



Cidade
Limpa

Aplicativo de Monitoramento de
Lixeiras para Cidades Inteligentes

Grupo de cinco alunos do
curso de **ANÁLISE E
DESENVOLVIMENTO DE
SISTEMAS 2024/1 - FIAP**

Adriano Lombardi (RM98113)
Alberto Frigatto (RM97807)
Danilo Fernando (RM99542)
Gustavo Lopes (RM99316)
Renato Brandão (RM97821)



Conteúdo

Objetivos -----	03
Tecnologia Escolhida -----	05
Aplicabilidade em Cidades Inteligentes -----	06
Telas do Projeto -----	09
Endereço (https) e regras do serviço -----	11
Conclusão -----	26

Objetivos



1. Localizar Lixeiras Próximas:

Facilita o descarte responsável e reduz o abandono de resíduos em locais impróprios.



2. Verificar Capacidade e Quantidade de Lixo:

Previne a superlotação de lixeiras e contribui para um planejamento de coleta mais eficiente.



Cidade
Limpa



3. Contribuir para a Eficiência da Coleta de Lixo:

Auxilia na redução de custos operacionais e na manutenção da limpeza urbana.



4. Promover a Consciência Ambiental:

Encoraja práticas sustentáveis e educa o público sobre gestão de resíduos.



**Cidade
Limpa**



Tecnologia Escolhida

Kotlin

- Menor risco de bugs com segurança de tipo e gerenciamento de nulidade.
- Maior produtividade com sintaxe concisa e moderna.
- Integração fácil com a vasta biblioteca existente de Java.

Jetpack Compose

- UIs mais intuitivas com uma abordagem declarativa.
- Manutenção e atualização do código simplificadas.
- Apoio da comunidade e atualizações regulares do Google.

Plataforma Android

- Maior alcance de mercado com diversidade de dispositivos e usuários.
- Flexibilidade e liberdade para personalizar o aplicativo.
- Apoio de um ecossistema robusto de desenvolvimento e distribuição.



Aplicabilidade em Cidades Inteligentes

Parte 1

Governança Inteligente (Smart Governance):

- Proporciona aos cidadãos meios para influenciar a gestão de resíduos e promove a reciclagem ativa.
- Utiliza dados para aprimorar as operações de coleta, resultando em economia para os serviços municipais.

Ambiente Inteligente (Smart Environment):

- Ajuda a diminuir a emissão de poluentes através de uma gestão de resíduos mais inteligente.
- Incentiva hábitos de reciclagem que levam a uma menor dependência de aterros.

Aplicabilidade em Cidades Inteligentes

Parte 2

Vida Inteligente (Smart Living):

- Contribui para ambientes mais saudáveis, prevenindo o acúmulo de resíduos que podem causar doenças.
- O aplicativo facilita o encontro de pontos de descarte e promove a educação sobre a importância de práticas sustentáveis.

Economia Inteligente (Smart Economy):

- Fomenta o avanço tecnológico e cria espaço para o surgimento de novas empresas no setor de sustentabilidade.
- Traz eficiência operacional que pode reduzir os custos com gestão de resíduos ao longo do tempo.





Aplicabilidade em Cidades Inteligentes

Parte 3

Pessoas Inteligentes (Smart People):

- Aumenta a sensibilização sobre temas ambientais, reforçando a importância de cada indivíduo na reciclagem.
- Promove a colaboração entre moradores para manter a cidade limpa e sustentável.

Mobilidade Inteligente (Smart Mobility):

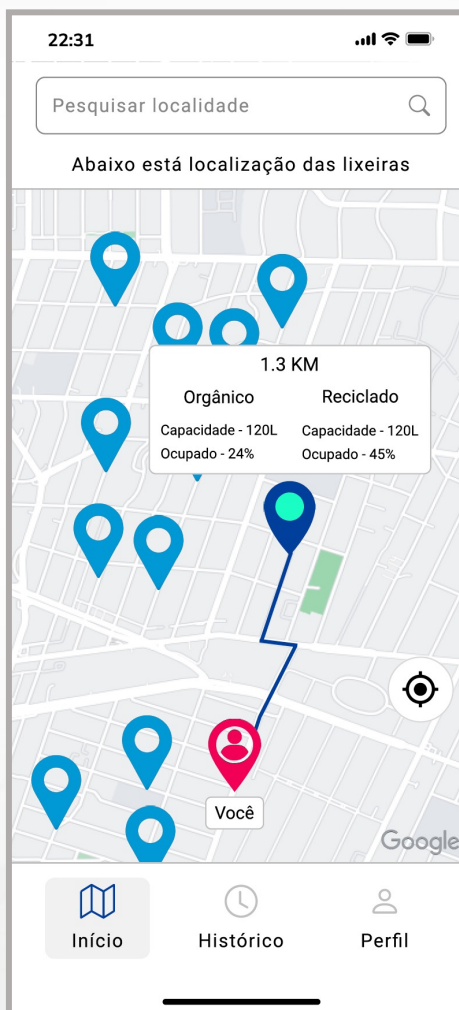
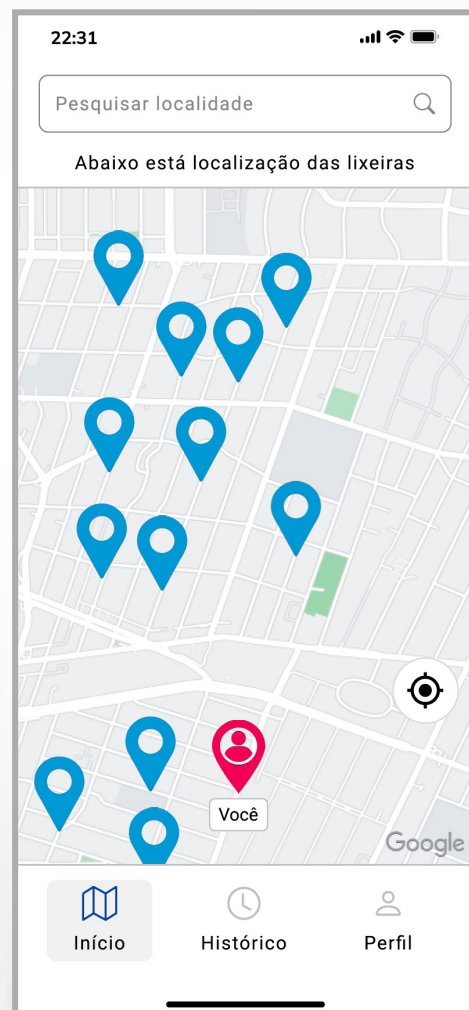
- Melhora a eficácia das rotas de coleta de resíduos, economizando tempo e reduzindo o impacto no tráfego.
- Contribui para um trânsito mais fluido ao minimizar o movimento dos veículos de coleta.

Telas do Projeto

Menu



Telas do Projeto



Endereço (https) e regras do serviço

APIS:

Consulta de CEPs:

Endereço:

<https://h-apigateway.conectagov.estaleiro.serpro.gov.br/api-cep/v1/consulta/cep/{cep}>

Exemplo de utilização:

```
https://h-apigateway.conectagov.estaleiro.serpro.gov.br/api-cep/v1/consulta/cep/60130240
```

Retorno:

```
{
  "cep": "60130240",
  "tipoCep": "logradouro",
  "subTipoCep": "S",
  "uf": "CE",
  "cidade": "Fortaleza",
  "bairro": "São João do Tauape",
  "endereco": "Avenida Pontes Vieira",
  "complemento": "De 2 Até 1550 Lado Par",
  "codigoIBGE": ""
}
```

Endereço (https) e regras do serviço


ONDE TEMOS:

```
{  
  "cep": "string",  
  "tipoCep": "string",  
  "subTipoCep": "string",  
  "uf": "string",  
  "cidade": "string",  
  "bairro": "string",  
  "endereco": "string",  
  "complemento": "string",  
  "codigoIBGE": "string"  
}
```


Endereço (https) e regras do serviço

DESENVOLVIMENTO API:

```
{
  "openapi": "3.0.0",
  "info": {
    "version": "1.0.2",
    "title": "API Consulta CEP",
    "description": "A API `Consulta CEP` é disponibilizada através de serviço `HTTP REST`. O
objetivo da API Consulta CEP é disponibilizar um determinado endereço a partir do CEP\n##
Campos retornados \n \n| Nome | Tipo | Conteúdo | \n| ----- | ----- | :----
----- | \n cep | integer | Ex: 60130240 | \n tipoCep |
string | Possibilidades: logradouro, localidade, cx_postal, gr_usuario, uni_operacional | \n
subtipocep | string | Ex: S | \n uf | string | Ex: CE | \n cidade | string | Ex:
Fortaleza | \n bairro | string | Ex: São João do Taupe | \n endereco | string | Ex:
Avenia Pontes Vieira | \n complemento | string | Ex: \"- até 1550 - lado par\" | \n codigoBGE
| string | Ex: | \n \n\n# Authentication\n\n<!-- ReDoc-Inject: <security-definitions> -->"
  },
}
```



```

"paths": {
  "/consulta/cep/{cep}": {
    "get": {
      "tags": [
        "Consultar CEP"
      ],
      "summary": "Consulta CEP",
      "x-auth-type": "Application & Application User",
      "x-throttling-tier": "Unlimited",
      "description": "\nApresenta o endereço referente a um CEP\n\n ## Exemplo\n\n #####  

Chamada\n \n  https://h-apigateway.conectagov.estaleiro.serpro.gov.br/api-  

cep/v1/consulta/cep/60130240\n \n ##### Retorno\n { \n  \"cep\": \"60130240\", \n  

\"tipoCep\": \"\"logradouro\", \n  \"subTipoCep\": \"S\", \n  \"uf\": \"CE\", \n  \"cidade\":  

\"Fortaleza\", \n  \"bairro\": \"São João do Tauape\", \n  \"endereco\": \"Avenida Pontes  

Vieira\", \n  \"complemento\": \"De 2 Até 1550 Lado Par\", \n  \"codigoIBGE\": \"\" \n }\",  

      "parameters": [
        {
          "in": "header",
          "name": "x-cpf-usuario",
          "description": "CPF do usuário da requisição",
          "required": true,
          "schema": {
            "type": "string",
            "default": ""
          }
        },
        {
          "in": "path",

```



```
"responses": {
  "200": {
    "description": "Retorna os dados de endereço do referentes ao CEP",
    "content": {
      "application/json": {
        "schema": {
          "$ref": "#/components/schemas/ConsultaCep"
        }
      }
    },
  },
  "400": {
    "description": "CEP inválido."
  },
  "404": {
    "description": "CEP não encontrado"
  },
  "500": {
    "description": "Erros do backend e/ou dos servidores de dados"
  }
},
"security": [
  {
    "OAuth2": []
  }
],
```

```
"servers": [  
  {  
    "url": "https://h-apigateway.conectagov.estaleiro.serpro.gov.br/api-cep/v1"  
  }  
],  
"components": {  
  "securitySchemes": {  
    "OAuth2": {  
      "type": "oauth2",  
      "description": "A autenticação é feita usando um token de acesso padrão JWT (JSON Web  
Token). Este token tem duração de 2 horas, recomendamos armazená-lo na aplicação cliente  
para uso nas Requisições ao serviço durante seu tempo de validade. Chamadas desnecessárias  
ao serviço de geração de tokens geram tráfego adicional na rede e podem degradar a  
performance da API, aumentando os tempos de resposta.",  
      "flows": {  
        "clientCredentials": {  
          "tokenUrl": "https://h-apigateway.conectagov.estaleiro.serpro.gov.br/oauth2/jwt-token",  
          "scopes": {}  
        }  
      }  
    }  
  },  
  "schemas": {  
    "ConsultaCep": {  
      "type": "object",  
      "properties": {  
        "cep": {  
          "type": "string"  
        },  
      },  
    },  
  },  
}
```





```
"tipoCep": {  
  "type": "string"  
},  
"subTipoCep": {  
  "type": "string"  
},  
"uf": {  
  "type": "string"  
},  
"cidade": {  
  "type": "string"  
},  
"bairro": {  
  "type": "string"  
},  
"endereco": {  
  "type": "string"  
},  
"complemento": {  
  "type": "string"  
},  
"codigoIBGE": {  
  "type": "string"  
}  
}  
}  
}  
}  
}
```



ACESSO AO GOOGLE MAPS (ABRIR MAPA)

- 1) Necessário instalar SDK do Maps para Android
- 2) No Console do Google Cloud Plataforma, Crie um projeto. Ative a opção de API.
- 3) Gere uma chave de API na pagina de credenciais do Google Cloud.
- 4) Abra arquivo chamado `local.properties` no diretório raiz do projeto e insira chave API:
`GOOGLE_MAPS_API_KEY=YOUR_KEY_HERE`
- 5) Adicione no `build.gradle` a seguinte linha de código:

```
plugins {  
    // ...  
    id 'com.google.android.libraries.mapsplatform.secrets-gradle-plugin'  
}  
  
buildscript {  
    dependencies {  
        // ...  
        classpath "com.google.android.libraries.mapsplatform.secrets-gradle-plugin:secrets-gradle-plugin:1.3.0"  
    }  
}  
  
dependencies {  
    // Dependency to include Maps SDK for Android  
    implementation 'com.google.android.gms:play-services-maps:17.0.0'  
}
```



6) Para transmitir a chave API, adicione as linhas de código no arquivo AndroidManifest.xml

```
<meta-data
    android:name="com.google.android.geo.API_KEY"
    android:value="${GOOGLE_MAPS_API_KEY}" />
```

No arquivo activity_main.xml, adicione:

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
```

```
    <fragment
        class="com.google.android.gms.maps.SupportMapFragment"
        android:id="@+id/map_fragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
```

```
</FrameLayout>
```

No arquivo activity_main, adicione:

```
<fragment xmlns:map="http://schemas.android.com/apk/res-auto"
    class="com.google.android.gms.maps.SupportMapFragment"
    <!-- ... -->
    map:mapId="YOUR_MAP_ID" />
```


ACESSO AO GOOGLE MAPS (ADICIONAR MARCADORES)

1) Logo abaixo a chamado de setContentView, adicione:

```
val mapFragment = supportFragmentManager.findFragmentById(  
    R.id.map_fragment  
) as? SupportMapFragment  
mapFragment?.getMapAsync { googleMap ->  
    addMarkers(googleMap)  
}
```

2) Adicione PlacesReader:

```
package com.google.codelabs.buildyourfirstmap.place
```

```
import android.content.Context  
import com.google.codelabs.buildyourfirstmap.R  
import com.google.gson.Gson  
import com.google.gson.reflect.TypeToken  
import java.io.InputStream  
import java.io.InputStreamReader
```




```
/**
 * Reads a list of place JSON objects from the file places.json
 */
class PlacesReader(private val context: Context) {

    // GSON object responsible for converting from JSON to a Place object
    private val gson = Gson()

    // InputStream representing places.json
    private val inputStream: InputStream
        get() = context.resources.openRawResource(R.raw.places)


    /**
     * Reads the list of place JSON objects in the file places.json
     * and returns a list of Place objects
     */
    fun read(): List<Place> {
        val itemType = object : TypeToken<List<PlaceResponse>>() {}.type
        val reader = InputStreamReader(inputStream)
        return gson.fromJson<List<PlaceResponse>>(reader, itemType).map {
            it.toPlace()
        }
    }
}
```



3) Para adicionar marcadores, utilize:

MainActivity.addMarkes()

```
/**
 * Adds marker representations of the places list on the provided GoogleMap object
 */
private fun addMarkers(googleMap: GoogleMap) {
    places.forEach { place ->
        val marker = googleMap.addMarker(
            MarkerOptions()
                .title(place.name)
                .position(place.latLng)
        )
    }
}
```



MONITORAMENTO DOS SENSORES NAS LIXEIRAS UTILIZANDO ORACLE CLOUD COMO INFRAESTRUTURA CLOUD PARA ESTE PROJETO:

- 1) Criar uma conta Cloud da Oracle ;
- 2) Vincular serviço de monitoramento de IOT (Production Monitoring);
- 3) Acesso ao Oracle IOT Production Monitoring Cloud Service REST API:

`https://instanceName-identityName.dataCenter.oraclecloud.com/productionMonitoring/clientapi/v2/resource-path`

Where:

- instanceName - Name you chose for your Oracle IoT Cloud Service when it was provisioned.
- identityName - Identity name you chose when you signed up with Oracle Public Cloud.
- dataCenter - Data center that your service is set up to use.
- resource-path - Relative path that defines the resource. For a complete list of resources, see [REST Endpoints](#).



EXEMPLO CONSULTA A UMA APLICAÇÃO ESPECÍFICA (CIDADE POR EXEMPLO):

GET <https://myiotcsiotjls-agallardo.uscom-central-1.oraclecloud.com/productionMonitoring/clientapi/v2/products?fields=id,description>

Resposta:



```
{
  "items": [
    {
      "id": "0-MI",
      "description": "Tab"
    },
    {
      "id": "0-AH",
      "description": "Cubes"
    },
    {
      "id": "0-IG",
      "description": "Thermostat"
    }
  ],
  "links": [
    { "rel": "canonical",
      "href": "http://iotserver/iot/api/v1/products" },
    { "rel": "self",
      "href": "http://iotserver/iot/api/products?fields=id,description"
    }
  ]
}
```

PARA AUTENTIFICACAO, UTILIZER VIA CURL A CHAVE DE ACESSO API GERADO PELA PLATAFORMA ORACLE. POR EXEMPLO:

```
curl -u jane@mySmartHome.com:password
```

```
-H "Content-Type: application/json"
```

```
-H "Accept: application/json"
```

```
-X GET
```

```
http://MyServiceName-  
MyIdentityName.iot.MyServiceDataCenter.oraclecloud.com/productionMonitoring/clienta  
pi/v2/private/server
```



CONCLUSÃO



Saúde Pública:

- Redução de vetores de doenças devido ao melhor gerenciamento de resíduos.
- Qualidade de Vida: Melhoria na limpeza dos espaços públicos eleva a satisfação dos moradores.

Engajamento Cívico:

- Aumenta a participação dos cidadãos na gestão eficaz de resíduos.

Prevenção de Alagamentos:

- Menos lixo nas ruas diminui o risco de obstrução dos sistemas de drenagem, reduzindo alagamentos.

Redução de Emissões:

- A otimização das rotas de coleta diminui a emissão de gases poluentes.

Conservação de Recursos:

- A prática da reciclagem preserva recursos naturais.

Biodiversidade:

- A redução do descarte irregular de resíduos melhora os ecossistemas urbanos.

Obrigado!

