

Marcelo Augusto Gonçalves Eliseu

PORTFÓLIO ACADÊMICO

Marcelo Augusto Gonçalves Eliseu

PORTFÓLIO ACADÊMICO

Trabalho de Graduação apresentado à
Faculdade de Tecnologia de Araras -
Antônio Brambilla, como requisito parcial
para obtenção do título de Tecnólogo em
Sistemas para Internet.

Orientadores:

Prof. Me Dhebora Souza Umbelino Silva

Prof. Dr Leonardo de Souza Lima

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

Marcelo augusto Gonçalves Eliseu

PORTFÓLIO ACADÊMICO

Trabalho de Graduação apresentado à Faculdade de Tecnologia de Araras - Antônio Brambilla, como requisito parcial para obtenção do título de Tecnólogo em Sistemas para Internet.

Orientadores:

Prof. Me Dhebora Souza Umbelino Silva

Prof. Dr Leonardo de Souza Lima

Banca Examinadora:

Prof. Me Dhebora Souza Umbelino Silva - Orientadora
Faculdade de Tecnologia de Araras – Antônio Brambilla

Prof. Dr Leonardo de Souza Lima - Orientador
Faculdade de Tecnologia de Araras – Antônio Brambilla

Prof. XXXX – Coordenador do Curso de Sistemas para Internet – Membro I
Faculdade de Tecnologia de Araras – Antônio Brambilla

XXXXXXXX – Membro II
Instituição / Empresa

FIGURAS

<i>Figure 1 – Monolítica / Microserviços.....</i>	<i>11</i>
---	-----------

Sumário

PROJETO 1 - Sistema de Apparking	6
INTRODUÇÃO	8
REVISÃO BIBLIOGRÁFICA	9
MÉTODOS E FERRAMENTAS	11
RESULTADOS.....	15
CONSIDERAÇÕES.....	17
REFERÊNCIAS.....	18

PROJETO 1 - Sistema de Apparking

RESUMO

O sistema de Apparking é um aplicativo de estacionamento inteligente onde o usuário encontra os estacionamentos mais perto da sua localidade, sejam eles de shopping, estacionamentos rotativos ou quaisquer outros segmentos previamente cadastrados no Aplicativo. Contendo a informação de vagas disponíveis e a proximidade de sua localidade, o aplicativo aponta as vagas por meio de sua interface gráfica, a qual foi desenvolvida para ser de fácil compreensão e de operação rápida.

Palavras-chave: Automóvel, Vaga de estacionamento, Estacionamento, Estacionar, Vaga disponível.

ABSTRACT

The Apparking system is an intelligent parking application where the user finds the parking lots closest to their location, be they shopping malls, rotating parking lots or any other segments previously registered in the Application. Containing the information of available vacancies and the proximity of your location, the application points out the vacancies through its graphical interface, which was developed to be easy to understand and quick to operate.

Keywords: Car, Parking space, Parking, Parking, Available space.

INTRODUÇÃO

O aumento no fluxo de carros nas grandes cidades do mundo todo gera um enorme problema também no dimensionamento de estacionamentos nas cidades. É comum nos grandes centros urbanos não encontrar vagas para estacionar. No Distrito Federal, por exemplo, tem-se um déficit de cerca de 40 mil vagas, o que faz com que os motoristas optem por deixar seus carros com flanelinhas nas ruas. Na cidade de São Paulo, estacionar o carro também é uma tarefa muito difícil. A falta de vagas é ainda maior do que no Distrito Federal, chegando a aproximadamente mais de 125 mil vagas. Essa situação gera um enorme transtorno para o trânsito dessas regiões e para a mobilidade urbana das cidades. (Revista Produção Industrial & Serviços – ISSN 2526-1894 - v. 4 n. 1 (2017): A engenharia de produção na indústria) e (ERNST& YOUNG, 2014).

Para contribuir na atenuação deste problema foi proposto um aplicativo que indique locais de estacionamento e exiba vagas livres, informando a melhor rota até o estacionamento e a vaga e auxiliando o motorista para que na hora de sair lembre-o onde foi estacionado o veículo.

Através de sensores que informam quando as vagas estão ou não disponíveis, seria possível alimentar um banco de dados que é acessado posteriormente pelo Aplicativo Móvel por meio de uma API (Application Programming Interface). Deste modo, facilita-se o uso das informações coletadas para serem consumidas independentes da plataforma cliente, podendo ser implementada tanto para aplicativos de smartphones como para placas digitais. As informações atualizadas em tempo real permitem ao usuário visualizar facilmente estacionamentos na região ou vagas disponíveis no estacionamento.

REVISÃO BIBLIOGRÁFICA

Como forma de buscar insights para nosso projeto, fizemos uma pesquisa por implementações similares a que estávamos propondo. Dessa forma, encontramos os sistemas desenvolvidos pela Infocomp, pela Mobhis e pela Inrix.

A infoComp (www.inforcomp.com.br/) que tem no seu produto um sistema de gerenciamento de estacionamentos com uso de sensores. que podem ser instalados de modo autônomo (para informar se está vaga ou ocupada por um sinal luminoso que pode ser visualizado de longa distância) ou em rede que é utilizada para alimentar uma central, alocada em servidores em nuvem. Para o usuário ele se apresenta na forma de um equipamento de sinalização luminoso através de 2 cores verde (vaga disponível) ou vermelha (vaga ocupada) também na forma de painéis indicativos de quantidade de vagas no acesso ao estacionamento.

Já a empresa Mobhis (mobhis.com.br/) que tem no seu produto com sensor para identificação individual das vagas, acesso do usuário com gerenciamento de vagas através de APP, acesso do administrador sendo feito por agentes de trânsito para controle do estacionamento via Web/Painel.

A empresa Inrix (Tecnologia não aplicada) - inrix.com – que identifica as vagas através de ondas ultrassônicas emitidas por dispositivos instalado no automóvel. Com acesso de usuário ao que parece ainda em desenvolvimento pelo GPS do próprio veículo.

Tabela 1 - Comparativo entre as plataformas de gerenciamento de estacionamento

	Interface com motorista	Sistema de gerenciamento	Hospedagem	Sensor
infoComp	Pontos luminosos acima das vagas	Web/Painel	Nuvem	Individual
Mobhis	Pontos luminosos acima das vagas	Web/Painel	Nuvem	Individual
Inrix	Pontos luminosos acima das vagas	Ainda em desenvolvimento	Nuvem	Ondas ultrassônicas

Fonte: o autor.

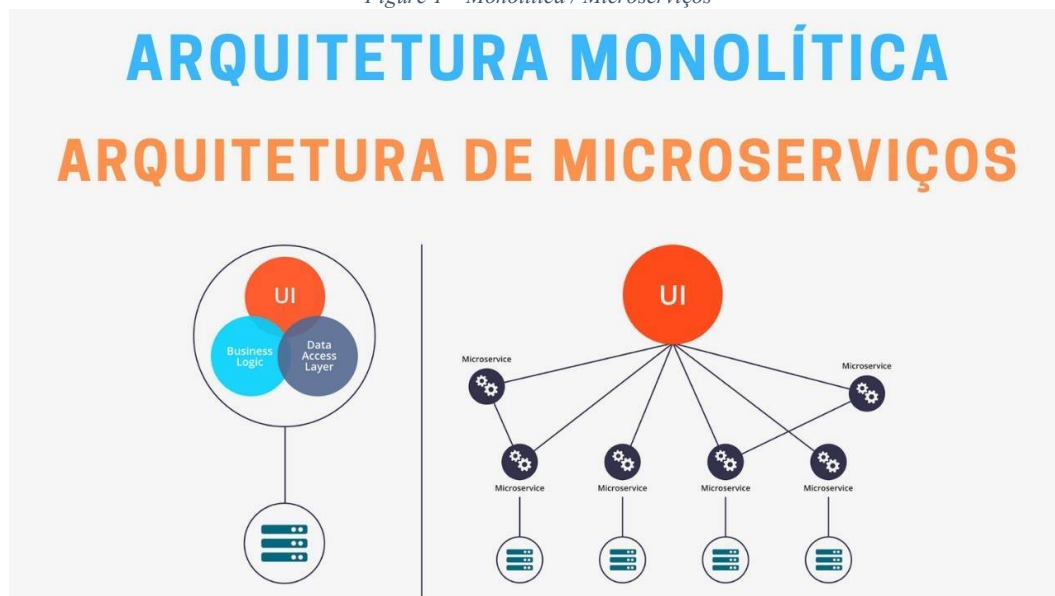
Observamos que todos esses produtos se destinam a oferecer soluções que funcionam em apenas um estacionamento. Nenhum dos possíveis concorrentes apresentou uma solução que abranja vários estacionamentos e que possa ser acessada diretamente pelo usuário, facilitando o uso de estacionamentos no dia a dia, sendo uma funcionalidade gratuita. Dessa forma, nos propomos a criar uma solução que possa preencher essa lacuna no mercado.

MÉTODOS E FERRAMENTAS

Diferentemente da arquitetura monolítica onde a aplicação é formada por uma única unidade constituída por interface de usuário, banco de dados e aplicação no lado do servidor, a arquitetura de microserviços Application Programming Interface (API) RESTful (BIEHL, 2016) é um conjunto de pequenos serviços, onde cada um deles tem uma única responsabilidade e se comunicando por meio de mecanismos leves.

Uma aplicação monolítica utiliza um único banco de dados que suporta toda a aplicação e seus serviços. Enquanto que na arquitetura de microserviços, cada serviço possui seu próprio banco de dados, dividindo cargas de processos e dependendo das necessidades do negócio e dos requisitos da aplicação pode ser relacional ou não relacionado (SQL ou NoSQL). Juntos, esses serviços e seus bancos de dados entregam a experiência do usuário, conforme demonstrado na Figura 1.

Figure 1 – Monolítica / Microserviços



Fonte: Pesquisa Google. Disponível em: <https://www.google.com/search?q=q=imagem+Monolítica+Micro+serviços>

Cada micro serviço é desenvolvido, testado e implantado com independência e autossuficientes possuindo seu próprio banco de dados que não é compartilhado, onde se obtém as vantagens: a) Agilidade de implantação de novas versões; b) Facilidade para trabalhar com versões novas de banco de dados, linguagem de programação e bibliotecas; c) Maior disponibilidade, já que as atualizações são realizadas mais facilmente por serem independentes e em microserviços; d) Flexibilidade para replicação e escalabilidade com segurança; e) Facilidade para o Gerente de projetos/produtos de software compor equipes pequenas para cada micro serviço.

Um Web Service é um sistema de software desenvolvido para suportar interoperabilidade entre máquinas em uma rede. A tecnologia dos Web Services possibilita que diferentes aplicações interajam entre si e sistemas desenvolvidos em plataformas distintas se tornem compatíveis. Os Web Services são componentes que permitem que aplicações enviem e recebam dados em formatos variados. A característica fundamental dos Web Services diz respeito à possibilidade de utilização de diferentes formas de transmissão de dados pela rede. Logo, a arquitetura de Web Services pode trabalhar com protocolos, tais como HTTP, SMTP, FTP, RMI/IIOP ou protocolos de mensagem proprietários (W3C, 2004).

As primeiras versões das especificações de Web services ofereciam apenas o HTTP como meio de transporte de dados (mensagens SOAP XML) e comunicação entre clientes e servidores, sendo ainda o protocolo mais utilizado atualmente (MORO; DORNELES; REBONATTO, 2011).

Para criar um Web Service é necessário conhecer alguns conceitos importantes:

1. O HTTP é um protocolo de comunicação para distribuição de objetos de hipermídia referenciados por uma URL. A transmissão de informações entre um emissor e um receptor caracteriza uma comunicação. É igual a uma comunicação interpessoal, a cordialidade é essencial. O protocolo HTTP estabelece um cabeçalho para suas requisições e respostas. O cabeçalho de uma mensagem são de uso

do cliente e servidor. Através das informações passadas pelo cabeçalho, é possível informar em uma requisição a preferência de idiomas, as codificações e os formatos de conteúdo para uma resposta. O cabeçalho da resposta, por sua vez, contém o código de status, codificação, formato e tempo de expiração do conteúdo (EMER, 2014);

2. A idempotência de um método é relativa às modificações que são realizadas em informações do lado do servidor. Trata-se do efeito que uma mesma requisição tem do lado do servidor, se a mesma requisição, realizada múltiplas vezes, provoca alterações no lado do servidor como se fosse uma única, então esta é considerada idempotente (SAUDATE, 2013);
3. API: Na grande maioria das vezes uma API se comunica com diversos outros códigos interligando diversas funções em um aplicativo. Um exemplo de uma API muito utilizada é a API dos sistemas operacionais que possuem diversos métodos e se comunicam com diversos processos do sistema operacional. Já no contexto de desenvolvimento web, uma API é um conjunto definido de mensagens de requisição e resposta HTTP, geralmente expressado nos formatos XML ou JSON. Ainda que esse termo seja um sinônimo para Web Service, a chamada Web 2.0 está aos poucos depreciando o modelo de serviços SOAP para a técnica REST (MEDEIROS, 2012).
4. HTTPS - O protocolo HTTP, disponibiliza um mecanismo padrão de proteção contra o problema da interceptação dos dados. A solução mais usada para trafegar as informações de maneira segura é o uso do HTTPS (HyperText Transfer Protocol over Secure Sockets Layer), e baseia-se na aplicação de uma camada de segurança com SSL (Secure Socket Layer) sobre o HTTP. Esta camada utiliza certificados digitais para encriptar as comunicações e garantir a autenticidade do servidor e, opcionalmente, do cliente (SAUDATE, 2013).
5. REST (Representational State Transfer) é um estilo de arquitetura para sistemas de hipermídia distribuídos. Esse termo foi utilizado pela primeira vez por Roy Fielding (um dos criadores do protocolo HTTP), em sua tese de doutorado publicada no ano 2000. Assim, é perceptível

que o protocolo REST é guiado pelo que seriam as boas práticas de uso de HTTP (SAUDATE, 2013, p. 26):

- a) Uso adequado dos métodos HTTP;
- b) Uso adequado de URLs;
- c) Uso de códigos de status padronizados para representação de sucessos ou falhas;
- d) Uso adequado de cabeçalhos HTTP;
- e) Interligações entre vários recursos diferentes.

RELATO DO PROCESSO

Iniciamos nosso projeto e definimos que iríamos realizar a parte de backend que é o conjunto de serviços através de uma API interagir com qualquer interface do aplicativo mobile, do painel que informa vagas disponíveis, dos dispositivos que instalados nas vagas que informam o status se ela está ou não ocupada podendo a informação ser consumida ou informada por qualquer interface ou dispositivo que se comunique através da rede utilizando protocolo HTTPS onde se assegura a segurança na comunicação.

Para tanto, usamos a linguagem Python, os bancos de dados foram realizados por Caio Texeira usando PostgreSQL. Foi usado Django REST framework (DRF) que é uma biblioteca Python / Django de código aberto, madura e bem suportada que visa construir APIs web sofisticadas. É um kit de ferramentas flexível e cheio de recursos com arquitetura modular e personalizável que possibilita o desenvolvimento de terminais API simples e prontos para uso e construções REST complicadas.

RESULTADOS

O resultado foi uma API onde podemos de forma mais eficiente obter o tráfego de informações entre os diversos pontos do sistema utilizando conjunto de funções estabelecidas, que funcionam como uma interface intermediária, para a utilização de funcionalidades por aplicações externas. Utilizando o serviço da internet REST, as requisições são feitas à URI de um recurso, e esperam obter uma resposta no formato JSON.

A resposta pode ser uma confirmação de uma alteração no status da vaga ou mesmo uma lista das vagas disponíveis de um estacionamento específico. É utilizado o protocolo HTTPS, com as operações:

- GET (Pegar): essa operação é utilizada para buscar;
- POST (Postar): essa operação é utilizada para criar;
- PUT (Colocar): essa operação é utilizada para alterar;
- DELETE (Deletar): essa operação é utilizada para deletar;

Como, por exemplo, obter (GET) a lista de vagas do estacionamento de código 34 :

Link para acesso ->

```
https://enderecodaAPI/api/vagas/free?estacionamento=34
```


Resposta em formato JSON ->

```
{
  "content": [
    {
      "vagas": {
        "id": 1,
        "estacionamento": 34,
        "status_free": true
      },
      "ocupedTime":
    },
    {
      "vagas": {
        "id": 2,
        "estacionamento": 34,
        "status_free": true
      },
      "ocupedTime":
    },
    {
      "vagas": {
        "id": 3,
        "estacionamento": 34,
        "status_free": true
      },
      "ocupedTime":
    }
  ]
}
```

Onde temos as informações de quais vagas estão livres no estacionamento de código 34 podendo ser consumida por um aplicativo que demonstra de forma gráfica esta informação.

CONSIDERAÇÕES

Durante o desenvolvimento do projeto tivemos a ideia de abordar a tecnologia onde exercesse o nó de comunicação e armazenamento das informações que seria alimentada e consumida por diversas tecnologias que é a API “Application Programming Interface” possibilitando benefícios como a segurança dos dados entre outras, desenvolvida utilizando a linguagem de programação Python com o framework Django e Django Rest. Por meio desse projeto pudemos melhor compreender a implementação com este conjunto de tecnologias, que atualmente são de extrema relevância para a manipulação segura de informações em diversas aplicações.

REFERÊNCIAS

Revista Produção Industrial & Serviços – ISSN 2526-1894 - v. 4 n. 1 (2017): A engenharia de produção na indústria) e (ERNST& YOUNG, 2014.

JSON API. (2013). JSON API - A specification for building APIs in JSON. Disponível em: <http://jsonapi.org/>

SAUDATE, A. REST. Construa API's inteligentes de maneira simples, 1. ed. Casa do Código 2013. 101 p.