

Software para gestão de resíduos sólidos da construção civil

Software for solid waste management from civil constructions

Gustavo Rodrigues de Oliveira Abreu

gustavor@gmail.com

Thiago Augusto Mendes

engenhaoaugusto@gmail.com

Resumo

Esse artigo tem como objetivo apresentar como é possível desenvolver uma ferramenta digital na área de gestão ambiental como um aplicativo web utilizando-se de conteúdos teóricos apresentados no curso de Engenharia Civil na área de algoritmos e gestão de resíduos sólidos. Foi criado um aplicativo web de fácil operação, dispensando qualquer processo de instalação, e que funcionam em diversos dispositivos, incluindo celulares e tablets. O software é capaz de facilitar o gerenciamento de resíduos sólidos da construção civil em canteiros de obras (RSCC) desde a geração até a destinação final abordando às exigências legais como a geração do documento de Controle de Transporte de Resíduos (CTR) conforme Norma Brasileira 15.112/2004 (ABNT, 2004) estabelece. Também foi desenvolvido para que, uma vez obtido através de seu endereço na internet, funcione sem mesmo estar conectado à rede mundial de computadores. Todos os processos realizados pelo aplicativo ocorrem localmente e, caso o aplicativo dependa de dados externos, estes são sincronizados ao servidor externo quando uma conexão for disponível. Isso permite a utilização desse aplicativo em áreas remotas sem requerer que o aplicativo tenha que ser instalado manualmente no dispositivo. Para que esse processo fosse simplificado foram utilizadas novas tecnologias para a plataforma web, como o cache off-line de aplicativos e o IndexedDB. A ferramenta digital desenvolvida atendeu satisfatoriamente o objetivo, pois, foi elaborada a partir de elementos teóricos desenvolvidos em sala de aula, testada e validada com mais de 1.500 entradas de dados hipotéticos sobre RSCC no que diz respeito às etapas de cadastro, envio, armazenamento e geração de gráficos e relatórios sobre os dados de entrada, gerando até patente registrada no Instituto Nacional de Propriedade Intelectual (INPI).

Palavras-Chave: *ferramenta digital; gestão; javascript; resíduos sólidos.*

Introdução

A Engenharia Civil possui diversas áreas de atuação onde a tecnologia da informação (TI) pode ser aplicada. Procurou-se neste artigo, trabalhar com as áreas da Engenharia Civil que tivessem ligação direta com o meio ambiente,

sustentabilidade e informática como as áreas da construção civil, resíduos sólidos e algoritmos.

Em consonância com as áreas citadas anteriormente têm-se a lei federal n.12.305/2010, que estabelece a Política Nacional de Resíduos Sólidos, a lei estadual n.8.544/1978 que dispõe sobre a Prevenção e o Controle da Poluição do Meio Ambiente e as Normas Brasileiras 10.004/2004 e 15.112/2004 que subsidiaram teoricamente sobre os elementos, definições, classes e tipos de resíduos sólidos, principalmente, os resíduos sólidos da construção civil (RSCC).

Desta forma, o artigo demonstra como foi possível o desenvolvimento de um *software* (aplicativo *web*) capaz de facilitar a gestão e o planejamento de atividades ligadas às áreas de geração e destinação dos RSCC produzidos em canteiro de obras. Para o desenvolvimento do aplicativo *web* foram utilizados a linguagem *JavaScript*, com parte do código sendo compilada com base na linguagem *LiveScript*, sendo também usadas bibliotecas dessa linguagem, como *jQuery*, *PouchDB* e *Morris*, e *Hypertext Markup Language (HTML)* e *Cascading Style Sheets (CSS)*, baseados na *framework Bootstrap*. A forma com que foram utilizadas estas ferramentas no desenvolvimento do aplicativo *web* está melhor descrita na metodologia deste artigo.

Com relação aos dados de entrada para o aplicativo *web*, foram levados em conta a possibilidade de caracterização, descrição e hierarquização dos locais onde os RSCC são gerados (canteiros de obras), inclusão sobre os tipos de acondicionamento, unidades e RSCC gerados, conforme estabelece a lei federal n.12.305/2010 (BRASIL, 2010) e normas brasileiras NBR 10.004/2004 e NBR 15.112/2004 (ABNT, 2004) e, sequência lógica de entrada de dados.

No *software* desenvolvido, há opções (funções de saídas) que permitem gerenciar todos os dados de entrada e produzir relatórios dos mais diferentes tipos e classificação. Buscou-se facilitar o uso do *software* em diferentes situações de uso. Entre as diversas condições pensadas têm-se a utilização do *software* em dispositivos móveis, situações onde não é possível o acesso imediato à *internet*, ou onde o acesso é limitado.

Objetivou-se desta forma, desenvolver uma ferramenta computacional que abrangesse a aplicação dos conteúdos de algumas áreas e disciplinas básicas e específicas da Engenharia Civil e outras engenharias (algoritmos, resíduos sólidos, meio ambiente e transporte e logística), além de atender e auxiliar a gestão da geração e destinação final dos RSCC produzidos em canteiros de obras preconizada pela lei federal n.12.305/2010.

As tendências da tecnologia da informação (TI) na engenharia civil

A tecnologia vem se mostrando forte em vários pontos da Engenharia Civil, principalmente nas áreas de cálculo e *computer-aided design* (CAD). Embora muito tenha se evoluído, como por exemplo, utilização de *softwares* com versões para *tablets* e celulares, nos quais proporcionam a leitura e edição rápida de projetos, algumas limitações, como não oferecer as mesmas funções em todas as plataformas e apresentar problemas de compatibilidade entre sistemas diferentes, ainda ocorrem.

Enquanto existem as tendências de se criar e usar aplicativos multiplataforma, representado principalmente pelas redes sociais, que se adaptaram para funcionar a praticamente qualquer dispositivo com conectividade, os *softwares* convencionais exigem que se tenha um computador em funcionamento, seja ele portátil ou não, instalar e abrir o aplicativo para finalmente poder operá-lo.

Softwares compatíveis com a computação móvel tornariam essas etapas mais dinâmicas, uma vez que esses dispositivos sempre estão ligados e o acesso aos *softwares* se torna muito mais rápido e prático. Além disso, eles tendem a ser muito simples e intuitivos

nesses dispositivos, já que o menor espaço da tela e a ausência de um teclado são compensados pela possibilidade de usar outros métodos de entrada que funcionam melhor nesses dispositivos.

Em pesquisa desenvolvida por Appert e Zhai (APPERT e ZHAI, 2009) o uso de gestos como atalhos para dispositivos com telas baseadas em toque (*touch screen*) foi investigado e como conclusão os autores mostram que esses atalhos por meio de gestos podem ser tão efetivos quanto com o uso de atalhos em um teclado, por exemplo (APPERT e ZHAI, 2009).

Tecnologias computacionais existentes e utilizadas

Existem algumas tecnologias para o desenvolvimento de *software*, podendo classificá-las por natureza, entre elas: aplicações *desktop* (o que inclui aplicativos *mobile*), *softwares* para servidores, *scripts*, aplicações *web*, entre outros. Desses, optou-se por utilizar três, com destaque: aplicações *desktop*, *softwares* para ser executado em servidores e aplicações *web off-line*, ou ricas.

Aplicações *desktop* são *softwares* convencionais onde o código é transferido para a máquina e executado na máquina somente após ações autorizadas pelo usuário, sendo limitado somente pelo sistema operacional. Passando por

menos limitações podem ter um maior controle de seus processos ao trabalhar diretamente, ou quase que diretamente com o código de máquina, porém isso leva a dificuldade de compatibilidade entre os diversos sistemas.

Aplicativos *mobile* se encaixam na descrição para *desktop*, pois tem limitações semelhantes, como a falta de compatibilidade entre sistemas e a necessidade do usuário ter que realizar diversas ações antes de utilizar o *software*, como: abrir loja de aplicativos, procurar o aplicativo, obtê-lo via *download* ou mídia física, instalá-lo e finalmente operá-lo. Numa situação ideal, os dois primeiros passos poderiam ser encurtados caso alguém já tivesse o endereço do aplicativo na loja, mas não os outros.

Essas limitações se devem ao maior nível de permissões que um aplicativo pode ter no sistema, como acesso ilimitado ao sistema de arquivos e *internet*. Dessa forma evita-se que o usuário possa correr riscos por abrir um programa e perder dados facilmente, o que ocorre comumente em vírus de computador. Porém nem todas as aplicações precisam de acesso às funções de baixo nível, um programa que irá realizar um cálculo simples, só precisa calcular, acessando arquivos em situações como carregar dados ou salvar resultados.

Uma solução para este tipo de situação

seria transferir todo o processamento para um computador dedicado, geralmente um servidor. O usuário só irá precisar conectar a esse servidor para usar o *software*, precisando somente de um *software* instalado no computador, chamado também de cliente. Esse é o exemplo de *software* para servidores, um modelo que é amplamente usado por *sites* e aplicações *web*. Além de requerer menos permissões por parte do sistema ele também tem a característica de manter o código no servidor, sem que possa ser acessado.

Porém, esse modelo não atende um dos objetivos desse trabalho, pois dependendo de uma conexão a outro computador não pode ser usado em situações onde a conexão não é possível, como em regiões remotas. Para resolver isso, deve-se passar ao menos parte da lógica do *software* para o cliente. Uma forma seria, sem exigir mais etapas na utilização, como um processo de instalação, utilizar as aplicações *web off-line*, ou ricas.

Aplicações *web off-line* ou ricas são aquelas que podem, mas não necessariamente, ser acessadas por meio de um endereço na *internet*. Assim como as aplicações *web* convencionais são *websites* e assim o único requerimento para esse tipo de aplicação é um navegador, algo que quase todos os sistemas operacionais já têm pré-instalados.

Além disso, ambas as aplicações *web* convencionais e *off-line* dispensam um processo de instalação, bastando abrir o endereço e o *software* já estará funcionando.

As diferenças entre as aplicações *web* convencionais e as *off-line* são os *off-line* se assemelham mais aos aplicativos *desktop*: enquanto aplicações convencionais se restringem em processar dados no servidor e envia-los prontos ao cliente, as aplicações *off-line* transferem o máximo possível do processamento para o cliente possibilitando que esse trabalhe independentemente do servidor.

Dessa forma, além do ganho inicial que ocorre ao requerer menos processamento do servidor, esses aplicativos ainda podem funcionar, em parte ou até mesmo completamente, sem conexão à *internet*. Dado o crescente número de novas tecnologias suportadas pelos navegadores, eles também tem acesso a diversas funções antes restritas somente as aplicações *desktop*, tais como: bancos de dados, *multithreading*, um sistema de arquivos próprio, acesso aos periféricos, como dados de geolocalização (via GPS ou Wi-Fi), microfone, câmera, entre outros.

Caso seja preciso usar mais funções do que está disponível, aplicativos *web* também podem ser facilmente transformados em aplicativos *desktop*. Além disso, o projeto *Open Web Apps* (aplica-

tivos *web* abertos), desenvolvido pela *Mozilla*, permitirá a instalação de aplicativos *web* de maneira bastante semelhante aos aplicativos convencionais, por exemplo, possibilitando que funcionem separados de um navegador.

Além disso, isso permite a uniformização da interface entre diversos dispositivos. Uma tendência também presente em outros tipos de *software*, como exemplo disso tem-se o *Windows 8* e seus sucessores, que são sistemas operacionais criados para atenderem dispositivos com configurações diversas, como *tablets* e *desktops*. Para conseguirem essa mudança fizeram alterações no *design*, simplificando ele em relação as versões anteriores, reduzindo efeitos gráficos e assim reduzindo o consumo de memória. Dessa forma foi possível que mais dispositivos pudessem executar o sistema, sem que houvesse perda de funcionalidade.

A criação de aplicativos *web* torna essa compatibilidade mais simples uma vez que precisam de poucas adaptações entre plataformas. Uma forma de se obter isso é pela abordagem *progressive enhancement*:

You start by establishing a basic level of user experience that all browsers will be able to provide when rendering your web site, but you also build in more advanced functiona-

lity that will automatically be available to browsers that can use it. (WORLD WIDE WEB CONSORTIUM, 2012)¹

Metodologia

Além de ampla pesquisa bibliográfica realizada sobre a legislação federal no que tange a gestão dos resíduos sólidos da construção civil (RSCC) conforme preconiza BRASIL (2010), CONAMA (2002) e ABNT (2004), a metodologia empregada no desenvolvimento do *software* apresentado neste trabalho foi composta das seguintes etapas: escolha das tecnologias da informação e linguagens de programação aplicáveis, desenvolvimento do sistema e verificação e validação (funcionamento) do *software* sob diferentes configurações (cenários) de entrada.

Escolhas para o desenvolvimento do software – as ideias

Após analisar as várias atividades e experimentos que poderiam envolver as disciplinas básicas da Engenharia Civil no desenvolvimento de um *software*, optou-se por desenvolver um aplicativo *web* gerenciador de resíduos sólidos

gerados em canteiro de obras (RSCC). Para o desenvolvimento do *software*, inicialmente pensou-se que o usuário pudesse entrar com os dados de entrada básicos necessários para a gestão (exemplo: dados da obra, responsáveis, quantidade, unidade e tipos de RSCC), para então o *software* processá-los e classificá-los segundo resolução CONAMA n.307/2002 estabelece, para posteriormente, elaborar gráficos, tabelas, diagramas e outros tipos de resultados (dados de saída).

No *software* pode-se entrar com os dados físicos dos diversos RSCC gerados em uma ou mais obras e/ou empreendimentos e obter informações e resultados para gestão baseado em cálculos internos do *software* (processamentos) realizados com os dados de entrada. Os formulários usam atributos que permitem uma utilização mais fácil em dispositivos onde o teclado é virtual, por exemplo, mostrando um teclado numérico em determinados campos.

Sendo a navegação simples, baseado nos modelos do *Bootstrap 2.3.2*, ela funciona bem em telas pequenas, sem desperdício de espaço e adaptando-se bem a telas maiores e possibilitando o uso do

¹ Tradução: “Você começa estabelecendo um nível básico de experiência do usuário que todos os navegadores serão capazes de fornecer ao representar o seu site, mas você também constrói funcionalidades mais avançadas, que estarão automaticamente disponíveis para os navegadores que puderem usá-las.”

aplicativo com diversos meios de entrada, como teclado, *mouse* e *touch screen*.

Tecnologia da Informação (TI) envolvida e aplicações

O *software* foi desenvolvido para ser usado independente de situação ou localidade, isto é, podendo ser usado em escritórios, canteiros de obras, regiões remotas, mesmo sem acesso à internet. Ainda pode-se usá-lo em dispositivos mais simples, como celulares e *tablets*, dispensando o uso de dispositivos como computadores de mesa e *notebooks*. Também se possibilitou que o aplicativo seja iniciado mais rápido quando em dispositivos portáteis, uma vez já que esses geralmente permanecerem sempre ligados dispensando-se ter que iniciar o sistema para então usar o aplicativo.

Quando há dados para serem armazenados, como formulários e relatórios, o *software* conta com um pequeno banco de dados local que é sincronizado com o servidor central assim que é possível realizar uma conexão entre ambos, via *internet* e/ou *intranet*. Assim é possível manter um sistema centralizado, permitindo o acesso e controle dos dados, porém também prevenindo possíveis falhas que podem acontecer devido à falta de conexão em um canteiro de obra, mantendo um espelho do banco de dados principal localmente.

Tanto o banco de dados central baseado no *CouchDB 1.6.1*, que é implementado em *Erlang*, quanto o banco de dados local baseado no *PouchDB*, que é uma implementação em *JavaScript* bem similar e compatível com o *CouchDB*, foram criados para permitir essa integração. Ou seja, eles contam com ferramentas que tornam possível o uso dos aplicativos baseados neles sem haver a necessidade de uma conexão constante com o servidor central. Também ao sincronizar esses clientes entre si e com o banco central de dados há um risco quase nulo de que os dados sejam sobrescritos ou perdidos no processo. Primeiro, evitando que duas entradas do banco tenham o mesmo identificador e segundo, resolvendo possíveis conflitos através de processo já presente nesses bancos.

O modelo no qual esses bancos de dados são baseados é chamado de *Multi-Version Concurrency Control* (Controle de Concorrência em Multi-Versão – MVCC), que é descrito na documentação do *CouchDB* da seguinte forma:

MVCC means that CouchDB can run at full speed, all the time, even under high load. Documents in CouchDB are versioned, much like they would be in a regular version control system such as Subversion. If you want to change a value in a document, you create an entire new version of that document and save it over the old one. After doing this, you end

up with two versions of the same document, one old and one new. (APACHE SOFTWARE FOUNDATION, 2013)²

Assim dados criados localmente, como novos formulários, não irão entrar em conflito com dados já presentes no servidor principal, mesmo quando não é possível uma conexão com ele e conflitos gerados por alterações são resolvidos sem causar perdas. Ainda alterações realizadas no banco de dados central refletirão nos espelhos sem dificuldades ou conflitos e com baixo consumo de banda de dados através de um processo de replicação incremental.

O *software* desenvolvido foi nomeado pelos autores de W3RESÍDUOS e registrado no Instituto Nacional de Propriedade Intelectual (INPI) sob o registro de número BR512014001536-0 na data de 17/12/2014.

Desenvolvimento do *software*

Escolheu-se criar um aplicativo *web* com foco em *desktops* e dispositivos móveis. Usou-se as funções de armazenamento *off-line*, através do mecanismo *Application Storage*. Também se fez o uso do banco de dados *IndexedDB* através da biblioteca *PouchDB*, o que ofereceu ferramentas para que esse

banco de dados fosse sincronizado com um banco de dados central e aumentou a compatibilidade com navegadores antigos. Também foram utilizadas bibliotecas que permitiram acessar a modularização do tipo *Asynchronous Module Definition* (Definição de Módulo Assíncrona – AMD), sincronização de dados com o servidor e criação de gráficos.

Por conta da necessidade de criar um código compatível com qualquer sistema, utilizou-se também sistemas que normatizaram as diferenças entre diferentes versões deles, como o *normalize.css*, que resolve inconsistências nos estilos padrões que há entre diferentes navegadores.

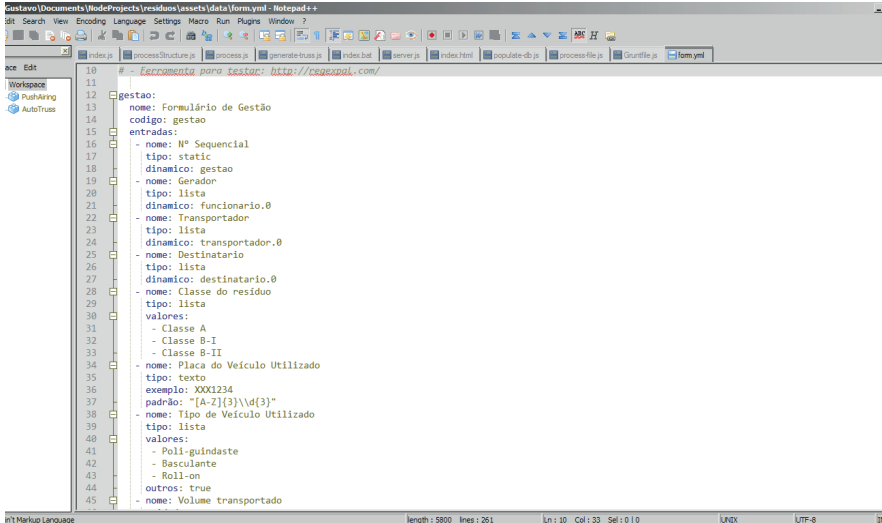
Para fornecer uma base para o desenvolvimento do *software* foi utilizada a ferramenta *Yeoman*, que gerou um modelo básico contendo as ferramentas básicas para a criação de *websites* e aplicativos *web*. Como parte desse modelo foram inclusas a *framework Bootstrap 2.3.2*, para o estilo e organização padrão dos componentes usados, e a biblioteca *jQuery 1.9.1* para manipulação dos elementos da página. Também foi utilizada a biblioteca *Morris 0.4.3* para a geração de gráficos.

² Tradução: “MVCC significa que CouchDB pode ser executado a toda a velocidade, o tempo todo, mesmo sob carga elevada. Documentos em CouchDB são versionados, bem como seriam em um sistema de controle de versão regular, como Subversion. Se você quiser alterar o valor em um documento, você cria toda uma nova versão desse documento e salva-o sobre o antigo. Após fazer isso, você termina com duas versões de um mesmo documento, um velho e um novo.”

O *software* é iniciado através da abertura de páginas em marcação *Hypertext Markup Language* (HTML) os quais definem a estrutura do aplicativo e cria a união entre o conteúdo, a apresentação e a interação.

Algumas dessas páginas foram geradas com o gerador *assemble* com base em uma configuração escrita em YAML, tornando possível uma configuração fácil do programa, como pode ser visualizada na Figura 1.

Figura 1
Configuração do aplicativo, com a descrição de todos os dados de entrada do aplicativo, assim como a configuração da validação desses dados, escrita em YAML.



```
10 # Ferramenta para testar: http://rampopal.com/
11
12 gestao:
13   nome: Formulário de Gestão
14   codigo: gestao
15   entradas:
16     - nome: Nº Sequencial
17       tipo: static
18       dinamico: gestao
19     - nome: Gerador
20       tipo: lista
21       dinamico: funcionario.0
22     - nome: Transportador
23       tipo: lista
24       dinamico: transportador.0
25     - nome: Destinatario
26       tipo: lista
27       dinamico: destinatario.0
28     - nome: Classe do resíduo
29       tipo: lista
30       valores:
31         - Classe A
32         - Classe B-I
33         - Classe B-II
34     - nome: Placa do Veículo Utilizado
35       tipo: texto
36       exemplo: XXX1234
37       padrão: "[A-Z]{3}\\d{3}"
38     - nome: Tipo de Veículo Utilizado
39       tipo: lista
40       valores:
41         - Polí-guindaste
42         - Basculante
43         - Roll-on
44     outros: true
45     - nome: Volume transportado
```

Esses três contextos são organizados em tipos de arquivos diferentes por permitirem o reaproveitamento de conteúdo, conforme recomendação da W3C (*World Wide Web Consortium*):

Experience shows that the separation of content, presentation, and interaction promotes the reuse and device-independence

of content. [...] designing content that that can be recombined on the client also tends to make that content applicable to a wider range of devices. This design also improves caching efficiency and offers users more presentation options.³

Os arquivos HTML, como dito anteriormente, definem a estrutura básica do aplicativo, assim cabe a eles organização

3 Tradução: “A experiência mostra que a separação do conteúdo, apresentação e interação promove a reutilização e independência do conteúdo em relação ao dispositivo. [...] concebendo conteúdos que podem ser recombinados no cliente também tendem a fazer com que o conteúdo seja aplicável a uma vasta gama de dispositivos. Este projeto também melhora a eficiência do cache e oferece aos usuários mais opções de apresentação.”

dos elementos que compõem o aplicativo, como formulários, botões, campos de texto, mensagens de texto, etc. Os arquivos de estilo definem a aparência dos elementos do aplicativo e são codificados usando CSS (*Cascading Style Sheets*). Os *scripts* definem a lógica do aplicativo e são codificados na linguagem *JavaScript*. Esses *scripts* podem ser escritos em *JavaScript* ou escritos em outra, como *CoffeeScript*, *TypeScript*, *Elm*, *Haxe*, entre

outras, e depois convertida para *JavaScript*. O *software* desenvolvido utilizou a linguagem *LiveScript* em seu desenvolvimento inicial, por ela ter uma sintaxe simples e baseada na *indentação*. Assim a estrutura do código ficou mais fácil de ser compreendida, mesmo para quem não têm conhecimento da linguagem de programação usada. A Figura 2 mostra um exemplo de estruturação de código fonte utilizado.

```
/*
 * ## carregar-database
 * Abstraí a API do PouchDB em forma de promises jQuery.
 */
carregar-database = (nome, opcoes = {}) ->
  # Cria um Deferred (base para as promises jQuery):
  deferred = $.Deferred!
  # Cria um objeto representando o banco de dados:
  new PouchDB nome, opcoes, (err, result) !->
    # Caso ocorra um erro (servidor offline, por exemplo):
    if err
      # Mostra o erro no console:
      console.error err
      # Resolve a promise com um valor nulo
      # (entendido como um erro nesse aplicativo)
      deferred.resolve null
    else
      # Resolve a promise com o resultado
      deferred.resolve result
  # Retorna a promise jQuery:
  deferred.promise!
```

Figura 2

Exemplo de código fonte (forma de estruturação da linguagem de programação utilizada para elaboração do *software* para gestão de resíduos sólidos da construção civil – RSCC)

Por essa linguagem (*LiveScript*) gerar um código limpo e de fácil compreensão e para ter mais controle do código executado, as mudanças e otimizações ocorridas no final do desenvolvimento do programa foram realizadas diretamente no código *JavaScript*.

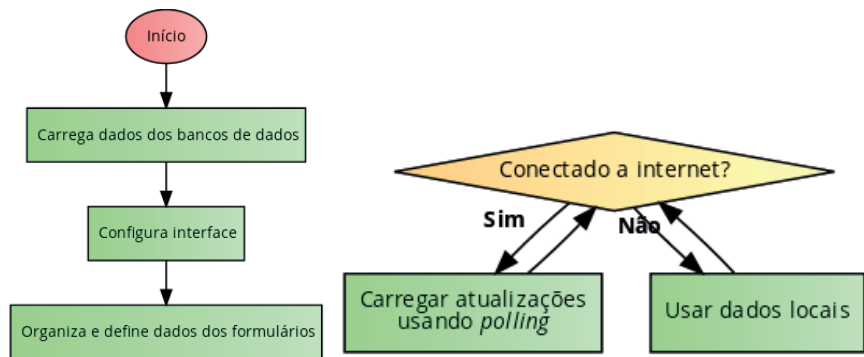
Depois de carregados todos os arquivos da página, inicia-se a parte lógica, sendo executadas funções que possibilitam que o programa seja executado, como atribuir funções a botões e *links* da página e carregar dados a partir de um banco de dados externo.

Nessas funções também são atribuídas o funcionamento do aplicativo, como cálculos, entrada e armazenamento de dados, processos diversos, entre outros.

Todos os arquivos do aplicativo, listados em um arquivo de manifesto são pré-carregados e armazenados na primeira utilização. A partir das utilizações seguintes os arquivos são carregados diretamente da memória tornando

o carregamento muito mais rápido do que anteriormente. Isso também permite que o aplicativo possa funcionar sem uma conexão à rede. Atualizações do aplicativo ocorrem quando é verificada alterações no arquivo de manifesto, que é atualizado a cada utilização do aplicativo. Esse funcionamento pode ser representado conforme diagramas e fluxogramas da Figura 3.

Figura 3
Diagrama da inicialização do *software*, assim como o processo de atualização de dados.



Dados de entrada do software

O *software* usa dois tipos básicos de dados: campos e listas. Os campos se dividem em campos de texto, numéricos e texto padronizado. As listas mostram opções a partir de dados salvos no banco de dados.

Os campos numéricos podem ser acompanhados de um campo de lista onde é especificada a unidade utilizada no campo, como m³ ou toneladas. Os campos de texto padronizado são usados para casos como CEP, CNPJ, telefone, entre outros. As listas podem com-

plementadas com um campo de texto caso o usuário escolha uma opção que não seja presente na lista.

Os dados de entrada do *software* podem ser adicionados nos *menus* cadastro (empresa, empreendimento, funcionário responsável, transportador e destinatário) e gestão (tipos de resíduos, unidade, quantidade).

Processamento

Para realizar os cadastros primeiramente devem-se preencher os formulários.

Os dados passam por uma validação simples na qual uma vez não atendendo os requisitos de formato (programados) o usuário será alertado do erro e deverá corrigir os campos.

Depois de feitos os cadastros necessários, a página de gestão ficará ativa. Nela o usuário pode preencher os dados para a geração do CTR. Após isso ele será redirecionado a página onde ele poderá imprimir o CTR gerado.

Assim que o *software* passa a ter acesso a dados suficientes e validados é possível acessar a página de gráficos, na área de gestão, podendo correlatar a quantidade de resíduos gerados em relação ao tempo, classe de resíduos gerados por empreendimento ou empresa, com a possibilidade de filtrar dados por período, empresas e classes de resíduos escolhidos.

Dados de Saída

O programa permite a geração de formulários de Controle de Transporte de Resíduos (CTR) e gráficos mostrando o volume transportado por dia em um determinado intervalo de tempo com a possibilidade de filtrar os dados conforme a classe dos resíduos, transportadores, geradores e os destinatários. A exportação dos gráficos, tabelas e dados de saídas e também a impressão são possíveis a qualquer momento.

Funcionamento do *software* (verificação e validação)

Para avaliar o funcionamento do aplicativo *web* (verificação e validação) foi testado o preenchimento dos formulários de entrada, inclusive o preenchimento automático, inúmeras vezes (cerca de 1.500 entradas) com dados hipotéticos, e então, verificado os resultados (dados de saída) se estavam corretos de acordo com os dados entrados (validação) e se não ocorreram problemas durante a utilização, filtragem, geração e impressão dos dados de saída (verificação). Todos os testes (cerca de 100 vezes) ocorreram nas seguintes situações: *off-line*, conectado a um servidor na mesma rede e conectado a um servidor na internet; em um computador *desktop* e em um celular; nos navegadores *Internet Explorer*, *Google Chrome* e *Mozilla Firefox* e demandou um tempo total de simulação de aproximadamente 5 horas (cerca de um mês de testes).

A utilização prática (*in loco*) do aplicativo em construtoras, empreiteiras ou empresas, é uma etapa a ser realizada. O aplicativo web W3Resíduos foi apresentado ao órgão municipal do meio ambiente da cidade de Goiânia (AMMA) que demonstrou interesse no aprimoramento da ferramenta, embora até o momento da escrita desse

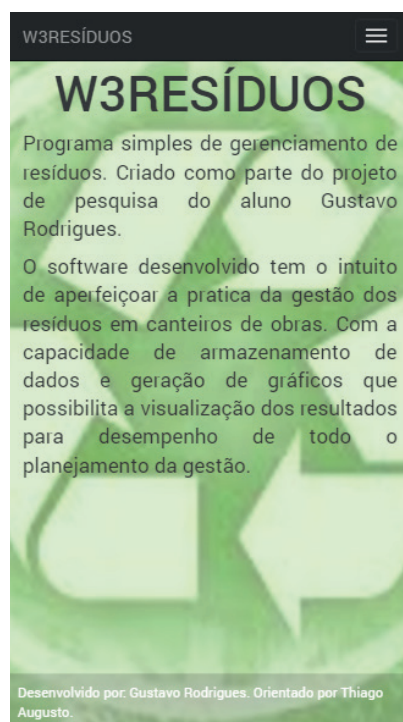
artigo não haviam sido concretizadas parcerias com essa finalidade.

Resultados e discussões

O aplicativo *web* (*software*) foi desenvolvido com intuito de conectar os diversos elementos teóricos apresentados na disciplina de algoritmos com temas reais da Engenharia Civil, o que aconteceu para a área de gerenciamento de resíduos sólidos na construção civil (RSCC). O *software* desenvolvido

conta com quatro áreas: início, onde são mostradas informações sobre o *software* (Figura 4); cadastros, onde são cadastradas as informações de entrada como: empresas, empreendimentos, responsáveis técnicos, fornecedores, transportadoras e destinadores finais de resíduos (Figura 5); gestão, onde são preenchidos os dados para a geração do documento legal de Controle de Transporte de Resíduos (CTR) conforme Figura 6 e, relatórios e gráficos (Figura 7).

Figura 4
Telas de início e de formulário de gestão em modo *mobile*



O *menu* “Gestão” (Figura 5) possibilita a entrada de dados referente aos

tipos, quantidades, formas de acondicionamento (peso, volume ou unidades)

de RSCC gerados em cada canteiro de obra e empreendimento cadastrado anteriormente no *menu* “Cadastro”. Além disso, neste mesmo *menu* (ícone azul da Figura 5) é possível gerar o documento legal exigido (CTR) pela Norma Brasileira (NBR) 15.112/2004 (ABNT, 2004).

Os dados de entrada que se refere ao *menu* “Cadastro” (Figura 5) possibilitam a organização, hierarquia e armazenamento correto das obras e empreendimentos a serem cadastrados e armazenados no banco de dados. Tanto o preenchimento, inclusive o preenchimento automático (com relação aos formatos de CEP, CPF, CNPJ e datas), quanto o armazenamento e

hierarquização foram testados com dados hipotéticos (cerca de 150 entradas) das mais diversas possibilidades (número e tipos de caracteres, ações inesperadas dos usuários, falta de internet na ação de cadastramento e outros) no intuito de testar, verificar e validar o funcionamento da ferramenta computacional.

O *menu* “CTR” (Figura 6) possibilita monitorar todo o banco de dados criado por item com possibilidade de impressão do documento legal. Na Figura 6 percebe-se a quantidade de dados salvos no *software* comprovando que as etapas de verificação e validação do aplicativo foram realizadas em número satisfatório.

Figura 5
Tela do formulário de gestão de entrada para cadastros)

Ainda na Figura 6, destaca-se o local onde pode ser realizada a ação de im-

pressão e gerenciamento dos dados cadastrados para cada CTR a ser impressa.

Figura 6
Tela da lista de formulários cadastrados (Menu “CTR”)

Destinação	Gerador/Origem	Transportadora	Ações
45008404/Destinatário: Nome / Razão Social 1254	55675657/Empreendimento: Nome 9373	0962933/Transportador: Nome / Razão Social 6536	
51288032/Destinatário: Nome / Razão Social 1254	07067554/Empreendimento: Nome 9373	44416478/Transportador: Nome / Razão Social 6536	
42817140/Destinatário: Nome / Razão Social 1254	14739038/Empreendimento: Nome 9373	72424734/Transportador: Nome / Razão Social 6536	
67731000/Destinatário: Nome / Razão Social 1254	05178842/Empreendimento: Nome 9373	11684493/Transportador: Nome / Razão Social 6536	
96418058/Destinatário: Nome / Razão Social 1254	76154985/Empreendimento: Nome 9373	15521058/Transportador: Nome / Razão Social 6536	
82449765/Destinatário: Nome / Razão Social 1254	27228946/Empreendimento: Nome 9373	34956182/Transportador: Nome / Razão Social 6536	
49004410/Destinatário: Nome / Razão Social 1254	72003298/Empreendimento: Nome 9373	20874429/Transportador: Nome / Razão Social 6536	
76900139/Destinatário: Nome / Razão Social 1254	12408347/Empreendimento: Nome 9373	14285570/Transportador: Nome / Razão Social 6536	
82858377/Destinatário: Nome / Razão Social 1254	35416718/Empreendimento: Nome 9373	18555971/Transportador: Nome / Razão Social 6536	
96935378/Destinatário: Nome / Razão Social 1254	93152748/Empreendimento: Nome 9373	27652316/Transportador: Nome / Razão Social 6536	
78750539/Destinatário: Nome / Razão Social 1254	51120639/Empreendimento: Nome 9373	63051120/Transportador: Nome / Razão Social 6536	
52531306/Destinatário: Nome / Razão Social 1254	99934005/Empreendimento: Nome 9373	24443868/Transportador: Nome / Razão Social 6536	
82905536/Destinatário: Nome / Razão Social 1254	65465929/Empreendimento: Nome 9373	74809097/Transportador: Nome / Razão Social 6536	
65736418/Destinatário: Nome / Razão Social 1254	48889793/Empreendimento: Nome 9373	21866961/Transportador: Nome / Razão Social 6536	
89497987/Destinatário: Nome / Razão Social 1254	59431113/Empreendimento: Nome 9373	45033014/Transportador: Nome / Razão Social 6536	
88468428/Destinatário: Nome / Razão Social 1254	96345621/Empreendimento: Nome 9373	97584344/Transportador: Nome / Razão Social 6536	
36772999/Destinatário: Nome / Razão Social 1254	15524502/Empreendimento: Nome 9373	16912493/Transportador: Nome / Razão Social 6536	

A opção “Gráficos” permite aos usuários produzir, salvar e exportar diversos gráficos e relatórios por meio de filtros que

relacionam as diversas possibilidades de interesse e demanda para cada obra e/ou empreendimento cadastrado (Figura 7).

Figura 7
Tela apresentando gráfico de gestão



O correto funcionamento do *software*, principalmente no que concerne ao armazenamento dos dados quando não há internet e ao envio dos dados e

armazenamento quando há internet, só foi possível devido a fase de testes, verificação e validação do *software*, onde foram identificados erros na geração

dos gráficos e relatórios, quantificação detalhada dos diferentes tipos de RSCC gerados e *layouts*, os quais foram posteriormente solucionados.

Conclusões

Após o planejamento e implementação do aplicativo *web* conclui-se que é possível relacionar a teoria com a prática, principalmente envolvendo as áreas existentes na Engenharia Civil. Também há muitas outras aplicações que ainda podem ser desenvolvidas e melhoradas no que tange a tecnologia da informação (TI) aplicada às engenharias, abrangendo uma maior gama de dispositivos e situações que estão surgindo recentemente.

Procurou-se trabalhar com tecnologias atuais que permitem que o aplicativo idealizado (gestão de resíduos sólidos da construção civil – RSCC) pudesse ser desenvolvido de forma alternativa aos aplicativos convencionais. Assim, o aplicativo W3RESÍDUOS desenvolvido funciona também em situações como: dispositivos fixos e móveis, indisponibilidade de uma conexão à internet, além de possibilitar a compatibilidade entre plataformas.

Também, com o auxílio de bibliotecas e *frameworks*, foi possível que o aplicativo pudesse funcionar em diversas situações, ou seja, diferentes tamanhos de tela e meios de entrada

e equipamentos. Dessa forma uniformizou-se o formato de utilização do aplicativo nessas situações o que facilita a utilização e a familiarização com o modo de operá-lo.

Desta forma, foi possível aplicar bem os conceitos teóricos desenvolvidos em sala de aula referente às disciplinas de algoritmos (linguagem de computadores), logística e meio ambiente do curso de Engenharia Civil no desenvolvimento real de um aplicativo *web* para auxiliar a gestão da geração e destinação final de RSCC.

Na fase de testes, simulações e verificações do *software* detectou-se a necessidade de melhoria da usabilidade do aplicativo simplificando os processos de instalação ao exigir somente um navegador e que um *website* seja aberto, sendo então todos os recursos necessários pelo aplicativo carregados pelo navegador e permitindo que ele funcione localmente, de forma semelhante a um aplicativo que passa por um processo de instalação. Erros no cadastramento, quantificação de cada tipo e classe de resíduos, geração de gráficos e relatórios e *layouts* para cada tipo de equipamento de acesso também foram identificados e sanados nesta etapa.

Sendo assim, o objetivo proposto foi alcançado, mesmo que todo o funcionamento dos sistemas envolvidos

no processamento do *software* utilizaram-se de dados hipotéticos para validação de finalidade, funcionamento e geração das saídas.

As tecnologias utilizadas no desenvolvimento do *software* podem ser aplicadas para outras áreas do conhecimento. Embora sejam tecnologias relativamente novas e algumas ainda sen-

do discutidas e em desenvolvimento foi visto que elas podem ser muito úteis e já podem ser aplicadas e melhoradas.

Pretende-se futuramente implementar em ambiente de teste o *software* W3RESÍDUOS em empresas de construção civil no município de Goiânia/GO para trabalhar em escala de mercado (dados reais em situações locais).

Referências

- APACHE SOFTWARE FOUNDATION. *Apache CouchDB 1.5 Documentation*. [S.l.]: Apache Software Foundation, 2013. Disponível em: <<http://docs.couchdb.org/en/latest/index.html>>. Acesso em: 19 jun 2014.
- APPERT, C.; ZHAI, S. Using Strokes as Command Shortcuts: Cognitive Benefits and Toolkit Support. In: CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 1., 2009, Boston/MA. *Proceedings...* Boston/MA: Interaction Design Foundation 2009, p. 2289–2298.
- ABNT (Associação Brasileira de Normas Técnicas). *NBR 10.004: resíduos sólidos – classificação*. 2. ed. Rio de Janeiro, 2004.
- _____. *NBR 15.112: resíduos da construção civil e resíduos volumosos – áreas de transbordo e triagem – diretrizes para projeto, implantação e operação*. Rio de Janeiro, 2004.
- BRASIL. Lei federal n.12.305, de 2 de agosto de 2010. Institui a Política Nacional de Resíduos Sólidos; altera a Lei n.9.605, de 12 de fevereiro de 1998, e dá outras providências. *Diário Oficial da União*, Brasília, DF, 2 ago. 2010.
- BRASIL. Conselho Nacional do Meio Ambiente (Conama). Resolução n.307, de 5 de julho de 2002. Estabelece diretrizes, critérios e procedimentos para a gestão dos resíduos da construção civil. *Diário Oficial da União*, Brasília, DF, 17 set. 2002.
- WORLD WIDE WEB CONSORTIUM. *Graceful degradation versus progressive enhancement*. 2012. Disponível em: <http://www.w3.org/wiki/Graceful_degradation_versus_progressive_enhancement>. Acesso em: 22 jun. 2014.
- _____. *Architecture of the World Wide Web, Volume One*. 2014. Disponível em: <<http://www.w3.org/TR/webarch/#pci>>. Acesso em: 21 jun. 2014.
- _____. *Indexed Database API*. 2014. Disponível em: <<https://dvcs.w3.org/hg/IndexedDB/raw-file/tip/Overview.html>>. Acesso em: 21 jun. 2014.

Abstract

This article aims to show how is possible developing a digital tool in the environmental management area as a web application using the theoretical concepts presented in the Civil Engineering course in the area of algorithms and solid waste management. A web application was created for easy operation, eliminating any installation processes, and to work on multiple devices, including phones and tablets. The software is able to improve the management of solid waste from construction sites (RSCC) from generation until to final disposal, addressing the legal requirements as the generation of the Controle de Transporte de Resíduos (CTR – Waste Transport Control) document as established by the Norma Brasileira 15.112/2004 (ABNT, 2004). It was also designed for, once obtained through his internet address, work without being connected to the World Wide Web. All processes performed by the application occur locally and, if the application relies on external data, these are synchronized to the external server when a connection is available. This allows the use of this application in remote areas without requiring a manual installation of the application on the device. In order to simplify the process were used technologies for the web platform like the off-line application cache and IndexedDB. This digital tool satisfactorily met the goal because, it was created from theoretic elements developed in classroom, tested and validated with more than 1.500 hypothetical entries about RSCC in the registration, sending, storage and geration of graphs and reports about the inputted data, even generating a patent registered in the Instituto Nacional de Propriedade Intelectual (INPI).

Keywords: *digital tool, management, javascript, solid waste.*