocorre detalhadamente alguma ação do sistema em tempo de execução.

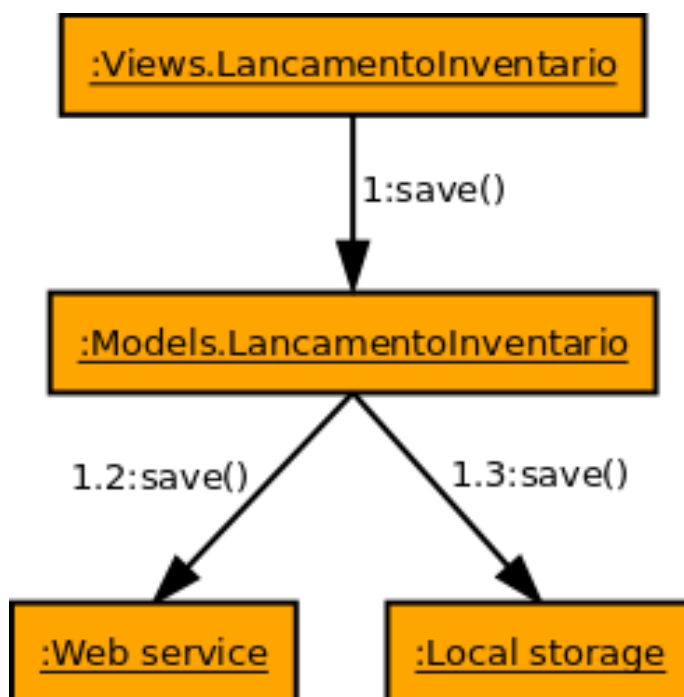


Figura 24: Diagrama de comunicação - Salvar dados no sistema.

O diagrama de comunicação acima mostra que ao executar a ação de salvar dados nos objetos de modelo do sistema, ele dispara a gravação dos dados tanto no *Web Service* quanto no *Local Storage* do navegador, sendo que pode acontecer do *Web Service* não estar acessível, e as informações são persistidas somente no *Local Storage*. Caso isso ocorra, posteriormente, quando a aplicação entrar em modo *on-line* os dados que não foram persistidos no *Web Service* são re-transmitidos na sincronização.

3.3 COMO O SISTEMA FOI IMPLEMENTADO

Como já mencionado anteriormente, a aplicação foi dividida em duas grandes frentes, a primeira com algumas páginas simples e *Web Services* responsáveis por persistir os dados no banco de dados e conversar com o sistema gerencial da empresa, enquanto a segunda frente foi responsável por funcionar de forma *off-line* para coletar os dados durante o inventário, e sincronizar com o servidor assim que perceber que está conectado na rede interna da empresa, e consegue acessar novamente o servidor.

Para atingir esse objetivo, na primeira frente foi utilizado como linguagem de programação a linguagem Python, junto com o *framework* MVC (*Model View Controller*) Pyramid, que possui algumas visões de páginas normais, e outras que atuam como um *Restful Web Service*, além do mapeador objeto-relacional SQLAlchemy para persistência dos dados.

Na segunda frente, foi utilizada a linguagem de programação Coffeescript, junto com o *framework* MVP (*Model View Presenter*) Backbone.js.

Com essas tecnologias, foi criada a estrutura básica para solucionar o problema proposto nesse trabalho, mas além disso, existiu alguns requisitos não funcionais como a necessidade de fazer os sistema utilizar um *layout* que se ajuste a diversos aparelhos, além de ser otimizado para não consumir muita rede, pensando primeiro em um ambiente móvel.

Para conseguir estes outros objetivos, foi utilizada a aplicação webassets, com os filtros cssmin, uglifyjs e gzip responsáveis pela compressão e mesclagem dos arquivos Javascript e CSS, além do pré-processador Sass (*Syntactically Awesome Stylesheets*), e bibliotecas Compass e Susy que facilitam a criação do estilo e um *layout* responsivo para funcionar para vários dispositivos.

3.3.1 Python

A linguagem Python é uma linguagem de programação multi paradigma, ou seja, é possível criar estruturas de programação procedural, orientadas a objetos, orientadas a aspecto, além de possuir algumas características das linguagens funcionais (PYTHON SOFTWARE FOUNDATION, 2012).

Sua característica mais marcante é a legibilidade e facilidade de manutenção no código, podendo ser utilizada para desenvolver tanto aplicações *desktop* quanto aplicações *web*.

3.3.2 Pyramid

Pyramid é um *framework* MVC para aplicações *web*. Assim como outros *frameworks* MVC inspirados em Ruby on Rails, ele traz um *workflow* simples e dinâmico, dando mais velocidade ao desenvolvimento das aplicações, e como

diferencial das outras soluções MVC escritas em Python, ele foca em dar a estrutura MVC, mas não prende o desenvolvedor em alguma ferramenta, ou seja, é possível que em um projeto seja utilizada para a gravação das informações o framework SQLAlchemy, enquanto em outro projeto utilize o banco ZODB, ao mesmo tempo existe a possibilidade de escolher entre vários gerenciadores de *templates*, por exemplo: Chameleon; Mako templates; Jinja2; Mustache, além de dar possibilidade de substituir a ferramenta para mapear URLs *transversal* pelo *URL Dispatcher*. Na prática funciona como se fosse um quebra cabeças, onde pode ser montada a combinação de aplicativos que mais se aproxima da necessidade da aplicação (PYLONS PROJECT, 2012).

Além de tudo isso, Pyramid traz consigo um conjunto de ferramentas para dar suporte ao desenvolvimento, como uma ferramenta para *debug* embutida em suas páginas, com a possibilidade de inspecionar e analisar o código assim que uma exceção é disparada enquanto estiver desenvolvendo a aplicação (PYLONS PROJECT, 2012).

3.3.3 SQLAlchemy

O SQLAlchemy é um mapeador objeto-relacional (ORM - *Object Relational Mapper*), ou seja, uma ferramenta que esconde as diferenças de paradigmas entre a linguagem SQL (*Structured Query Language*), e o paradigma orientado a objetos, podendo facilitar o desenvolvimento de aplicações e aumentar a velocidade, já que o desenvolvedor não precisa mudar seu modo de pensar sobre o problema, além de aproveitar os recursos de reutilização de código existentes na orientação a objetos (SQLALCHEMY, 2012).

Além do mapeamento objeto-relacional, o SQLAlchemy cria uma camada de abstração entre os banco de dados, dando a possibilidade de migrar de um sistema gerenciador de banco de dados (SGBD) para outro sem dificuldades (SQLALCHEMY, 2012).

3.3.4 Coffeescript

Coffeescript é uma linguagem de programação simples e atual, trazendo as facilidades de programação existentes nas linguagens mais atuais porém com um diferencial: ao ser compilada essa linguagem gera código Javascript (BATES, 2012).

Dessa forma, o desenvolvimento de aplicações que funcionam diretamente no navegador, e inevitavelmente precisam utilizar Javascript, se torne mais rápida e menos propensa a erros, já que o código gerado por Coffeescript utiliza os melhores padrões Javascript, além de funcionar com qualquer biblioteca escrita para funcionar em Javascript (BURNHAM, 2012).

3.3.5 Backbone.js

Os sites de internet geralmente possuem uma divisão no processamento das informações, metade do processamento da aplicação *Web* ocorre no servidor, e a outra metade acontece no navegador de internet, também denominado lado cliente da aplicação.

Geralmente a parte principal das aplicações *Web*, é criada no lado servidor da aplicação, enquanto no lado cliente se concentra uma lógica mais simples, responsável pela interação entre o usuário do site e a sua interface.

Após a criação de *smartphones* e *tablets*, esse cenário começou a mudar, e existe uma tendência para que aplicações *Web* fiquem cada vez menos triviais do lado cliente da aplicação. Uma das mudanças é que a conexão com o servidor pode ser mais lenta do que as aplicações *Web* tradicionais, ou ainda pode ocorrer de não existir conexão com o servidor em determinado momento, e a aplicação funcione em modo *off-line*, como é o caso da aplicação criada neste trabalho.

Para que esse tipo de aplicação se torne possível é necessário replicar a estrutura da aplicação que antes existia somente do lado servidor para o lado cliente, o que torna o lado cliente cada vez mais complexo.

Com o objetivo de lidar com essa complexidade extra no lado do cliente, foi criado o framework Backbone.js, que traz as peças fundamentais para

que seja possível criar uma aplicação que funcione tanto em modo *on-line* quanto *off-line*, além de realizar a sincronização dos dados do lado cliente com o servidor quando necessário (OSMANI, 2012).

Ao invés de utilizar o tradicional MVC encontrado em aplicações mais comuns, o Backbone.js utiliza o padrão MVP (*Model View Presenter*). O MVP foi originado pela empresa Taligent no início da década de 1990 para ser utilizado com a linguagem C++, e sua diferença com o MVC é que a visão do MVP tem tanto a lógica da visão quanto a lógica do controlador encontrado no MVC, e o *presenter* do MVP não passa de uma visão burra, ou seja, no MVC a visão conversa tanto com o controlador quanto com o modelo, observando alterações no modelo para se atualizar, enquanto no MVP, a visão (*presenter*) somente conversa com o controlador (*view*), e reflete o seu estado atual (OSMANI, 2012).

Os criadores do Backbone.js construíram sua estrutura utilizando o MVP ao invés do MVC por acreditarem que esse modelo se encaixa melhor no contexto de aplicações web (OSMANI, 2012).

Além de dar a estrutura necessária para aplicações robustas que funcionam diretamente no navegador, Backbone.js pode trabalhar em modo *off-line*, persistindo seus dados no *Web Storage* do navegador, ou ainda sincronizar com *Restful Web Services*, o que veio exatamente de encontro com a necessidade deste projeto (BACKBONEJS, 2012).

3.3.6Sass

Um pré-processador CSS traz a vantagem de reutilização de código e utilização de variáveis de uma linguagem de programação na criação de layouts de páginas. Sass (CATLIN, 2012) é um dos pré-processadores CSS mais conhecidos atualmente principalmente pela biblioteca Compass (EPPSTEIN, 2012), que dá ao desenvolvedor a possibilidade de utilizar recursos CSS3 de uma forma mais simples, já que ainda é necessário escrever as *vendor prefix* enquanto CSS3 não se torna um padrão, além disso, Compass possui um conjunto de bibliotecas que reúne as melhores práticas de desenvolvimento *web* em um formato que pode ser reutilizado facilmente (CATLIN, 2011).

Outra biblioteca escrita em Sass é o Susy, que dá a opção de utilizar grids fluidas, de tamanho fixo, e grids que misturam os dois, sendo de tamanho fixo em telas grandes, e fluido em telas menores. O Susy segue uma linha diferente de outros sistemas de grids, ele foi baseado na teoria exposta na apresentação *CSS Systems* de Natalie Downe, que sugere que os layouts das páginas devem ser os mesmos para todos os navegadores, ao contrário do que a outros sistemas de grids, que costumam trazer um arquivo css separado para o navegador internet explorer, ou utilizar comentários condicionais para fazer os ajustes para este navegador (DOWNE, 2012). Segundo o site principal do Susy, “essa implementação só foi possível graças ao pré-processador Sass, e foi tornada mais fácil com a utilização do Compass” (MEYER, 2012).

Mesmo tendo um layout totalmente fluido, para manter a legibilidade em dispositivos com tela pequena é interessante mudar o layout para que os elementos da página se acomodem melhor em telas pequenas, por isso com Susy é possível utilizar demarcações de acordo com o tamanho da tela para re-arranjar o layout, através de *media queries* css, conseguindo um layout responsivo (MEYER, 2012).

3.3.7 Webassets

Para gerenciar a geração automática de arquivos Javascript a partir de arquivos Coffeescript, a geração de arquivos CSS a partir de arquivos Sass, mesclar diversos arquivos em um só, minificar e comprimir, foi utilizado a ferramenta Webassets (ELSDÖRFER, 2012).

Webassets é um gerenciador de arquivos estáticos para sistemas web, com ele é possível criar um conjunto de regras para manipulação de arquivos, e associar a dois modos de operação, o modo de *debug* e o modo normal. No modo de *debug*, só são aplicadas as regras que são necessárias para fazer a página funcionar, mas não as regras que deixam o código gerado ilegível, como as regras para minificar o código e comprimir, dessa forma é possível inspecionar e encontrar problemas na aplicação facilmente (ELSDÖRFER, 2012).

Além disso, webassets pode funcionar sozinho, ou ainda se integrar a *frameworks web* escritos em Python, como o Pyramid, trazendo um conjunto de

regras e integração com aplicativos como o Coffeescript, Sass e Compass (ELSDÖRFER, 2012).

A configuração do webassets utilizada neste projeto, fez com que os arquivos javascript e css fossem mesclados em dois únicos arquivos, minificados e compactados, o que diminui as requisições ao carregar a página, possibilitando que a página carregue mais rápido em aparelhos móveis utilizando uma conexão lenta.

4 CONCLUSÃO

Esta monografia descreveu um estudo de tecnologias abertas e padrões web para facilitar e trazer novas possibilidades ao processo de inventário de patrimônio.

Através da aplicação criada, foi possível eliminar a necessidade de anotar informações em pranchetas de papel para depois ser digitado e conferidas as informações no computador. Além disso, o processo fica mais simples, pois a partir do segundo inventário as informações do inventário anterior já são carregadas no sistema, necessitando somente de uma conferência das informações lançadas no inventário anterior. Outro objetivo alcançado foi o de poder armazenar fotos do bem durante o inventário, conseguindo informações ricas sobre o estado do bem.

Utilizando as técnicas introduzidas na versão 5 da linguagem HTML, foi possível criar uma aplicação que funciona tanto conectado no servidor quanto desconectado, conseguindo um ambiente propício para a realização de inventários, sem perder a flexibilidade das páginas de internet, que podem funcionar em qualquer dispositivo móvel atual.

As etiquetas de código de barras introduzidas nesse trabalho trouxeram facilidade para integrar o mundo físico com o mundo digital dos aparelhos móveis, e o Backbone.js provou ser uma biblioteca robusta para armazenar as informações *off-line* e realizar a sincronia de dados com o servidor, quando re-conectado a rede.

Como trabalho futuro, sugiro implementar uma aplicação com recursos similares a esta, porém para solucionar outros problemas, procurando primeiramente atualizar a pesquisa dos recursos existentes, já que existem diversas formas de criar aplicações *Web* que funcionam *off-line*, e a tendência é que cada vez surjam soluções mais simples, sabendo que muitas aplicações precisam solucionar este mesmo problema.

Outra sugestão de trabalho futuro é de realizar um estudo para melhoria da interface do sistema. Durante esse trabalho não foi dada a devida atenção a interface, alguns botões ficaram muito pequenos para a interface touch, e algumas telas poderiam ser remodeladas utilizando uma navegação passo a passo (*step by step navigation*).

REFERÊNCIAS

AFIXCODE. **Inventário Patrimonial**. Disponível em: <<http://www.afixcode.com.br/servicos/inventario-patrimonial/>>. Acesso em: 11 abr. 2012.

BACKBONEJS. **Backbone.js**. Disponível em: <<http://backbonejs.org/>>. Acesso em: 12 out. 2012.

BATES, Mark. *Programming in CoffeeScript*. Addison Wesley. 2012. 283p.

BIDELMAN, Eric. **A Beginner's Guide to Using the Application Cache**. Disponível em: <<http://www.html5rocks.com/en/tutorials/appcache/beginner/>>. Acesso em: 20 jul. 2012.

BURNHAM, Trevor. **CoffeeScript – Accelerated Javascript Development**. The Pragmatic Bookshelf. 2011. 129p.

CATLIN, Hampton; CATLIN, Michael Lintorn. **Pragmatic Guide to Sass**. The Pragmatic Bookshelf. 2011. 112p.

CATLIN, Hampton; WEIZENBAUM, Nathan; EPPSTEIN, Chris. SASS, Sass. **Syntactically Awesome Stylesheets**. Disponível em: <<http://sass-lang.com/>>. Acesso em: 12 out. 2012.

CHRISTIAN, Mark; DIAZ, Dustin; LUBBERS, Peter. **Appcache Facts**. Disponível em: <<http://appcachefacts.info/>>. Acesso em: 20 jul. 2012.

COFFEESCRIPT. **CoffeeScript is a little language that compiles into JavaScript**. Disponível em: <<http://coffeescript.org/>>. Acesso em: 12 out. 2012.

CONSELHO FEDERAL DE CONTABILIDADE. **NBC T 3.1**, dez. 1990. Disponível em: <http://www.cfc.org.br/sisweb/sre/docs/RES_686.doc>. Acesso em: 01 set. 2012.

CONSELHO FEDERAL DE CONTABILIDADE. **NBC T 19.1**, abr. 2005. Disponível em: <http://www.cfc.org.br/sisweb/sre/docs/RES_1025.doc>. Acesso em: 01 set. 2012.

DENSO WAVE INCORPORATED. **QR Code Features**. Disponível em: <<http://www.qrcode.com/en/qrfeature.html>>. Acesso em: 20 jul. 2012.

DIAS, Célio Sales. **A Importância da Contabilidade e do Balanço Patrimonial nas Empresas**, sep. 2011. Disponível em: <<http://gestaofitness.wordpress.com/tag/celio-dias>>. Acesso em: 11 jun. 2012.

DIXIT, Shwetank. **Running your web applications offline with HTML5 AppCache**. Disponível em: <<http://dev.opera.com/articles/view/offline-applications-html5-appcache/>>. Acesso em: 20 jul. 2012.

DIXIT, Shwetank. **Web Storage: easier, more powerful client-side data storage**. Disponível em: <<http://dev.opera.com/articles/view/web-storage/>>. Acesso em: 20 jul. 2012.

DOWNE, Natalie. **CSS Systems**. Disponível em: <<http://www.slideshare.net/nataliedowne/css-systems-presentation>>. Acesso em: 12 out. 2012.

ELSDÖRFER, Michael. **Asset management application for Python web development**. Disponível em: <<http://webassets.readthedocs.org/en/latest/>>. Acesso em: 12 out. 2012.

EPPSTEIN, Christopher M. **Compass is an open-source CSS Authoring Framework**. Disponível em: <<http://compass-style.org/>>. Acesso em: 12 out. 2012.

FIPECAFI/USP. **Contabilidade Introdutória**. 10 ed. São Paulo: Atlas, 2006.

HICKSON, Ian. **HTML Living Standard**. Disponível em: <<http://www.whatwg.org/specs/web-apps/current-work/multipage>>. Acesso em: 20 jul. 2012.

IBRACON, Instituto dos Auditores Independentes do Brasil. **IBRACON NPC nº 27 – DEMONSTRAÇÕES CONTÁBEIS**. Disponível em: <<http://www.portaldecontabilidade.com.br/ibracon/npc27.htm>>. Acesso em: 11 jun. 2012.

INFOMONEY. **Balço patrimonial: ferramenta importante do planejamento financeiro**. Disponível em: <<http://www.infomoney.com.br/minhas-financas/noticia/430679>>. Acesso em: 11 jun. 2012.

INTERNATIONAL STANDARD. **ISO/IEC 18004**, jun. 2000. Disponível em: <http://raidenii.net/files/datasheets/misc/qr_code.pdf>. Acesso em: 02 mar. 2013.

MEYER, Eric A. **Responsive grids for Compass**. Disponível em: <<http://susy.oddbird.net/>>. Acesso em: 12 out. 2012.

MACCAW, Alex; ASHKENAS, Jeremy. **The Little Book on CoffeeScript**. O'REILLY. 2012. 45p.

OSMANI, Addy. **Developing Backbone.js Applications**. O'REILLY. 2012. 145p.

PILGRIM, Mark. **Geolocation**. Disponível em: <<http://diveintohtml5.info/geolocation.html>>. Acesso em: 20 jul. 2012.

PILGRIM, Mark. **The Past, Present & Future of Local Storage for Web Applications**. Disponível em: <<http://diveintohtml5.info/storage.html>>. Acesso em: 20 jul. 2012.

PYLONS PROJECT. **Pyramid Documentation**. Disponível em: <<http://docs.pylonsproject.org/en/latest/docs/pyramid.html>>. Acesso em: 12 out. 2012.

PYTHON SOFTWARE FOUNDATION. **Python programming language**. Disponível em: <<http://www.python.org/>>. Acesso em: 12 out. 2012.

REFSNES DATA. **HTML5 Geolocation**. Disponível em: <http://www.w3schools.com/html/html5_geolocation.asp>. Acesso em: 20 jul. 2012.

REFSNES DATA. **HTML5 Web Storage**. Disponível em: <http://www.w3schools.com/html5/html5_webstorage.asp>. Acesso em: 20 jul. 2012.

RELIASOFT CORPORATION. **Orion eAPI**. Disponível em: <<http://www.reliasoft.com.br/orion/index.htm>>. Acesso em: 11 abr. 2012.

RICHARDSON, Leonard; RUBY, Sam. **RESTful Web Services**. O'REILLY. 2012. 440p.

ROBINSON, Mike. **Go offline with application cache**. Disponível em: <<http://html5doctor.com/go-offline-with-application-cache/>>. Acesso em: 20 jul. 2012.

SHEPHERD, Eric et al. **Using the application cache**. Disponível em: <https://developer.mozilla.org/en-US/docs/HTML/Using_the_application_cache>. Acesso em: 20 jul. 2012.

SILVA, José Pereira da. **Contabilidade Aplicada ao Setor Público, mudança cultural acima de tudo**, out. 2011. Disponível em:

<<http://www.agili.com.br/editoriais/contabilidade-aplicada-ao-setor-publico-mudanca-cultural-acima-de-tudo-30.aspx>>. Acesso em: 01 set. 2012.

SILVA, José Pereira da et al. **A justa mensuração do Patrimônio Público Brasileiro**, dez. 2011. Disponível em: <<http://www.agili.com.br/editoriais/a-justa-mensuracao-do-patrimonio-publico-brasileiro-32.aspx>>. Acesso em: 01 set. 2012.

SOFTEXPERT SOFTWARE FOR PERFORMANCE EXCELLENCE. **SE Asset**. Disponível em: <<http://www.softexpert.com.br/gestao-ativos.php>>. Acesso em: 11 abr. 2012.

SQLALCHEMY. **The Database Toolkit for Python**. Disponível em: <<http://www.sqlalchemy.org/>>. Acesso em: 12 out. 2012.

W3C, World Wide Web Consortium. **Geolocation API Specification**. Disponível em: <<http://dev.w3.org/geo/api/spec-source.html>>. Acesso em: 20 jul. 2012.

WINPATRIMONIO. **WinPatrimonio**. Disponível em: <<https://www.sistemascontabeis.com.br/Livre/SistemaDeControlePatrimonial.aspx>>. Acesso em: 11 abr. 2012.

WROBLEWSKI, Luke. **Mobile First**. A Book Apart. 2012. 138p.

APÊNDICE I – ENTREVISTA SOBRE INVENTÁRIO DE PATRIMÔNIO

Para conseguir um embasamento maior no que é e para que serve um Inventário de Patrimônio, foi realizada a seguinte pesquisa com Contadores para trazer uma qualidade maior nas informações apresentadas no trabalho. Seguem as perguntas e respostas na íntegra da pesquisa:

1. Qual o seu nome completo?

R1: Andréa Carla Heim de Castro

R2: José Pereira da Silva

R3: Marlau Adolfo da Silva

2. Qual o seu número de CRC?

R1: PR-36035/O

R2: 6100/O-5

R3: PR050903/O-5

3. Para que serve a Gestão Patrimonial?

R1: A Gestão Patrimonial serve para administração dos bens da empresa.

R2: O patrimônio imobilizado para o poder público ou qualquer outro segmento refere-se a grande parte do seu ativo, no entanto, historicamente o poder público não tem dado a atenção necessária para este assunto.

A Gestão Patrimonial bem efetuada apresenta uma condição ao gestor de mensuração correta do seu ativo patrimonial, esta prática vem de encontro à própria evolução da contabilidade governamental, que caminha para a adoção integral do regime de competência instituído pelas NBCASP(Normas brasileiras de contabilidade aplicada ao setor público). (SILVA, out, 2011)

Portanto o controle correto e apropriado dos bens patrimoniais trará principalmente a valoração correta dos ativos da entidade e com isso representará de maneira fidedigna os resultados patrimoniais.

R3: Controlar e gerir o imobilizado da empresa, saber o que se tem, onde esta, a situação financeira de cada ativo, acompanhar o desgaste, prever e programar reposições, manutenções, minimizando custos, e evitando que se pare uma produção por dano no equipamento.

4. Para que serve o Balanço Patrimonial, e qual a relação entre Gestão Patrimonial e Balanço Patrimonial?

R1: *O Balanço Patrimonial representa a posição patrimonial e financeira de uma empresa.*

R2: *O Balanço Patrimonial tem como principal função representar o ativo e passivo da entidade de forma consolidada e comparativa. Este documento deve demonstrar a real situação da entidade, onde será possível identificar seus saldos positivos e negativos, efetuando um confronto entre eles e apurando um resultado que poderá ser superavitário ou deficitário.*

A gestão patrimonial (SILVA, dez, 2011), se bem executada, fará com que o grupo de contas que representam o imobilizado, do Balanço patrimonial, tenha a sua demonstração de forma coerente com a realidade e conseqüentemente trará maior precisão ao resultado patrimonial, ao passo que uma gestão patrimonial precária trará resultados imprecisos e não confiáveis levando o gestor a tomar possíveis decisões em dados inverídicos trazendo maiores prejuízos para a organização.

R3: *Balanço patrimonial ira demonstrar a relação ente bens e direitos de uma empresa com suas obrigações, e uma das principais peças contábeis para análise. Gestão patrimonial e entendi o termo na pesquisa, seria gerir as informações Patrimoniais que estão demonstradas no balanço patrimonial, uma esta implícita na outra.*

5. Para que serve o Inventário de Patrimônio, e qual a relação entre Balanço Patrimonial e Inventário de Patrimônio?

R1: *O Inventário de Patrimônio serve para verificar a existência física dos bens e sua condição de uso.*

R2: *O inventário do patrimônio serve para evidenciar analiticamente os bens patrimoniais, demonstrando suas principais características e localização, bem como seus valores individuais e atualizados. Este documento servirá comprovar a veracidade dos valores registrados no Balanço patrimonial, no grupo imobilizado. De modo que, uma possível auditoria provavelmente irá confrontar os valores registrados no Balanço patrimonial com o inventário físico e financeiro dos bens e, caso haja compatibilidade, inclusive em relação à localização geográfica do bem, ficará evidenciada a regularidade do lançamento trazendo exatidão à peça contábil.*

R3: *O Inventario do Patrimônio, serve para que a empresa, saiba o que ela tem no imobilizado, e saiba onde e quais as condições desses bens.*

6. Na prática, quais informações costumam ser coletadas em um Inventário de Patrimônio?

R1: Coleta-se o número de identificação constante da plaqueta de identificação, a localização do bem, sua descrição e sua condição.

R2: Na elaboração de um inventário, o objetivo é trazer transparência e exatidão quanto à mensuração e localização dos bens patrimoniais, desta forma, os principais dados que devem ser coletados são: tipo do bem (móvel, imóvel, etc), descrição completa do bem, localização física exata (com todos os níveis necessários, inclusive com respectivas coordenadas), valor atual com referência a documento comprobatório, data de aquisição, imagens, medidas, características físicas, origem do bem, classificação do bem por grupos com diversos níveis, estado de conservação, pessoa responsável pelo bem, data do levantamento, enfim, quanto maior precisão de informações, mais facilidade trará para a identificação.

R3: Numero do bem, localização do bem, estado de conservação do bem. Pois valores e depreciação já estão devidamente lançados no sistema de controle.

7. O Sr.(a) utiliza algum sistema informatizado que gerencia essas informações? Qual o nome desse sistema?

R1: Utilizamos um sistema informatizado para controle dos bens. O sistema foi desenvolvido em progress.

R2: Sim, utilizo sistema de empresa localizada na cidade de Londrina com abrangência em todo território nacional.

R3: Não, por trabalhar em escritório de contabilidade, o gerenciamento do Patrimônio, se da na empresa, não participamos do mesmo, é muito comum nesses casos, contabilmente se ter um valor patrimonial muito diferente do que de fato se tem no físico.

8. Existiria algum problema ou recursos que seu sistema não resolve ou suporta que você gostaria que fosse contemplado?

R1: Nossas placas de identificação poderiam conter um código de barras para captura por leitor no momento do inventário.

R2: Gostaria que possuísse módulo Web com recursos para coleta de informações em campo, com possibilidade de captação de dados com dispositivos móveis permitindo carga inicial de acordo com o banco de dados da entidade, onde

seriam atualizados durante a coleta de dados e posteriormente com gravação das informações no servidor, demonstrando todas as discrepâncias encontradas e permitindo que o usuário faça as devidas correções em interface amigável, contudo obedecendo limitações parametrizadas, visando evitar diferenças contábeis, como por exemplo: alteração de valor, apenas com processo de reavaliação ou depreciação, seguindo as normas pertinentes a este tipo de rotina, etc.

R3: ?

9. Poderia informar quais os campos que esse sistema possui para serem cadastrados no Inventário de Patrimônio, e qual a relevância dos mesmos (utilidade prática).

R1: *Data de inclusão, Centro de Custo, Descrição, Valor de aquisição*

R2: *Os principais campos são: Código, descrição resumida, descrição detalhada, tipo (móvel, imóvel, industrial), código de tombamento, grupos e sub-grupos, natureza, estado do bem, fornecedor, data de aquisição, data de tombamento, origem(dominiais, uso comum, especial), licitação, processo, empenho, valor de aquisição, valor atualizado, data de atualização, se for veículo, todas as informações pertinentes a este tipo de bem, tais como: placa, chassi, renavan, etc. Se tratar-se de bem imóvel da entidade deve possuir vínculo com a matrícula imobiliária. Enfim, estas são as informações básicas e de grande relevância para a correta identificação e mensuração do patrimônio, podendo ainda existir vários outros campos.*

R3: ?

10. Seu sistema possui alguma integração com aparelhos para auxiliar na coleta das informações (palmtop, smartphones ou tablets)? Caso possua, qual o aparelho?

R1: *Não possui.*

R2: *Não possui, conforme descrito na questão 8.*

R3: ?

APÊNDICE II – TELAS DO SISTEMA MOBILE EM FORMATOS RESPOSIVOS

Um sistema pensado para funcionar em diversos aparelhos móveis deve possuir a característica de acomodar a sua tela em tamanhos diferentes, dependendo da tela do aparelho que está sendo utilizado. Essa característica de se adaptar ao tamanho da tela e a perspectiva que o aparelho está, retrato ou paisagem, é chamada de tela responsiva.

A aplicação deste trabalho projetada para funcionar em dispositivos móveis utiliza uma tela com formato responsivo, e neste apêndice será adicionada as fotos das telas nos formatos mais utilizados nos aparelhos móveis, que são 480x320 para smartphones, 800x600 para tablets pequenos e 1024x768 para tablets grandes, tanto em formato retrato quanto paisagem.

Neste apêndice, é mostrado como uma das telas mais elaboradas do sistema se acomoda em formatos responsivos.

Lançamento de Inventário (Fotos) – 320x480



Figura 25: Lançamento de Inventário (Fotos) – 320x480

Lançamento de Inventário (Fotos) – 480x320



Figura 26: Lançamento de Inventário (Fotos) – 480x320

Lançamento de Inventário (Fotos) – 600x800

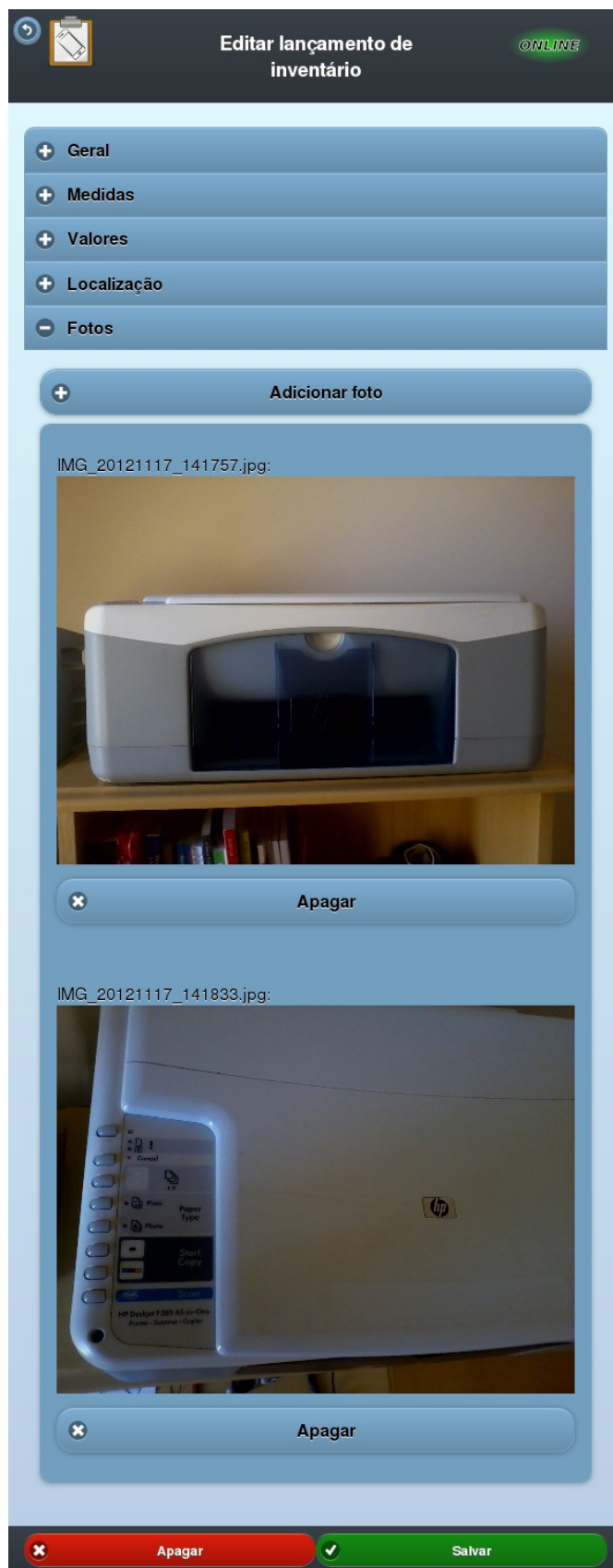


Figura 27: Lançamento de Inventário (Fotos) – 600x800

Lançamento de Inventário (Fotos) – 800x600

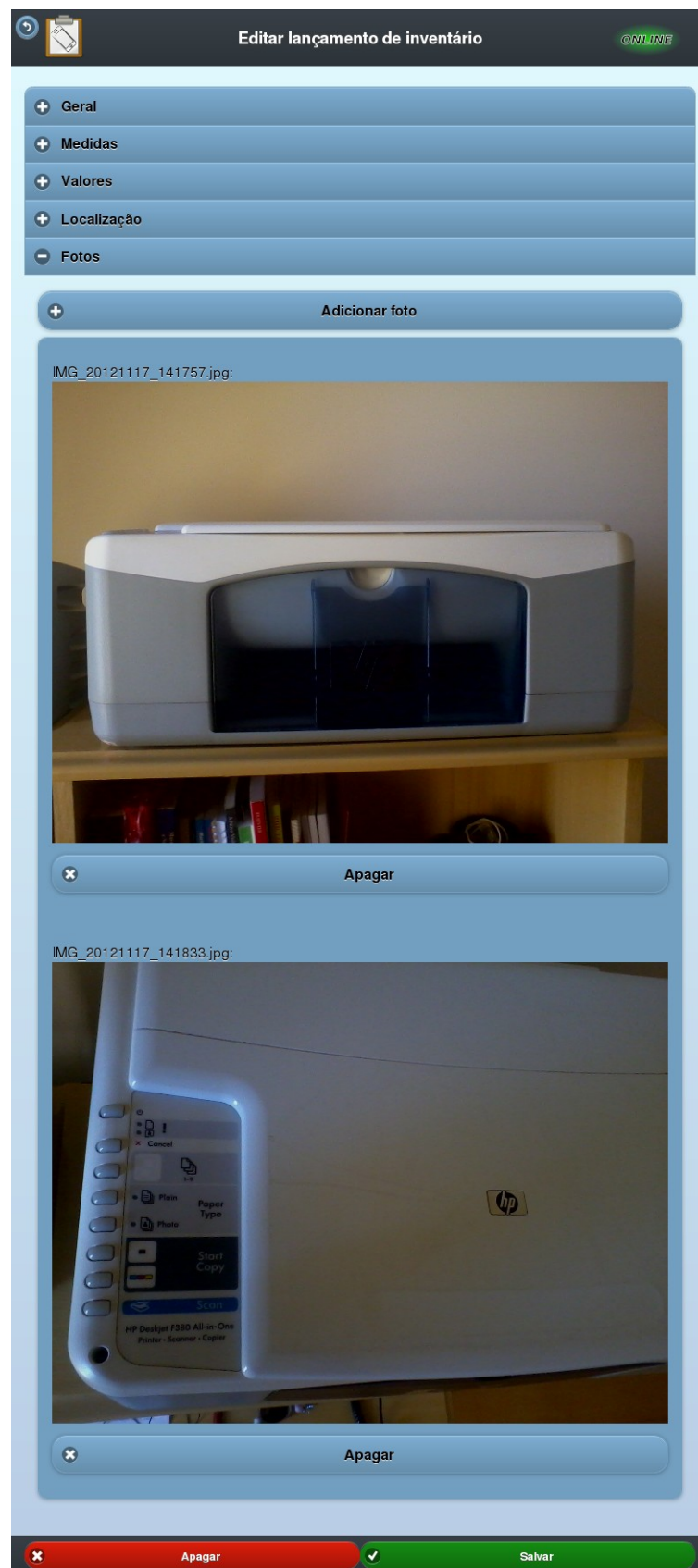


Figura 28: Lançamento de Inventário (Fotos) – 800x600

Lançamento de Inventário (Fotos) – 768x1024

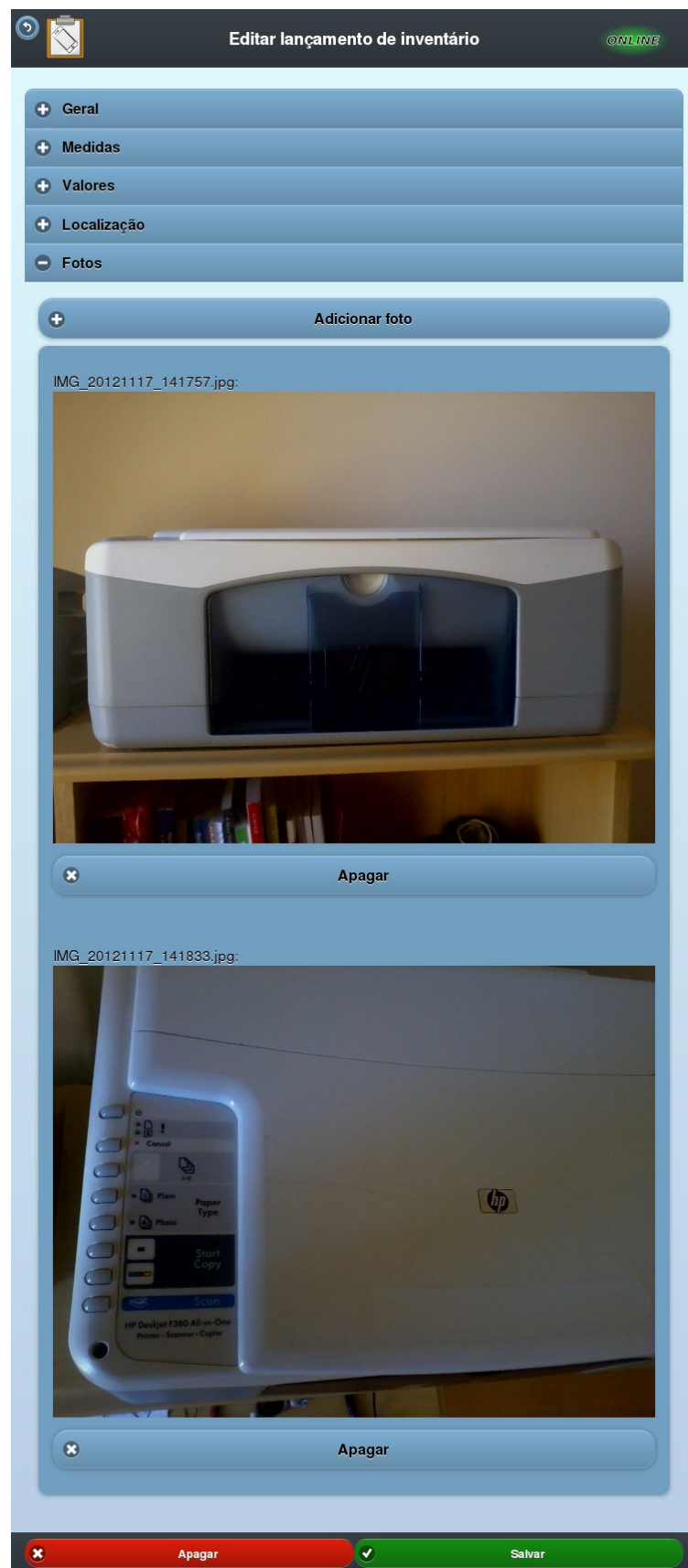


Figura 29: Lançamento de Inventário (Fotos) – 768x1024

Lançamento de Inventário – 1024x768



Figura 30: Lançamento de Inventário – 1024x768