

**Eco-on**  
**Um Sistema de Denúncia de Descarte Irregular de Resíduos Sólidos**

**Eco-on**  
**A System for Reporting Irregular Solid Waste Disposal**

DOI:10.34117/bjdv7n4-403

Recebimento dos originais: 07/03/2021

Aceitação para publicação: 15/04/2021

**Celso Maciel da Costa**

Doutorado em Computação

Universidade Estadual do Rio Grande do Sul (Uergs)

Endereço: Rua Santa Maria, 2300 - Jardim dos Lagos, 92500-000 - Guaíba - RS, Brasil

E-mail: celso-costa@uergs.edu.br

**Daniela Mueller de Lara**

Doutorado em Ambiente e Desenvolvimento

UERGS - Universidade Estadual do Rio Grande do Sul, Soledade, RS, Brasil

E-mail: daniela-lara@uergs.edu.br

**Guilherme Machado Lague**

Graduando em Engenharia de Computação

Universidade Estadual do Rio Grande do Sul (Uergs)

Endereço: Rua Santa Maria, 2300 - Jardim dos Lagos, 92500-000 - Guaíba - RS, Brasil

E-mail: guilherme-lague@uergs.edu.br

**Rafael Araldi**

Bacharel em Gestão Ambiental.

UERGS - Universidade Estadual do Rio Grande do Sul

Soledade, RS, Brasil

E-mail: rafael-araldi@uergs.edu.br

**RESUMO**

Cidades inteligentes utilizam recursos tecnológicos na construção de soluções para grandes problemas, tendo em vista a melhoria da qualidade de vida da população. Um desses grandes problemas é a quantidade de resíduos produzidos nas cidades e a existência de descartes irregulares, realizados em locais impróprios. Esse estudo apresenta um sistema que tem por objetivo permitir que os cidadãos façam denúncias da existência de descartes de resíduos sólidos em locais impróprios. É formado por um aplicativo, que roda em smartphones, e um site, que permite ao gestor público acessar o sistema. Com o uso do smartphone os cidadãos podem fazer denúncias, informando o local e anexando fotos dos resíduos descartados irregularmente. O site permite ao gestor público acessar as denúncias e tomar providências para a solução do problema. O usuário pode acompanhar o andamento de suas denúncias e também visualizar todas as denúncias existentes no município.

**Palavras-chaves:** cidades inteligentes, gestão de resíduos sólidos, aplicativos móveis.

## ABSTRACT

Smart cities use technological resources to build solutions to major problems, with a view to improving the quality of life of the population. One of these major problems is the amount of waste produced in cities and the existence of irregular discharges, carried out in inappropriate places. This study presents a system that aims to allow citizens to make complaints about the existence of solid waste discharges in inappropriate places. It consists of an application, which runs on smartphones, and a website, which allows the public manager to access the system. With the use of smartphones, citizens can make complaints, informing the location and attaching photos of the irregularly discarded waste. The website allows the public manager to access the complaints and take steps to solve the problem. The user can follow the progress of their complaints and also view all the complaints existing in the municipality.

**Keywords:** smart cities, solid waste management, mobile apps.

## 1 INTRODUÇÃO

Cidades inteligentes são aquelas que fazem uso de recursos tecnológicos no planejamento de ações para a solução de problemas relacionados ao meio ambiente, a educação, ao

transporte, aos recursos hídricos, etc. [1], [2]. Um problema também atual é que em muitas cidades o tipo de vida dos cidadãos tem levado a uma falta de preocupação com os recursos naturais e com as questões relativas ao meio ambiente.

Cidades sustentáveis visam à criação de um sistema econômico e social que possibilite aumento de renda, melhorias na saúde, na educação. Necessitam oferecer acesso à segurança, energia, alimentos e tratar com as questões climáticas [3], [4], [5]. Também é de fundamental importância que sejam desenvolvidas ações que visem promover a participação dos cidadãos, na elaboração de estratégias de desenvolvimento econômico e social, na busca por melhorias na qualidade de vida.

A internet das coisas (IoT), vem mudando a forma de interação com o mundo. Existem cada vez mais objetos físicos (celulares, computadores, relógios, veículos, prédios, ...) dotados de sistemas embarcados e conexão com a Internet capazes de receber, armazenar e enviar dados. IoT é altamente adequada para utilização na construção da infra-estrutura necessária para a implementação de soluções aos grandes desafios das cidades sustentáveis [6], [7].

Aliando a tecnologia a sustentabilidade, torna-se possível permitir ao cidadão participar da gestão pública, informando situações irregulares e acompanhando às ações do Gestor público na solução dos problemas.

Diretamente relacionado à saúde pública e contaminação ambiental, impactos sociais e econômicos, a gestão de Resíduos sólidos é de interesse coletivo [5]. A gestão correta dos resíduos sólidos é fator essencial para o avanço nas questões socioambientais, cabendo ao poder público e à sociedade estabelecer e executar medidas com o objetivo de garantir melhoria da qualidade de vida da população.

Este estudo apresenta um sistema que permite aos cidadãos realizar denúncias de descartes irregulares de resíduos sólidos. Possibilita ainda que o cidadão possa acompanhar o andamento das providências tomadas pela administração pública em relação as suas denúncias e ao conjunto de denúncias existentes no município. O objetivo do desenvolvimento desse sistema é o de disponibilizar para o gestor público e para a comunidade uma forma de colaboração para a gestão de um problema que afeta a qualidade de vida das pessoas. É formado por um programa aplicativo, que executa em smartphones com sistema operacional Android e por um site, que permite ao gestor tomar conhecimento das denúncias para a execução de ações para a solução dos problemas.

Este artigo está organizado da seguinte maneira:

A seção II aborda os sistemas que possuem relação com o aplicativo nomeado de Eco-on. A seção III apresenta uma visão geral do que é o Eco-on. A seção IV apresenta como o Eco-on foi implementado. A seção V detalha a implementação do sistema, a seção VI é dedicada a uma discussão do sistema implementado. Por fim, a seção VII apresenta as conclusões deste trabalho.

## **2 SISTEMAS RELACIONADOS**

A tecnologia está mudando rapidamente a maneira como ocorre a interação das pessoas com o meio ambiente. Visando atender às mais novas demandas dos consumidores, empresas estão desenvolvendo hoje produtos com interfaces tecnológicas que seriam inimagináveis há uma década. Sistemas automatizados que acendem as luzes e aquecem o jantar ao perceber que você está retornando do trabalho para casa, pulseiras e palmilhas inteligentes que compartilham com seus amigos o quanto você andou a pé ou de bicicleta durante o dia na cidade ou sensores que avisam automaticamente aos fazendeiros quando um animal está doente ou prenhe. Todos esses exemplos são manifestações consideradas tecnologias inovadoras associadas ao conceito que vem sendo construído de internet das coisas (Internet of things, IoT) [7].

A Internet das coisas tem sido uma aliada na construção de soluções para inúmeras questões relacionadas ao meio ambiente e a gestão pública. Em [8] é apresentado um sistema de monitoramento de florestas usando LoRa e VANTs. O artigo contém um estudo de um ambiente agrícola. Foi utilizado um módulo de comunicação e vários nós sensores (umidade, temperatura, luminosidade) conectados a um Arduíno UNO, com uma placa de comunicação. Foi usado um pequeno drone com um módulo gateway montado em sua fuselagem. O gateway possui uma placa de comunicação, uma CPU que roda Linux e rede wireless. O Arduíno coleta dados dos sensores em terra e envia para o gateway no Drone, que armazena localmente ou envia para um servidor na nuvem via uma interface de rede. O objetivo deste projeto é ajudar os agricultores a obter dados ambientais sobre um campo agrícola geograficamente grande e também em locais de difícil acesso ou perigosos para acessar.

O trabalho descrito em [9] apresenta um aplicativo móvel chamado RecycHong, com o qual os cidadãos poderão participar da gestão dos resíduos da cidade de Hong Kong. Segundo os autores, a capacidade dos aterros sanitários de Hong Kong estará esgotada em alguns anos e não existe uma política de reciclagem totalmente implementada. O aplicativo, será usado para gerenciar uma política pública, que tem um cunho social, com o cidadão contribuindo para a solução de um problema relevante.

RecycHong considera três categorias de resíduos: a primeira abrange alimentos, papel e têxtil, a segunda categoria contém metal, vidro e plásticos e a terceira categoria contém óleos e materiais com riscos biológicos.

Em uma primeira fase o aplicativo de reciclagem irá fornecer informações sobre reciclagem extraídas de sites governamentais de Hong Kong. A segunda fase será concentrada na participação do cidadão, que poderá adicionar informações sobre instalações de reciclagem e mapear geograficamente sua localização. Os usuários poderão também sugerir novas maneiras de reutilização dos resíduos reciclados e fazer doação de coisas que não serão mais utilizadas. Será implementado um sistema de coleta residencial gratuita.

O sistema apresentado em [10] tem como principal objetivo monitorar a qualidade do ar, prever, avaliar e identificar tendências no estado da atmosfera em uma cidade, para prevenir conseqüências negativas que afetam adversamente a saúde humana e o meio ambiente. Permite o monitoramento em tempo real de condições ambientais e meteorológicas e monitora todas as principais fontes de poluição. O sistema possui várias estações locais (coletores de dados), localizadas em diferentes partes da cidade, e uma

estação base. A estrutura da estação local inclui sensores para medir os parâmetros necessários, dispositivos de processamento eletrônico e um módulo de comunicação. A estação base é estacionária. Também contém sensores, dispositivos de processamento eletrônico, bem como equipamento para transmissão de informações. Os dados recebidos das estações local e base são transmitidos a um servidor e pode ser acessado pelo usuário do sistema através de uma interface da web ou aplicativo móvel.

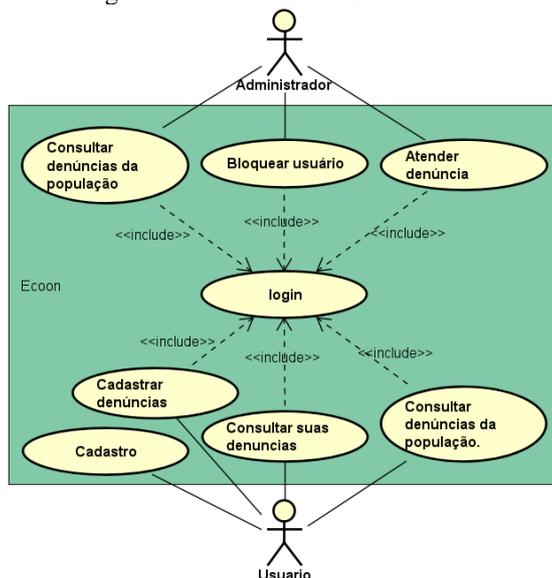
### **3 VISÃO GERAL DO ECO-ON**

Eco-on é um sistema projetado para ser utilizado pelos cidadãos para fazer denúncias de descartes irregulares de resíduos sólidos. Conta com dois tipos de usuário: cidadão e administrador (gestor do meio ambiente). O cidadão usa um aplicativo para notificar o gestor público da existência e da localização de um descarte irregular de resíduos. Essa notificação é formada pela localização e por fotos dos resíduos encontrados. Essas informações são armazenadas em um servidor web e podem ser acessadas por um cliente rodando em um computador (Laptop ou desktop).

O usuário instala o aplicativo e poderá realizar um cadastro (se desejar poderá permanecer anônimo). O usuário poderá cadastrar denúncias, com a opção de deixá-la como anônima, para preservar sua identificação, também poderá acompanhá-las, para ver as providências tomadas pelo gestor e também pode observar as denúncias de outros usuários, o que oferece uma visão geral das denúncias existentes sobre descartes irregulares e a efetividade das ações executadas pela administração pública.

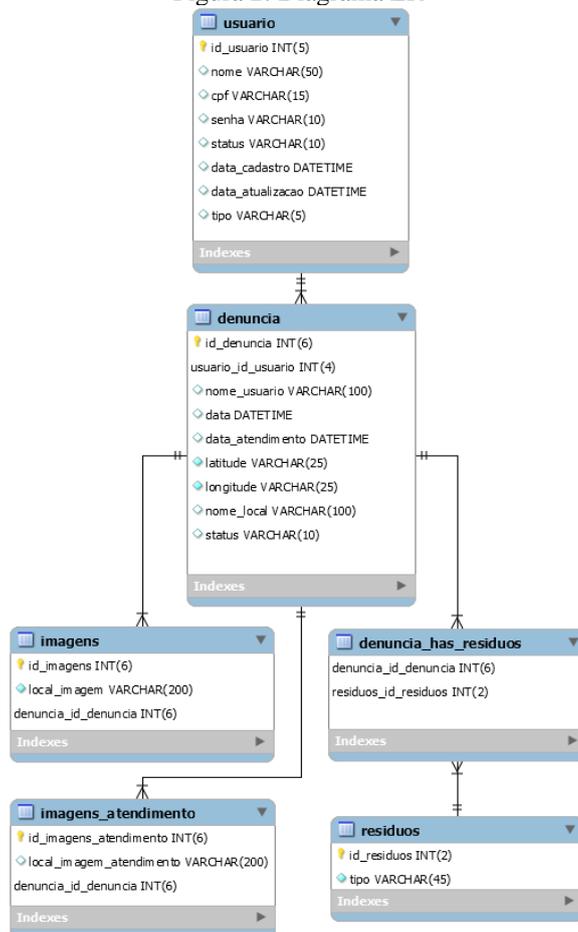
A aplicação web (servidor e cliente) foi desenvolvida para ser utilizada pelo gestor do meio ambiente. Tem como função gerenciar as denúncias cadastradas pelos usuários e os próprios usuários. Isso inclui excluir denúncias falsas, bloquear usuários que utilizam o aplicativo de má fé, atualizar as denúncias ao serem resolvidas, ter acesso a um mapa completo das denúncias existentes e realizar análises para um melhor entendimento da natureza dos descartes, gerando conhecimento para a adoção de soluções eficientes para o ambiente e o bem-estar da comunidade. A figura abaixo (Figura 1) mostra o conjunto de funcionalidades do sistema.

Figura 1: Funcionalidades do sistema



A figura a baixo (figura 2) mostra o diagrama do banco de dados do sistema, onde são armazenados todos os dados necessários referentes as denúncias realizadas.

Figura 2: Diagrama ER



#### 4 IMPLEMENTAÇÃO DO ECO-ON

O Eco-on pode ser separado em três partes: API, aplicativo e site.

API, acrônimo inglês de Application Programming Interface, é um conjunto de rotinas e padrões estabelecidos por um software para a utilização das suas funcionalidades por aplicativos que não pretendem envolver-se com detalhes da implementação, mas apenas usar seus serviços.

A API fica hospedada em um servidor na internet, sendo responsável por oferecer serviços ao aplicativo e ao site.

Os serviços que a API disponibiliza ao aplicativo são:

A. Cadastro do usuário.

Armazenamento dos dados do usuário no banco de dados, que são nome, CPF, senha, status, data de cadastro, data de atualização do cadastro e o tipo de nível de usuário (usuário ou administrador).

B. Login de usuário.

Verificação da existência do cadastro do usuário para permitir acesso ao aplicativo.

C. Cadastro de denúncia.

Armazenamento dos dados da denúncia no banco de dados. São armazenados os dados principais da denúncia, como: o código do usuário que realizou a denúncia, nome, data da denúncia, data do atendimento, latitude, longitude, endereço do local e o status da denúncia. Também são armazenados os tipos de resíduos encontrados (uma lista pré-determinada é mostrada no app) e as fotos do local da denúncia.

D. Visualização de suas próprias denúncias.

Acesso às denúncias que o usuário já realizou.

E. Visualização de denúncias da cidade.

Acesso a um mapa completo contendo o horário, local, e o nome das denúncias existentes na cidade, realizadas pelo próprio usuário e também por outros usuários.

Para o site, os serviços que a API oferece são:

A. Visualização do cadastro dos usuários.

Acesso ao cadastro de todos os usuários no banco de dados.

B. Bloqueio e desbloqueio de usuário.

Ação para permitir o uso ou não do aplicativo pelo usuário.

C. Visualização de denúncias.

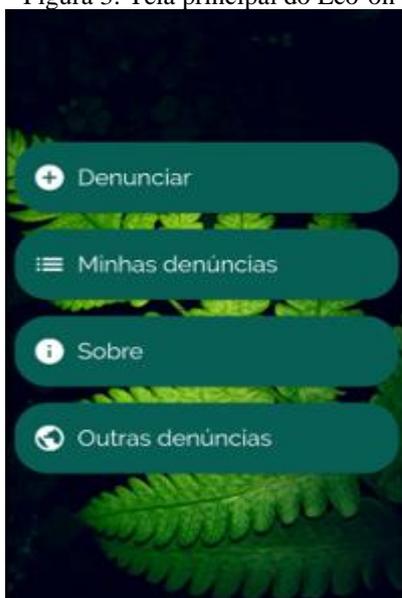
Acesso a todas as denúncias cadastradas no banco de dados. Poderão ser visualizados os seguintes dados: Código da denúncia, código do usuário, nome, data, data de atendimento, latitude, longitude, endereço, status e fotos.

D. Atendimento da denúncia.

Cadastro de fotos do local limpo ou de apenas um status sinalizando uma denúncia falsa.

O aplicativo é a principal união do cidadão com a gestão pública. É com ele que o cidadão poderá tirar fotos, selecionar os tipos de resíduos, o local e irá registrar a denúncia com seu nome ou como anônimo. A tela principal do aplicativo é apresentada na Figura 3, a seguir.

Figura 3: Tela principal do Eco-on



O aplicativo desenvolvido ocupa pouco espaço de armazenamento, o que possibilita que possa ser utilizado pela maioria dos smartphones.

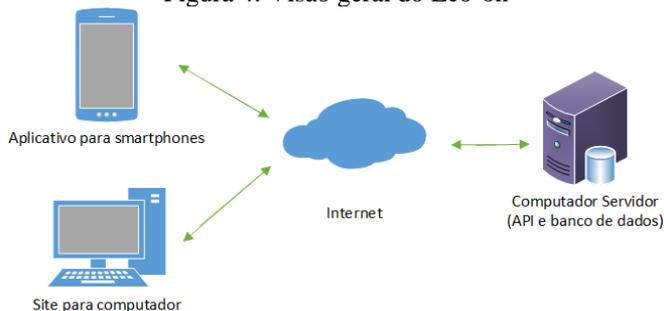
O Eco-on se comunica com a API do servidor pela internet, enviando dados relativos a uma denúncia e consultando informações referentes as denúncias existentes.

O Eco-on possui um site que contém um painel administrativo. O painel é de uso exclusivo de um responsável do meio ambiente, onde ele poderá acompanhar as denúncias cadastradas. Isso envolve o atendimento da denúncia, que pode ser o registro da mesma como falsa ou o registro como atendida. Se o responsável se deparar com fotos não relacionadas com a denúncia, ele pode registrá-la como falsa. Se as imagens parecerem ser verdadeiras, o responsável pode contatar uma equipe especializada, que tomará providências para a retirada dos resíduos. Após a limpeza do local, devem ser feitas fotos do local limpo, para que o responsável anexe á denuncia e a registre como atendida. Quando o usuário responsável pela denuncia acessar o aplicativo, poderá conferir as imagens do local limpo, fechando assim um ciclo da colaboração do cidadão com a gestão pública.

O site também oferece um controle para os usuários do Eco-on. Além de o responsável ter acesso aos dados cadastrais, pode bloquear o usuário, em casos de denúncias falsas, não permitindo o acesso ao aplicativo ou a realização de um novo cadastro com o mesmo CPF.

A figura abaixo (Figura 4) ilustra a infra-estrutura da comunicação do sistema Eco-on. As setas indicam a comunicação entre cada dispositivo. O aplicativo no Smartphone armazena e visualiza os dados via API de acesso aos dados armazenados no banco de dados. Da mesma forma, o cliente da aplicação web acessa esses mesmos dados no servidor web, também via API.

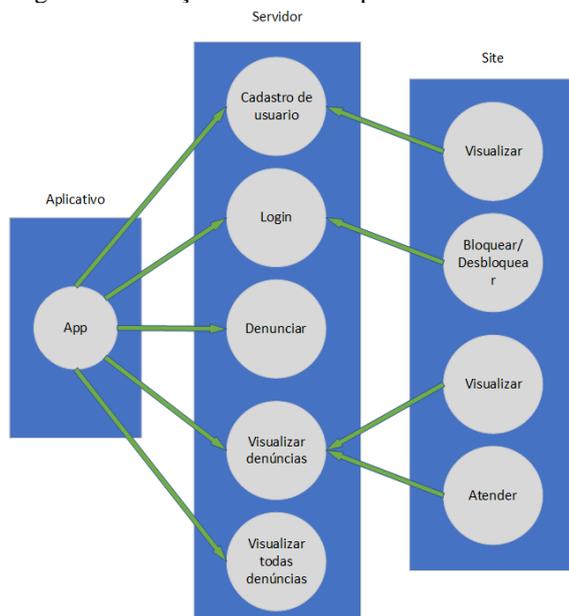
Figura 4: Visão geral do Eco-on



## 5 DETALHES DA IMPLEMENTAÇÃO DO ECO-ON

A figura abaixo, Figura 5, apresenta uma visão mais detalhada da interação entre os componentes do Eco-on.

Figura 5: Interação entre os componentes do Eco-on



As principais funções que rodam no aplicativo, na API do servidor e no cliente web (site) serão apresentadas a seguir:

### A. Funções que rodam no aplicativo

*Future<void> buscaDenunciasPublico() async {...}* faz uma requisição para o servidor, que retorna latitude, longitude, nome do autor da denúncia e data da denúncia, de todas as denúncias cadastradas no sistema.

**Future<void> buscaDenuncias(String idusuario, BuildContext context) async {...}** faz uma requisição para o servidor enviando o id do usuário. O servidor retorna todos os dados das denúncias associadas ao id do usuário.

**Future<void> buscaDenuncia(String idusuario, String iddenuncia, BuildContext context) async {...}** faz uma requisição para o servidor enviando o id do usuário e o id de uma denúncia específica. O servidor retorna todos os dados da denuncia identificada pelo id do usuário e pelo id da denúncia.

**void logar(String cpf, String senha, BuildContext context) async { ...}** faz uma requisição para o servidor enviando o cpf e senha do usuário. O servidor retorna os dados completos do usuário, liberando acesso ao aplicativo.

**Future<void> cadastrar(String nome, String cpf, String senha, String confirmsenha, BuildContext context) async { ...}** envia para o servidor nome, CPF e senha, que o usuário informou no cadastro. Esses dados serão armazenados na base de dados.

**Future<void> cadastraDenuncia(String idusuario, String data, String latitude, String longitude, String local, List<File> images, List<String> residuos, String cpf, String nomeusuario, BuildContext context) async { ...}** envia para o servidor o id do usuário, data da denúncia, latitude, longitude, local (endereço), lista de imagens, lista de resíduos, cpf, e nome do usuário. Esses dados serão armazenados na base de dados, associados ao usuário que está fazendo a denúncia.

**void verificaLogin(){ ...}** faz uma requisição para o servidor que retorna os dados do login, caso exista, ou retorna uma mensagem informando que não existe o login informado.

#### B. Funções executadas pelo cliente (site)

**Future<void> atenderDenuncia(String cpf, String iddenuncia) async { ...}** faz uma requisição para o servidor enviando as fotos do local limpo e troca o status da denuncia para atendida. Em caso de denúncia falsa, apenas o status da denúncia é trocado para falsa.

**Future<void> buscaDenuncias(String idusuario, BuildContext context) async { ...}** faz uma requisição para o servidor enviando o id do usuário. O servidor retorna todos os dados das denúncias que possuem relação com o id do usuário.

**Future<void> buscaDenuncia(String idusuario, String iddenuncia, BuildContext context) async { ...}** faz uma requisição para o servidor enviando o id do usuário, id da denúncia. O servidor retorna todos os dados de uma única denuncia que possui relação com o id do usuário e possui o id da denúncia.

**void logar(String cpf, String senha, BuildContext context) async { ...}** faz uma requisição para o servidor enviando o cpf e senha do usuário, que retorna os dados completos do usuário, liberando acesso ao aplicativo.

**void verificaLogin(){ ...}** faz uma requisição para o servidor que retorna os dados do login, caso seja de um usuário cadastrado, ou retorna uma mensagem informando que não existe o login informado.

**Future<void> bloquearOuDesbloquearUsuario() async { ...}** faz uma requisição para o servidor solicitando o bloqueio ou desbloqueio do usuário. Se o usuário estiver com status “ativo”, ele será bloqueado. Se estiver com status “Bloqueado” ele se tornará “Ativo”.

**Future<void> buscaUsuario(String idusuario, BuildContext context) async { ...}** faz uma requisição para o servidor enviando o id do usuário, que retorna os dados completos do cadastro do usuário.

**Future<void> buscaUsuarios () async { ...}** faz uma requisição para o servidor que retorna os dados completos de todos os usuários cadastrados no sistema.

### C. API do servidor

A seguir serão apresentadas as principais funções da API do servidor, que implementam a comunicação com o aplicativo e com o site. Essas funções manipulam os dados do banco de dados e retornam esses dados para o aplicativo ou para o site, no momento de uma requisição.

**exports.verTodosUsuarios = async(req, res, next) => { ...}** recebe uma requisição (req), busca os dados dos usuários no banco de dados e retorna esses dados para o site.

**exports.verUsuario = async(req, res, next) => { ...}** essa função recebe uma requisição (req), acessa os dados do usuário no banco de dados e retorna esses dados para o site ou para o aplicativo.

**exports.verificarCPFUsuario = async(req, res, next) => { ...}** essa função recebe uma requisição (req) com o cpf do usuário, verifica se o cpf já existe no banco de dados e retorna uma mensagem indicando se existe ou não.

**exports.cadastraUsuario = async(req, res, next) => { ...}** função que recebe uma requisição (req) contendo os dados que o usuário inseriu no aplicativo, que serão armazenados no banco de dados. Retorna para o aplicativo um código de registro no banco (id do usuário).

**exports.cadastraUsuario = async(req, res, next) => { ...}**

recebe uma requisição (req) com os dados que o usuário alterou no aplicativo ou com os dados que foram alterados pelo administrador no site, que são armazenados no banco de dados.

**exports.bloquearUsuario = async(req, res, next) => { ...}** função que recebe uma requisição (req) sem dados. Essa função apenas compara o status do usuário, se “ativo” passará para bloqueado, se “Bloqueado”, passará para “Ativo”.

**exports.verDenuncia = async(req, res, next) => { ...}** essa função recebe uma requisição (req) com o id da denúncia e o id do usuário e retorna os dados da denúncia que contenham os id's informados.

**exports.verDenunciaResiduos = async(req, res, next) => { ...}** essa função recebe uma requisição (req) com o id da denúncia e o id do usuário e retorna os dados dos resíduos que contenham os id's informados.

**exports.verDenunciaFotos = async(req, res, next) => { ...}** essa função recebe uma requisição (req) com o id da denúncia e o id do usuário e retorna as fotos da denúncia que contenham os id's informados.

**exports.verDenuncias = async(req, res, next) => { ...}** função que recebe uma requisição (req) com o id do usuário e retorna os dados de todas as denúncias relacionadas com o id do usuário. Esta função é usada apenas pelo cliente (site).

**exports.verDenunciasUsuario = async(req, res, next) => { ...}** função que recebe uma requisição (req) com o id do usuário e retorna os dados de todas as denúncias relacionadas com o id do usuário. Essa função é usada apenas pelo aplicativo

**exports.cadastraDenuncia = async(req, res, next) => { ...}** essa função recebe uma requisição (req) com os dados principais da denuncia e os armazena no banco de dados.

**exports.cadastraFotos = async(req, res, next) => { ...}** essa função recebe uma requisição (req) com o id da denúncia e as fotos enviadas pelo usuário. As fotos são armazenadas em uma pasta no servidor e o caminho de acesso às fotos é salvo no banco de dados.

**exports.cadastraFotosAtendimento = async(req, res, next) => { ...}** essa função recebe uma requisição (req) com o id da denúncia e as imagens selecionadas pelo administrador. As fotos são armazenadas em uma pasta no servidor e o caminho de acesso às fotos é salvo no banco de dados.

**exports.verDenunciaFotosAtendimento = async(req, res, next) => { ...}** função que recebe uma requisição (req) com o id da denúncia e retorna as fotos que foram cadastradas pelo administrador.

**exports.cadastraResiduos = async(req, res, next) => { ...}** função que recebe uma requisição (req) com o id da denúncia e retorna os resíduos que possuem relação com a id da denúncia.

**exports.verTodasDenuncias = async(req, res, next) => { ...}** recebe uma requisição (req) sem parâmetros e retorna os dados de todas as denúncias cadastradas no banco de dados.

**exports.atualizaDenuncia = async(req, res, next) => { ...}** recebe uma requisição (req) com os dados que o administrador alterou e os salva no banco de dados.

## 6 DISCUSSÃO

Após o desenvolvimento do aplicativo, o mesmo foi testado no município de São José do Herval, no Rio Grande do Sul. Tal ação preconizou a redução de impactos a curto e médio prazos, a melhoria do gerenciamento dos resíduos e, conseqüentemente, melhoria do bem-estar da população do município. A classificação dos resíduos foi feita conforme descrito na Política Nacional dos Resíduos Sólidos (Lei Federal 12.305/2010), sendo eles: resíduos sólidos domiciliares secos, resíduos sólidos domiciliares úmidos, resíduos verdes, resíduos volumosos, resíduos industriais, resíduos de agrossilvopastoris, resíduos de limpeza pública, resíduos de construção civil, resíduos de serviço de saúde, resíduos de serviços terrestres e resíduos de mineração.

O aplicativo foi disponibilizado na loja *Play Store*, a qual possui o acesso gratuito, sendo necessário somente a pesquisa pelo nome do aplicativo e proceder a instalação no smartphone. Para fazer o uso corretamente do aplicativo é necessário que o aparelho celular utilize a câmera fotográfica, o GPS (*Global Positioning System*) tenha conexão com a internet (Wi-Fi ou dados móveis).

O aplicativo tem a interface de fácil manuseio para que as possíveis denúncias possam ser efetuadas com agilidade e rapidez. Ao final do todo o processo de cadastramento das denúncias, o usuário poderá obter maiores informações sobre o andamento das mesmas, através do item do menu principal "*Minhas denúncias*". Para controle e gerenciamento das denúncias pela municipalidade, a mesma poderá ser classificada como aberta ou fechada. A primeira significa que os materiais ainda se encontram dispostos no mesmo local e a segunda que o material já foi retirado pelo órgão responsável por acompanhar a denúncia.

Durante o período de testes do aplicativo, de julho a agosto de 2019, foram computadas 23 denúncias e classificadas em: 28,85%, resíduos domiciliares secos, 25%

resíduos volumosos, 19,23% resíduos industriais, 9,62% resíduos verdes, 7,69% resíduos da construção civil 5,77% resíduos de limpeza pública e 3,84% resíduos domiciliares orgânicos.

Observou-se que o sistema desenvolvido contribuiu efetivamente na solução de identificação dos descartes irregulares de resíduos. O mesmo apresentou facilidade de utilização e de integração com o site, permitindo maior rapidez de localização dos descartes irregulares, assim como proporcionou à comunidade uma maior rapidez na resolução do problema.

É importante destacar que esse sistema somente irá obter os resultados esperados com o comprometimento do gestor público. É indispensável que haja essa integração da comunidade com a administração do município. Outro fator igualmente importante é a sensibilização ambiental da comunidade e a divulgação do aplicativo, que deve ser feita pelos órgãos públicos do município responsáveis pela gestão dos resíduos sólidos.

## 7 CONCLUSÃO

O objetivo do estudo, o desenvolvimento de um aplicativo, o Eco-on, e de um site para o gerenciamento de resíduos sólidos, foi alcançado. O sistema está operacional e mostrou-se de fácil manuseio, possibilitando que as denúncias possam ser feitas por usuários comuns, não especialistas em tecnologia.

Após o desenvolvimento do aplicativo, o mesmo foi testado no município de São José do Herval, no Rio Grande do Sul. A classificação dos resíduos baseou-se Lei Federal 12.305/2010. Foram realizadas 23 denúncias de resíduos dispostos em locais impróprios. Destes, 28,85%, resíduos domiciliares secos, 25% resíduos volumosos, 19,23% resíduos industriais, 9,62% resíduos verdes, 7,69% resíduos da construção civil 5,77% resíduos de limpeza pública e 3,84% resíduos domiciliares orgânicos.

A partir do site o gestor público pode visualizar as denúncias e atuar efetivamente na resolução do problema de uma maneira mais rápida, diminuindo os impactos ambientais gerados pelos resíduos sólidos depositados em locais impróprios.

Cabe ressaltar que o comprometimento da administração pública é fator essencial para que os cidadãos possam efetivamente participar na solução do problema da gestão dos descartes irregulares.

Uma contribuição importante desse trabalho é possibilitar a integração entre a comunidade e o poder público para a solução de grandes problemas, típicos da vida nas

idades, com o uso de soluções tecnológicas, na busca de melhoria de vida para a população.

## REFERÊNCIAS

1. Schröder, Carolin: A mobile app for citizen participation. - In: EGOSE 2014 : proceedings : International Conference Electronic Governance and Open Society: Challenges in Eurasia : St. Petersburg, Russia, 18-20 November 2014. - New York, New York: Association for Computing Machinery, 2014. ISBN: 978-1-4503-3401-3. - pp. 75–78. - DOI: 10.1145/2729104.2729137
2. Manju Khari, Manoj Kumar, Sonakshi Vij, Priyank Pandey, and Vaishali. 2016. Smart Cities: A Secure Data Transmission Model. In Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies (ICTCS '16). Association for Computing Machinery, New York, NY, USA, Article DOI:<https://doi.org/10.1145/2905055.2905216>.
3. Dora Marinova. 2017. Sustainability and Smart Technology. In Proceedings of the 26th International Conference on World Wide Web Companion (WWW '17 Companion). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 1529. DOI:<https://doi.org/10.1145/3041021.3054767>
4. Pearce, David. Green Economics. Environmental Values 1, no. 1. (1992): 3-13. <http://www.environmentandsociety.org/node/5454>.
5. ICLEI – Governos Locais pela Sustentabilidade [http://www.iclei.org.br/residuos/site/?page\\_id=349](http://www.iclei.org.br/residuos/site/?page_id=349). Acesso em 19 março 2020.
6. Rodger Lea and Michael Blackstock. 2014. Smart Cities: an IoT-centric Approach. In Proceedings of the 2014 International Workshop on Web Intelligence and Smart Sensing (IWWISS '14). Association for Computing Machinery, New York, NY, USA, 1–2. DOI:<https://doi.org/10.1145/2637064.2637096>
7. Farzad Samie, Lars Bauer, and Jörg Henkel. 2016. IoT technologies for embedded computing: a survey. In Proceedings of the Eleventh IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis (CODES '16). Association for Computing Machinery, New York, NY, USA, Article 8, 1–10. DOI:<https://doi.org/10.1145/2968456.2974004>
8. Sunnyeo Park, Shinhye Yun, Hyunjin Kim, Ryunhwan Kwon, John Ganser, and Smith Anthony. 2018. Forestry Monitoring System using LoRa and Drone. In Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics (WIMS '18). Association for Computing Machinery, New York, NY, USA, Article 48, 1–8. DOI:<https://doi.org/10.1145/3227609.3227677>
9. Julija Naskova. 2017. RecycHongs: Mobile App Co-design. In Proceedings of the 2017 ACM Conference Companion Publication on Designing Interactive Systems (DIS '17 Companion). Association for Computing Machinery, New York, NY, USA, 337–340. DOI:<https://doi.org/10.1145/3064857.3079188>
10. Kulyanda S. Duisebekova, Zhibek N. Sarsenova, Viktor T. Pygay, Zamira N. Tuyakova, Nurzhan T. Duzbayev, Altay Z. Aitmagambetov, and Saule T. Amanzholova. 2019. Environmental monitoring system for analysis of climatic and ecological changes

using LoRa technology. In Proceedings of the 5th International Conference on Engineering and MIS (ICEMIS '19). Association for Computing Machinery, New York, NY, USA, Article 15, 1–6. DOI:<https://doi.org/10.1145/3330431.3330446.25/03/2020>