

Bruna Do Espirito Santo Souza
Layane Grazielle Sousa Dias
Pedro Ivo Santana Melo

Arquitetura

SmartPark

Trabalho apresentado
na disciplina de Padrões
de Arquitetura de Software

Orientador: Dr. Jacson
Rodrigues Barbosa

Goiânia

1. Introdução	4
1.1 Finalidade	4
1.2 Escopo	4
1.3 Definições, Acrônimos e Abreviações	4
1.4 Visão Geral	4
2. Contexto da Arquitetura	5
2.1 Funcionalidades e Restrições Arquiteturais	5
2.2 Atributos de Qualidades Prioritários	6
2.2.1 Métricas para avaliar os atributos de qualidade	6
2.3 Tecnologias	7
3. Representação da Arquitetura Candidata	7
3.1 Diagrama pub/sub do sistema	8
4. Visão Geral	8
4.1 Descrição	8
4.2 Componentes Principais:	9
4.2.1 Publishers:	9
4.2.2 Subscribers:	9
4.2.3 Tópicos:	9
4.2.4 Gêmeo Digital:	9
5. Decisões arquiteturais	9
5.1 Requisitos Arquiteturalmente Significativos	9
6. Ponto de vista dos Casos de Uso	10
6.1 Descrição	10
6.2 Visão de Casos de Uso	10
7. Ponto de vista do Projetista	11
7.1 Descrição	11
7.2 Visão em Módulos	11
8. Ponto de vista de Segurança	12
8.1 Descrição	12
8.2 Visão de Segurança	12
9. Ponto de vista do Fluxo de Dados	13
9.1 Descrição:	13
9.1 Publicação de Dados:	13
9.2 Assinatura e Processamento:	13
9.3 Atualização do Gêmeo Digital:	14
10. Aspectos de computação ubíqua contemplados	14
10.1 Ciência de contexto:	14
10.2 Continuidade:	14
10.3 Consistência:	14

10.4 Complementariedade:	14
10.5 Descrição do conceito de gêmeo digital será integrado na aplicação/sistema.	15

1. Introdução

1.1 Finalidade

A principal finalidade deste documento é definir os aspectos essenciais da Arquitetura de Software, sendo direcionado aos stakeholders do projeto, possuindo grande foco para os Desenvolvedores e para a Equipe de implantação.

1.2 Escopo

O “SmartPark” é um estacionamento inteligente, que visa facilitar a busca por vagas de estacionamento, economizando tempo e reduzindo o tráfego desnecessário. O sistema fará isso através da identificação da chegada do motorista e da condução do mesmo até o estacionamento mais próximo, além de trazer uma visualização da situação do estacionamento para usuários administradores.

1.3 Definições, Acrônimos e Abreviações

Id.: Identificador.

Software: Conjunto de documentações, guias, metodologias, processos, códigos e ferramentas para a solução de um problema.

Stakeholder: Indivíduo, grupo ou organização que possua interesse no Sistema.

Visão Arquitetural: Produto resultante da interpretação de um Stakeholder do sistema.

Arquitetura de Software: Forma como os componentes são agrupados com o objetivo de construir um software ou sistema.

Ponto de Vista Arquitetural: Produto resultante da execução de uma Visão Arquitetural.

Javascript: Linguagem de programação de alto nível, de propósito geral, interpretada, de sintaxe concisa e clara.

1.4 Visão Geral

De maneira simples, o documento visa descrever a estrutura geral do sistema a ser desenvolvido, incluindo etapas de organização lógica e física, bem como de componentes e suas relações e ainda acerca de possíveis interfaces.

As principais decisões de projeto, restrições e metodologias adotadas para o desenvolvimento do projeto estão descritas e detalhadas. Além disso, vale ressaltar que esse Projeto Arquitetural se trata de um processo contínuo, sendo assim é suscetível a possíveis melhorias futuras.

2. Contexto da Arquitetura

2.1 Funcionalidades e Restrições Arquiteturais

Tipo	Id. do Documento de Requisitos
História de Usuário	HU-01
História de Usuário	HU-02
História de Usuário	HU-03
História de Usuário	HU-04
História de Usuário	HU-05
História de Usuário	HU-06
História de Usuário	HU-07
História de Usuário	HU-08
História de Usuário	HU-09
História de Usuário	HU-10
História de Usuário	HU-11

Requisitos não funcionais

RNF 01 - Usabilidade: o sistema deve ser intuitivo e fácil de usar para os usuários, com uma interface amigável que permita uma experiência simples e eficiente durante a entrada, saída e busca por vagas de estacionamento.

RNF 02 - Escalabilidade: o sistema deve ser capaz de lidar com um aumento no número de usuários, dispositivos e operações sem comprometer o desempenho. Ele deve ser escalável para acomodar um crescimento futuro de demanda sem perda de eficiência

RNF 03 - Flexibilidade: O sistema deve ser flexível o suficiente para se adaptar a diferentes configurações de estacionamento e requisitos específicos de cada local.

RNF 04 - Resistência a falhas: O sistema deve ser robusto e capaz de lidar com falhas de componentes individuais sem impactar significativamente a funcionalidade global. Deve ser projetado para garantir a disponibilidade contínua, mesmo em situações de falha, com mecanismos de recuperação rápidos.

RNF 05 - Segurança: O sistema deve garantir a proteção dos dados do usuário.

2.2 Atributos de Qualidades Prioritários

Para garantir a ciência de contexto o usuário precisa fazer login e permitir acesso em tempo real da sua localização. Com isso em vista, o atributo de qualidade de maior prioridade para o projeto é a **segurança** dos dados fornecidos pelo usuário.

Ainda sob o ponto de vista da qualidade, é necessário incluir na arquitetura elementos que contribuam com a **usabilidade** do sistema, sendo o segundo atributo de maior prioridade no projeto, visando uma experiência de uso mais agradável.

Em terceiro nível de prioridade estão os atributos de **escalabilidade**, sendo capaz de lidar com um aumento no número de usuários, dispositivos e operações sem comprometer o desempenho. E em quarto nível de prioridade o de **resistência a falhas**, garantindo a disponibilidade contínua das informações.

2.2.1 Métricas para avaliar os atributos de qualidade

RNF 01 - Usabilidade:

- Tempo de Aprendizado: avaliar o tempo que um usuário leva para se familiarizar e aprender a usar o sistema.
- Taxa de Erros: número de erros cometidos pelos usuários durante a operação do sistema, indicando a eficiência e facilidade de uso.
- Satisfação do Usuário: pesquisas ou avaliações para medir a satisfação geral dos usuários em relação à usabilidade do sistema.

RNF 02 - Escalabilidade:

- Desempenho sob Carga: tempo de resposta e eficiência quando há aumento significativo na carga de usuários.
- Capacidade de Usuários Concorrentes: número máximo de usuários que podem interagir simultaneamente com o sistema sem degradação significativa no desempenho.

RNF 04 - Resistência a falhas:

- Tempo de Recuperação: tempo necessário para o sistema se recuperar de uma falha e voltar ao seu estado operacional normal.
- Taxa de Disponibilidade: calcular o tempo total que o sistema está disponível em relação ao tempo total.

RNF 05 - Segurança:

- Taxa de Incidentes de Segurança: número de incidentes de segurança relatados em um determinado período.
- Controle de Acesso: medir a eficácia dos mecanismos de controle de acesso para prevenir acessos não autorizados.

2.3 Tecnologias

Linguagem de programação back-end: Java

Linguagens para o front-end: HTML, CSS e Javascript

Ambiente de execução: Aplicativo e navegador web

Frameworks: Bootstrap e JQuery

Broker: Cliente MQTT Móvel

Autenticação: Firebase

Nuvem: AWS (Servidor EC2)

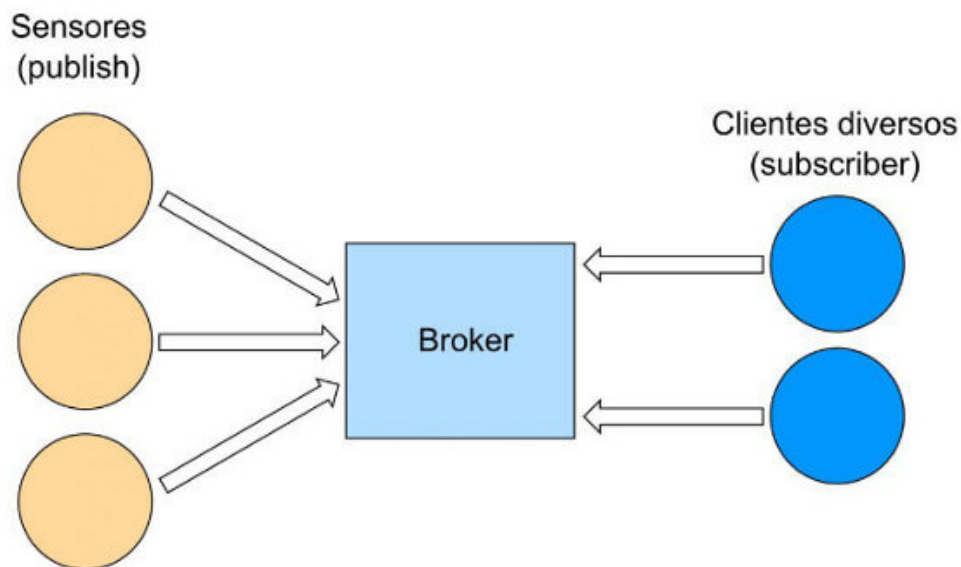
Hospedagem para controle de versão: Github

Sensores: Proximidade e GPS de celulares Android

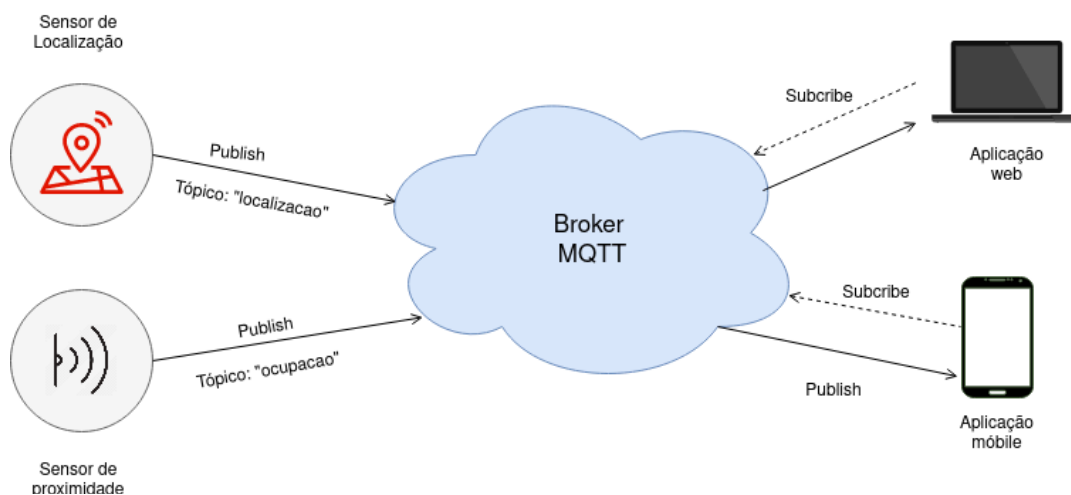
3. Representação da Arquitetura Candidata

O padrão escolhido para implementação do software SmartPark, foi o pub/sub usando um Broker MQTT. Esse é um modelo assíncrono de comunicação onde os participantes são divididos em publicadores, responsáveis por enviar mensagens para canais centrais, e assinantes, que indicam interesse em receber mensagens de canais específicos. Essa abordagem promove o desacoplamento entre os componentes do sistema, permitindo escalabilidade, flexibilidade e resiliência a falhas.

3.1 Diagrama pub/sub do sistema



4. Visão Geral



4.1 Descrição

Na visão geral é criada uma representação visual que fornece uma visão geral de um sistema ou processo. Ele mostra as principais entidades envolvidas e suas interações de forma simplificada.

Na visão geral da distribuição do sistema, temos 2 tipos de sensores, que serão os publicadores nos tópicos do broker MQTT e 2 tipos de clientes, que serão assinantes dos tópicos no servidor EC2 da AWS.

4.2 Componentes Principais:

4.2.1 Publishers:

Sensores de Proximidade: Publicam informações sobre a ocupação das vagas em tempo real.

Sensores de Localização: Publicam dados sobre a localização de dispositivos móveis, indicando a presença de usuários no estacionamento

4.2.2 Subscribers:

Servidor Central: Atua como um assinante para os dados publicados pelos sensores. É responsável por processar e distribuir as informações relevantes para outros componentes do sistema.

4.2.3 Tópicos:

Tópico de Proximidade: Onde os sensores de proximidade publicam o estado em tempo real das vagas no estacionamento.

Tópico de Localização: Onde os sensores de ocupação de vagas publicam informações sobre a presença de dispositivos móveis.

4.2.4 Gêmeo Digital:

Atualização por Subscrição: O gêmeo digital pode ser atualizado por subscrição aos tópicos relevantes, mantendo-se sincronizado com os dados do mundo real.

5. Decisões arquiteturais

5.1 Requisitos Arquiteturalmente Significativos

ID	RAS	Descrição do Cenário
----	-----	----------------------

01	Segurança	O sistema deve ser fácil de usar e entender
02	Usabilidade	O sistema deve ser fácil de usar e entender.
03	Escalabilidade	O sistema deve ser capaz de lidar com um aumento no número de usuários, dispositivos e operações sem comprometer o desempenho.
04	Resistência a falhas	O sistema deve garantir a disponibilidade contínua das informações

6. Ponto de vista dos Casos de Uso

6.1 Descrição

Na análise de requisitos, é criada uma visão de casos de uso para fornecer uma base para o planejamento da arquitetura e outros artefatos. Essa visão representa os casos de uso e cenários que o usuário terá, além de classes e riscos técnicos relevantes para a arquitetura do sistema. A visão de casos de uso é considerada em cada iteração do ciclo de vida do software.

6.2 Visão de Casos de Uso

Os requisitos funcionais foram utilizados para compor os casos de uso, resumindo as principais funcionalidades do sistema nos diagramas abaixo:

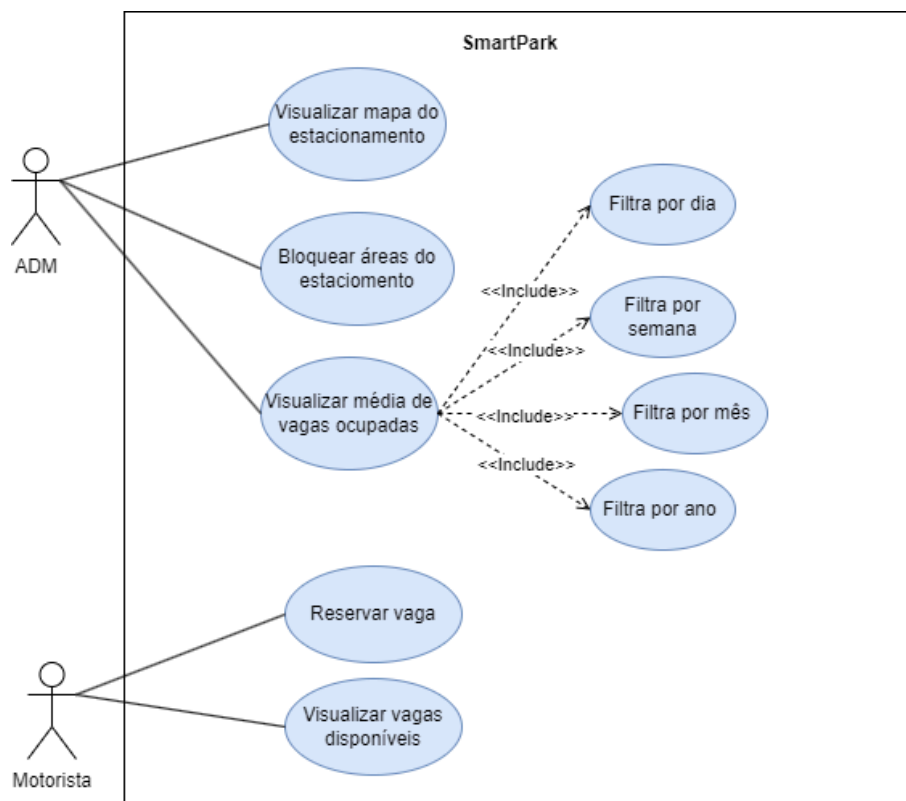


Figura 3: Casos de uso

7. Ponto de vista do Projetista

7.1 Descrição

O ponto de vista de projetista é um dos pontos de vista arquiteturais que se concentra nas decisões de projeto e nas preocupações técnicas específicas relacionadas à implementação do sistema. Ele fornece informações detalhadas e orientações para os projetistas e desenvolvedores envolvidos na construção do sistema.

7.2 Visão em Módulos

- **Módulo de Autenticação/Registro:**
 - Responsável pelo cadastro e autenticação dos usuários no sistema.
 - Funcionalidades: registro de novos usuários, login, recuperação de senha, gerenciamento de sessões.

- **Módulo das Vagas do estacionamento:**
 - Responsável por atualizar o status das vagas do estacionamento.
- **Módulo de Direcionamento a vaga disponível:**
 - Responsável por direcionar o usuário para a vaga de estacionamento disponível mais próximo

Estrutura Hierárquica:

1. Módulo de Autenticação/Registro
2. Módulo de Vagas do estacionamento
3. Módulo de Direcionamento à vaga disponível

8. Ponto de vista de Segurança

8.1 Descrição

É uma perspectiva específica dentro da arquitetura de um sistema que foca nos aspectos relacionados à proteção, privacidade e segurança das informações e dos recursos do sistema. Essa visão tem como objetivo identificar os requisitos de segurança, estabelecer as estratégias e controles de segurança adequados, e definir as medidas e práticas que devem ser implementadas para garantir a integridade, disponibilidade e confidencialidade do sistema.

8.2 Visão de Segurança

- **Autenticação e Controle de Acesso:**
 - Implementar um sistema de autenticação robusto para garantir que apenas usuários autenticados tenham acesso ao sistema.
 - Utilizar técnicas de controle de acesso para definir permissões e restrições de acordo com o papel ou perfil do usuário.
- **Proteção de Dados:**
 - Utilizar técnicas de criptografia para proteger dados confidenciais, como senhas de usuários e informações pessoais.

- Garantir que as informações sensíveis sejam armazenadas de forma segura e acessíveis apenas para usuários autorizados.
- **Prevenção de Injeção de Código:**
 - Implementar mecanismos de validação e sanitização de dados para evitar ataques de injeção de código, como SQL injection e XSS (Cross-Site Scripting).
- **Proteção contra Ataques de Força Bruta:**
 - Implementar mecanismos de segurança que limitem o número de tentativas de login, como bloqueio de contas após várias tentativas falhas.
- **Proteção de APIs e Integrações:**
 - Utilizar autenticação e autorização adequadas para proteger as APIs utilizadas na integração com os serviços de streaming.
 - Validar e filtrar dados recebidos e enviados pelas APIs, evitando a exposição de informações sensíveis.

9. Ponto de vista do Fluxo de Dados

9.1 Descrição:

O ponto de vista do fluxo de dados oferece uma visão holística da movimentação de informações dentro do sistema. Este fluxo é crucial para entender como os dados são capturados, processados e distribuídos.

9.1 Publicação de Dados:

A entrada de dados será por sensores de proximidade e localização publicam dados nos tópicos correspondentes sempre que houver uma alteração relevante.

9.2 Assinatura e Processamento:

O servidor central atua como assinante, recebendo e processando dados dos sensores. Ele pode usar algoritmos para otimizar a alocação de vagas e atualizar o gêmeo digital.

9.3 Atualização do Gêmeo Digital:

O gêmeo digital, por meio de um mecanismo de subscrição, é atualizado automaticamente com as mudanças nos tópicos de proximidade e localização, refletindo o estado atual do estacionamento.

10. Aspectos de computação ubíqua contemplados

10.1 Ciência de contexto:

Utilizando a localização do usuário e comparando com a entrada do estacionamento, quando essa localização for a mesma iniciar a rota da vaga mais próxima.

10.2 Continuidade:

Com a adaptabilidade das informações de status da vaga, para *smartphones* uma visão em lista e para *desktop* um mapa.

10.3 Consistência:

Com uma UI e experiência do usuário que é consistente entre as versões *mobile* e *desktop*.

10.4 Complementariedade:

Com múltiplos dispositivos de interface que se complementam - versão *mobile* do administrador com uma lista das vagas do estacionamento e a versão *desktop* com um mapa do estacionamento e dashboard

10.5 Descrição do conceito de gêmeo digital será integrado na aplicação/sistema.

O gêmeo digital representará o estacionamento físico, dando uma visualização do estacionamento, das vagas disponíveis e seus status ao usuário cliente e admin. Ele também irá indicar o melhor caminho para a vaga disponível mais próxima. Além disso, o gêmeo digital irá colher os dados sobre seu gêmeo físico, como: tempo de permanência, ocupação durante o dia, meses com mais movimento, etc.