|  |  |
| --- | --- |
| CURSO DE SISTEMAS DE INFORMAÇÃO – TCC ACADÊMICO | |
| ( ) PRÉ-PROJETO  (X) PROJETO | ANO/SEMESTRE: 2022/2 |

PORTARIA FÁCIL: UMA SOLUÇÃO SEGURA COM ARQUITETURA SERVERLESS PARA CONTROLE DE ACESSO

Jéssica Maria Lana de Melo Kohn

Prof. Simone Erbs da Costa – Orientadora

# Introdução

Com o decorrer dos anos e com a progressão da tecnologia, surgindo a cada dia alguma novidade ou tendência, os sistemas de controle de acesso vêm sendo fortemente utilizados, uma vez que a segurança é algo de extrema relevância nos últimos anos (AMARAL, 2020). De acordo com Oliveira *et al.* (2020), é cada vez mais importante monitorar a entrada e saída de pessoas dos ambientes, devido que isso auxilia a garantir a segurança das pessoas, o sigilo das informações e a integridade dos bens. Rovani (2019) complementa que o controle deve ser feito por meio de equipamentos eletrônicos, clausuras e fechaduras, nas quais o usuário deve estar cadastrado para acessar o ambiente.

Esses sistemas são projetados para atender os requisitos de segurança patrimonial (ROVANI, 2019). Nesse sentido, Silva (2018) coloca que os controles de acesso são geralmente agrupados em três tipos de controle: físico, lógico e administrativo. As empresas e instituições necessitam desses três tipos de controles. Silva (2018) ainda observa que as políticas de segurança das empresas e instituições, por meio da documentação dos padrões de segurança governam o uso desses controles. Já segundo Mariano e Montanha (2019), as empresas estão zelando cada vez mais pela segurança, contudo, o índice de violência sobe proporcionalmente com os investimentos em segurança. Por esse motivo para Mariano e Montanha (2019), o investimento no segmento precisa ser feito de maneira eficaz, ajustando pontos de falha na segurança.

Neste contexto, sistemas de segurança precisam estar disponível, íntegros e operacionais a todo o momento para garantir a segurança proposta. Para Satheler (2021), a alta disponibilidade e escalabilidade têm como objetivo garantir estas premissas. Satheler (2021) ainda coloca que a alta disponibilidade é uma qualidade de infraestrutura de computação, que permite que uma aplicação continue funcionando, mesmo quando alguns de seus componentes falham. Isso é um ponto crucial principalmente para sistemas críticos que não podem tolerar a interrupção do serviço, em que qualquer tempo de inatividade pode causar prejuízos, danos ambientais e até perda da vida humana (SATHELER, 2021).

Para Cloudflare (2022), a arquitetura *serverless* oferece uma série de benefícios aos desenvolvedores Web, incluindo disponibilidade e escalabilidade, tempo de colocação no mercado mais rápido e despesas menores. Diante do que foi apresentado, este trabalho propõe o desenvolvimento de uma aplicação Web e móvel com uma arquitetura *serverless*. Conjectura-se assim realizar o controle de acesso de pessoas para empresas e condomínios de forma escalável e integra, garantindo o acesso a todos mesmo sobre alta demanda.

## OBJETIVOS

O objetivo geral deste trabalho é desenvolver uma aplicação de controle de acesso escalável e integra com arquitetura *serverless*. Os objetivos específicos são:

1. controlar e agilizar o acesso de pessoas por meio de interfaces disponibilizadas;
2. garantir alta disponibilidade e escalabilidade com o uso de recursos e produtos da Amazon Web Services (AWS);
3. analisar e avaliar a usabilidade, a comunicabilidade e a experiência de usuário das interfaces desenvolvidas e de suas funcionalidades, por meio do Método Relationship of M3C with User Requirements and Usability and Communicability Assessment in groupware (RURUCAg).

# trabalhos correlatos

Nesta seção estão descritos três trabalhos correlatos que apresentam características semelhantes ao trabalho proposto. A subseção 2.1 traz uma aplicação para controle de acessos para gestão de pessoas em ambientes amplos e que foi desenvolvido com a tecnologia por Radio Frequency IDentification (RFID) (MORAIS; SILVA; DÁGOLA, 2018). A subseção 2.2 descreve uma aplicação que controla a entrada e saída de automóveis e motocicletas, também usando a tecnologia de identificação por RFID (SILVA, 2018). Por fim, a subseção 2.3 apresenta uma aplicação com funções de controle de acesso entre locador e locatário a ambientes físicos compartilhados fazendo o uso da tecnologia *blockchain* por meio de uma fechadura eletrônica (ZAPAROLI, 2019).

## SISTEMA DE CONTROLE DE ACESSO POR RFID

Morais, Silva e Dágola (2018) propuseram uma solução Web de controle de acesso para gestão de pessoas em ambientes amplos. A aplicação utiliza a tecnologia de RFID, desenvolvida para uso dos seguranças monitorarem a entrada e saída de alunos ou visitantes. Ao utilizar a tecnologia, o usuário recebe um cartão que tem a funcionalidade de uma carteirinha e como uma “chave” de acesso, registrar sua entrada. Algumas das principais funcionalidades de Morais, Silva e Dágola (2018) são: gerenciamento com interface; manter cadastro do usuário; manter registro de data e hora do evento de entrada e saída; filtrar dados; gerar relatórios; e ser disponibilizado na plataforma Web.

A interface da aplicação contém informações básicas para o usuário visualizar os dados que foram inseridos, como exemplo: o nome do usuário, tipo de usuário (aluno, visitante ou funcionário), horário do acesso, tipos de acesso (se está entrando no Instituto ou saindo) e a foto do usuário. Essa aplicação de Morais, Silva e Dágola (2018) contém a interface gráfica e funciona em um computador que está montado na portaria de entrada do Instituto. Nos cartões RFID destinados aos alunos, será gravada a matrícula de cada aluno, enquanto para os visitantes, será gravado o Cadastro de Pessoas Físicas (CPF), possibilitando a identificação da pessoa que acessou as dependências do Instituto (MORAIS; SILVA; DÁGOLA, 2018).

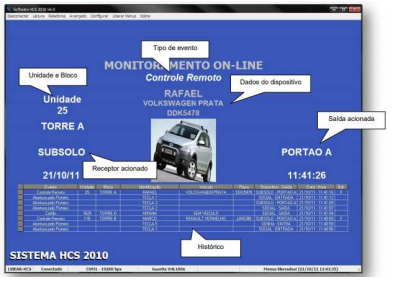
Segundo Morais, Silva e Dágola (2018), a aplicação cumpre a função de controlar e gerar informações sobre a dinâmica de entrada e saída dos alunos. Os dados até a atual etapa de andamento do projeto são consultados, somente, sob demanda. Ou seja, os responsáveis, a direção de ensino ou algum fato relevante que tenha ocorrido e necessite de informações de acesso do discente para tratamento da demanda e nesse momento, os dados são analisados. Dessa forma, o projeto ainda não conta com os dados analisados para geração de resultados que inclua todos os alunos para a descrição de padrões de comportamento do corpo discente (MORAIS; SILVA; DÁGOLA, 2018).

## SISTEMA DE CONTROLE DE ACESSO PARA ESTACIONAMENTO

Silva (2018) desenvolveu uma aplicação que controla a entrada e saída de automóveis, bem como motocicletas, por meio da tecnologia de identificação RFID. A aplicação utiliza cartões como chave de identificação e todos os eventos são registrados. Os eventos são armazenados na memória interna da aplicação, em aproximadamente 8.100 posições. Sempre que este limite é atingido, o equipamento automaticamente cria uma cópia de toda memória em um cartão Secure Digital (SD) devidamente instalado, passando então a sobrescrever os eventos mais antigos na memória interna. Se o cartão SD não estiver presente, apenas a memória interna será gerenciada. E por fim, será solicitado ao usuário se deseja visualizar o relatório dos dispositivos (SILVA, 2018). Essa aplicação utiliza equipamentos da empresa Linear HCS que tem o seu ramo de atuação no campo da segurança eletrônica (SILVA, 2018).

Cabe destacar, que algumas das principais funcionalidades desenvolvidas por Silva (2018) são: gerenciamento com interface; manter cadastro do usuário; manter cadastro de dispositivos; manter registro de data e hora do evento de entrada e saída; filtrar dados; gerar relatórios; e ser disponibilizado na plataforma Web. A tela de cadastramento dos usuários dispõe de campos distintos para identificação, sendo 18 caracteres para nome, seleção de 32 marcas (pré-definidas) de fabricantes de veículos, 16 cores e sete caracteres alfanuméricos para placa. O módulo de guarita dispõe de diversos recursos para auxiliar na segurança do patrimônio e dos usuários do sistema, como por exemplo: pânico de usuário, que pode ser disparado por meio de cartões RFID, controles remotos e tags tácteis, desperta porteiro, veículo carona, pânico entre condomínios e clonagem. A tela de monitoramento on-line, mostrado na Figura 1, apresenta o monitoramento da aplicação e exibe em tempo real qualquer acionamento ocorrido ao equipamento, registrando ainda um histórico com os oito eventos anteriores (SILVA, 2018).

Figura – Tela de Monitoramento On-line



Fonte: Silva (2018).

## SMARTLOCK: Controle de acesso através de Smart Contracts e Smart Property

Esse trabalho propõe o desenvolvimento de uma fechadura eletrônica controlada por contratos inteligentes registrados na plataforma *blockchain* da Ethereum. A aplicação abrange tanto hardware quanto software e é composta por três componentes: uma fechadura eletrônica, um servidor hospedado na nuvem e uma aplicação para celular. O hóspede se cadastra em uma plataforma de reservas, reserva a propriedade desejada e essa reserva quando aprovada pelo proprietário tem suas informações armazenadas em um contrato inteligente implantado na rede *blockchain* da Ethereum. Para acessar a propriedade, o hóspede utiliza a aplicação em seu celular para transmitir por som o código de credencial específico da reserva que é captado pela fechadura. A fechadura verifica as informações armazenadas no contrato inteligente e se forem válidas o acesso a propriedade é liberado. As seguintes características foram avaliadas: comodidade e segurança tanto para o hóspede quanto para o proprietário trazidos ao utilizar dispositivos móveis para destravar fechaduras eletrônicas agregando a segurança proporcionada pela plataforma *blockchain* da Ethereum. Já para verificar a viabilidade da solução, foram realizados testes com supostos usuários da aplicação que responderam um questionário. Após análise do questionário, constatou-se que o projeto é viável, mais atrativo que os modelos de contrato tradicionais, funcional e proporcionou conforto (ZAPAROLI, 2019).

Para Zaparoli (2019), a aplicação possui funcionalidades como: gerenciamento com interface; manter cadastro do usuário; manter registro de data e hora do evento de entrada e saída; e ser disponibilizada na plataforma Android. A aplicação possui três telas. Uma tela de *login*, uma tela de perfil do usuário com um botão para *logout* e outra tela na qual as reservas são listadas. Cada tela é atribuída a uma atividade *Activity* que é o componente que descreve o seu visual com o qual o usuário vai interagir. A especificação dos componentes visuais e como eles vão ser arranjados é definida em um arquivo eXtensible Markup Language (XML) para cada uma das telas (ZAPAROLI, 2019).

A tela inicial apresentada para o usuário é a tela de *login* e pode ser visualizada na Figura 2 (a). Nela existem dois campos do tipo *EditText*, que é um elemento de interface de usuário para inserir e modificar texto. Um deles é destinado para inserir o e-mail do usuário e o outro para inserir a sua senha. Logo abaixo deles há um botão com o texto LOG IN*.* Na tela de perfil, foi adicionado um botão para realizar o LOGOUT da aplicação. Ela pode ser visualizada na Figura 2 (b). Ao selecionar esse botão é executada a função de *logout* e após a sua execução o token é revogado e não tem mais validade para fazer novas requisições. Durante a criação da tela das reservas, Figura 2 (c), três operações importantes são realizadas. Primeiro, é configurado o protocolo Chirp.io. Segundo, é instanciada um objeto que será usado para gerar a credencial de acesso com as informações obtidas de cada uma das reservas usando o algoritmo de *hashing* do tipo SHA256. Terceiro, uma função é executada para obter as informações das reservas e listá-las. As informações das reservas são recebidas no formato JavaScript Object Notation (JSON) (ZAPAROLI, 2019).

Figura – Telas de (a) *login*, (b) perfil e (c) de reservas

Interface gráfica do usuário, Aplicativo, Tabela, Site

Descrição gerada automaticamente

Fonte: Zaparoli (2019).

# proposta DA APLICAÇÃO

Nesta seção serão descritas as justificativas para o desenvolvimento do trabalho proposto (subseção 3.1), também serão descritos os Requisitos Funcionais (RF) e os Requisitos Não Funcionais (RNF) (subseção 3.2). Para finalizar será descrito as metodologias e planejamento do cronograma para o desenvolvimento do trabalho aqui proposto (subseção 3.3).

## JUSTIFICATIVA

Segundo Amaral (2020), a portaria é a principal forma de acesso a locais ou propriedades, sendo de extrema importância a existência de um controle de entrada e saída de pessoas. Oliveira *et al.* (2020) complementam que é possível assegurar a entrada fácil e simplificada de pessoas autorizadas, bloquear os não autorizados, gerenciar diferentes níveis de acesso nos ambientes, garantir a segurança e integridade de bens e informações sigilosas. Nesse contexto, Morais, Silva e Dágola (2018), Silva (2018) e Zaparoli (2019) identificaram a necessidade de desenvolver uma aplicação com elementos e tecnologias para controlar e agilizar o acesso de pessoas, bloquear a entrada de não autorizados, promover a segurança de dados, seus bens e pessoas. No Quadro 1 é apresentado um comparativo entre os trabalhos correlatos descritos na seção 2, de modo que as linhas representam as características e as colunas os trabalhos relacionados.

Quadro – Comparativo dos trabalhos correlatos

|  |  |  |  |
| --- | --- | --- | --- |
| **Trabalhos Correlatos**  **Características** | **Morais, Silva e Dágola (2018)** | **Silva (2018)** | **Zaparoli** **(2019)** |
| Gerenciamento com interface ao usuário | ✓ | ✓ | ✓ |
| Manter cadastro dispositivos | X | ✓ | X |
| Manter cadastro do usuário | ✓ | ✓ | ✓ |
| Manter registro de data e hora do evento de entrada e saída | ✓ | ✓ | ✓ |
| Possuir filtros | ✓ | ✓ | ✓ |
| Gerar relatórios | ✓ | ✓ | X |
| Plataforma | Web | Web | Android |

Fonte: elaborado pela autora.

Ao analisar o Quadro 1 é possível identificar que as soluções de Morais, Silva e Dágola (2018), Silva (2018) e Zaparoli (2019) apresentam o gerenciamento por meio da sua interface. Em Silva (2018) é possível realizar o cadastro dos dispositivos ou manter uma leitura salva podendo visualizar os tipos na tela de gerenciador de dispositivos. A características de manter o cadastro de usuário é de suma importância devido ao gerenciamento ser feito por mais de um porteiro, estando presente em Morais, Silva e Dágola (2018), Silva (2018) e Zaparoli (2019). Já as características de manter um cadastro de data e hora dos eventos de entrada e saída, assim como possuir filtros também estão disponíveis nas aplicações de Morais, Silva e Dágola (2018), Silva (2018) e Zaparoli (2019). Essas são as principais características para controlar as entradas e saídas, pois elas permitem monitorar e gravar dia e horário que foi realizada a passagem pela portaria.

Morais, Silva e Dágola (2018) e Silva (2018) destacam-se pela geração de relatórios, cabe destacar que Morais, Silva e Dágola (2018) e Silva (2018) disponibilizaram a aplicação na plataforma Web, enquanto Zaparoli (2019) disponibilizou sua solução para plataforma Android, no qual a presente proposta aponta paridade com os três trabalhos correlatos apresentados, enfatizando o controle de acesso, o gerenciamento das pessoas que desejam acessar o locais controlados, o bloqueio de alguém não autorizado, a entrada facilitada de quem tem horário de entrada prevista. Além disso, a proposta se destaca por meio da alta disponibilidade, permitindo o controle de acesso a qualquer dia e horário de forma integra e operacional.

Com essas características e tal como as apresentadas no Quadro 1, sistemas que controlam acesso e segurança possuem uma grande relevância para a sociedade, pois com esta premissa de segurança é possível garantir a integridade física de pessoas e bens. Juntamente com as evoluções tecnologias é possível fornecer tal segurança de forma simples e pratica utilizando recursos da AWS, como Lambdas, para desenvolvimento de serviços Representational State Transfer (REST) de alta performance com Serverless *framework*, PostgreSQL para armazenamento de forma relacional e Ionic para o desenvolvimento de *front-ends* híbridos de aplicações Web e Android. A proposta trará ainda como contribuição acadêmica, o uso do método Relationship of M3C with User Requirements and Usability and Communicability Assessment in groupware (RURUCAg) na área da segurança, que poderá ser utilizado em trabalhos futuros.

## REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

Nessa subseção serão especificados os Requisitos Funcionais (RF) e Requisitos Não Funcionais (RNF), conforme o Quadro 2.

Quadro – Principais Requisitos Funcionais e Não Funcionais

|  |  |
| --- | --- |
| **O aplicativo deve:** | **Tipo** |
| permitir ao usuário realizar *login*/*logout* | RF |
| permitir ao usuário manter usuários do tipo porteiro (Create, Read, Update and Delete – CRUD) | RF |
| permitir ao usuário manter um cadastro de portarias (CRUD) | RF |
| permitir ao usuário manter um cadastro de pessoas (CRUD) | RF |
| permitir ao usuário manter um cadastro de tipo de visitas (CRUD) | RF |
| Permitir ao usuário manter um registro de data e hora na entrada e saída das pessoas (CRUD) | RF |
| permitir ao usuário manter agendamento de visitas (CRUD) | RF |
| permitir ao usuário visualizar quantidades de visitas (dia, mês, ano) | RF |
| ser construído utilizando um *framework* Ionic para o *front-end* | RNF |
| ser construído utilizando Lambdas com Serverless *framework* para o *back-end* | RNF |
| ser construído utilizando banco de dados PostgreSQL | RNF |
| utilizar o método RURUCAg para modelar a relação dos requisitos com as heurísticas de Nielsen | RNF |
| utilizar o método RURUCAg para avaliar a usabilidade e a experiência de uso | RNF |
| ser construído com base nos padrões do material design | RNF |
| ser de alta disponibilidade e escalabilidade | RNF |
| ser construído utilizando os recursos e produtos da AWS | RNF |
| ser disponibilizado na nuvem | RNF |
| ser feita a identificação das pessoas por meio do documento | RNF |
| ser feita a identificação das pessoas por meio do nome | RNF |
| possuir acesso a câmera | RNF |
| possuir acesso a internet | RNF |

Fonte: elaborado pela autora.

## METODOLOGIA

A metodologia dessa proposta será constituída pelos seguintes instrumentos metodológicos e será desenvolvido nas etapas relacionadas no Quadro 3:

1. aprofundamento bibliográfico: realizar aprofundamento na literatura sobre os assuntos como: a AWS Lambda, *serverless*, Serverless *framework*, Ionic, PostgreSQL, escalabilidade e disponibilidade;
2. levantamento dos requisitos: analisar os requisitos funcionais e não-funcionais já definidos e, se necessário, especificar outros a partir da etapa do aprofundamento realizado;
3. especificação e análise: formalizar as funcionalidades da aplicação por meio da construção de casos de uso e diagramas da Unified Modeling Language (UML), utilizando a ferramenta Draw.io;
4. implementação: desenvolver a aplicação utilizando no *front-end* Ionic e na Application Programming Interface (API) REST que será publicada utilizando recursos e produtos AWS como Lambdas com Serverless *framework* em conjunto com um banco de dados PostgreSQL;
5. verificação, validação e análise: paralelamente a implementação, será realizado testes da aplicação como funcionalidades, performance, usabilidade e experiência do usuário das interfaces desenvolvidas por meio do método RURUCAg.

Quadro – Cronograma

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Quinzenas**  **Etapas** | **2023** | | | | | | | | | |
| **fev.** | | **mar.** | | **abr.** | | **maio.** | | **jun.** | |
| **1** | **2** | **1** | **2** | **1** | **2** | **1** | **2** | **1** | **2** |
| aprofundamento bibliográfico |  |  |  |  |  |  |  |  |  |  |
| levantamento dos requisitos |  |  |  |  |  |  |  |  |  |  |
| especificação e análise |  |  |  |  |  |  |  |  |  |  |
| implementação |  |  |  |  |  |  |  |  |  |  |
| verificação, validação e análise |  |  |  |  |  |  |  |  |  |  |

Fonte: elaborado pela autora.

# REVISÃO BIBLIOGRÁFICA

Nesta seção os conceitos de maior relevância para o trabalho serão descritos e ela está organizada da seguinte forma: a subseção 4.1 apresenta o conceito de acesso e segurança; a subseção 4.2 contextualiza o tema sobre recursos e produtos AWS; e por fim, a subseção 4.3 aborda a arquitetura *serverless*.

## ACESSO E SEGURANÇA

Para Amaral (2020), o sistema de segurança é o cérebro do controle de acesso, adquire informações para o bloqueio físico, usa políticas e regras de segurança para processar as informações e retorna ao bloqueio físico para permitir ou proibir o acesso. Nesse sentido, Oliveira *et al.* (2020) colocam que o controle de acesso é qualquer sistema, mecanismo ou equipamento que limite o acesso a um ambiente ou informação, garantindo a segurança de dados, bens e pessoas.

Este conceito segundo Mariano e Montanha (2019), é baseado em aplicar procedimentos e/ou equipamentos com o objetivo de administrar o acesso físico de pessoas, veículos e materiais a um determinado perímetro de segurança ou área delimitada. Isso trará à organização uma segurança maior tanto na segurança física, com equipamentos e pessoal, como também a segurança das informações contidas na empresa (MARIANO; MONTANHA, 2019).

Rovani (2019) comenta que na maioria dos condomínios, o local mais vulnerável é o hall de entrada, pois os moradores na maioria das vezes esquecem de fechar portas, ativar o alarme e perdem suas chaves. Isso acontece, porque eles não possuem sistemas que automatizem estas funções, colocando todos os moradores em risco (ROVANI, 2019). Para ter um melhor controle, Souza (2010) complementa que é necessário estabelecer perímetros de controle, com acesso isolado, partindo de partes mais externas para as mais internas, focando tanto a entrada dos indivíduos quanto a saída. Os perímetros, normalmente são delimitados e controlados por barreiras físicas e tecnologias de detecção (SOUZA, 2010).

Sobre serviços de segurança para Oliveira (2004), são como barreiras físicas, equipamentos e recursos humanos para a defesa do patrimônio e dos interesses de proprietários, seguindo as normas e procedimentos por eles estabelecidos e dentro dos limites de seu direito de propriedade. Um sistema de segurança privadamente organizado tem por funcionalidade restringir, controlar e monitorar acessos a pessoas e patrimônios (OLIVEIRA, 2004).

Pinheiro (2008) complementa que sistemas de controle de acesso físico tem como objetivo permitir que somente usuários autorizados tenham acesso aos seus respectivos ambientes, impedindo os não autorizados e visam automatizar o processo de verificação de acesso ou auxiliar nas tarefas relativas à proteção patrimonial. Para serem usados para autenticação, precisam de uma base de dados contendo informações de identificação e, para o nível de permissão, informações do que o usuário pode fazer (PINHEIRO, 2008).

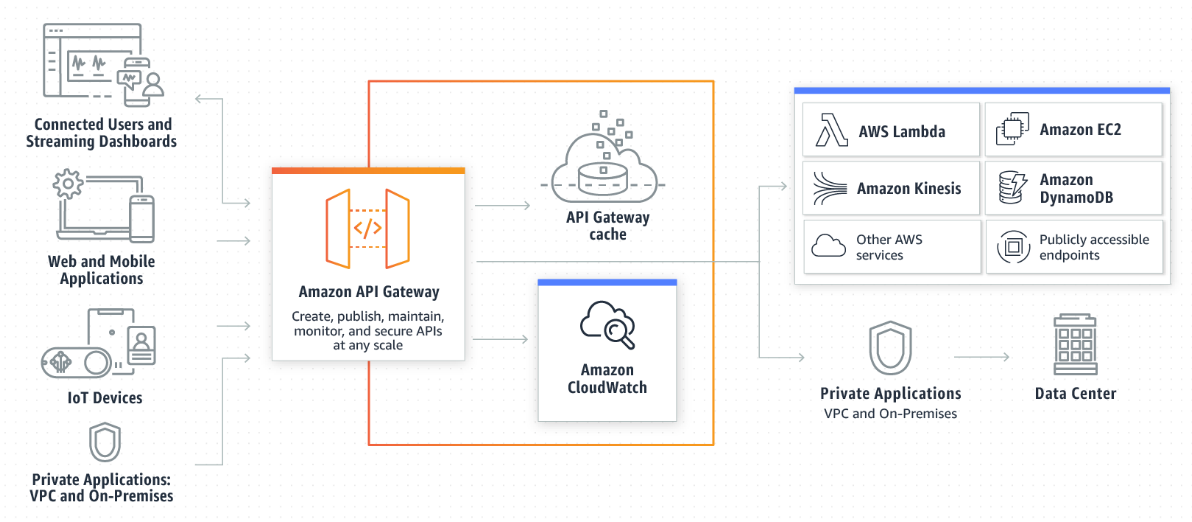
## RECURSOS E PRODUTOS AWS

A AWS Lambda é um serviço de computação *serverless* que executa em uma infraestrutura de computação de alta disponibilidade, incluindo a administração dos recursos computacionais, manutenção do servidor, do sistema operacional, provisionamento e escalabilidade automática da capacidade e registro em log do código (AMAZON, 2022b). Nesse ambiente, o código é organizado em funções, segundo Amazon (2022b), executa apenas quando necessário e se escala automaticamente. Isso ocorre quando existe algumas solicitações por dia a milhares por segundo. Seu custo será apenas pelo tempo de computação que consome, não existe cobrança quando o código não está em execução (AMAZON, 2022b).

Com isso, para essas aplicações com dimensionamento automático como a AWS Lambda, Serverless (2022) coloca que existe alguns serviços AWS, como o Serverless *framework*, que se destaca por ser uma ferramenta de desenvolvimento, Command-Line Interface (CLI) de código aberto. Com a possibilidade de habilitar métricas, logs e alertas em tempo real, consultar todas as invocações de uma Lambda, as solicitações do API Gateway e detalhar problemas específicos (SERVERLESS, 2022).

Amazon (2022c) ainda destaca que o suporte que o API Gateway oferece para arquiteturas *serverless* é um serviço gerenciado que permite que desenvolvedores criem, publiquem, mantenham, monitorem e protejam APIs em qualquer escala com facilidade. APIs agem como a “porta de entrada” para aplicativos acessarem dados, lógica de negócios ou funcionalidade do seu *back-end* (AMAZON, 2022c). Na Figura 3 é possível observar algumas das funcionalidades do API Gateway, como o processamento de centenas de milhares de chamadas de APIs simultâneas. Além disso, pela referida figura é possível verificar o gerenciamento de tráfego, suporte de Cross-Origin Resource Sharing (CORS), controle de autorização e acesso, com fluxo controlado, monitoramento e gerenciamento de versões de API (AMAZON, 2022c).

Figura – As funcionalidades do API Gateway



Fonte: Amazon (2022c).

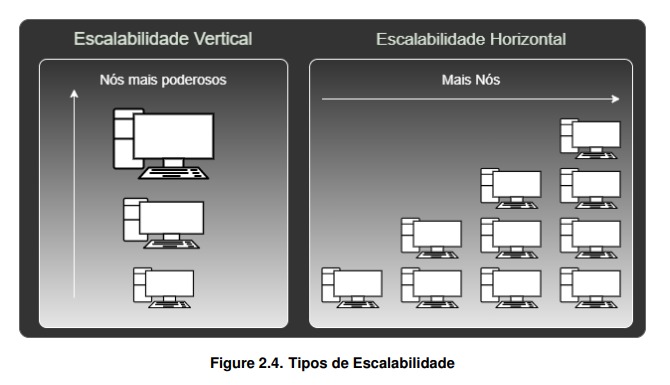
## arquitetura *serverless*

A computação sem servidor está se tornando cada vez mais atraente para arquitetos e desenvolvedores de soluções em nuvem (SATHELER, 2021). Segundo Satheler (2021), esse novo paradigma de computação deu origem a plataformas de funções como serviço, também conhecidas como Function-as-a-Service (FaaS), que permitem a implantação de funções sem se preocupar com a infraestrutura. Satheler (2021) ainda acrescenta que um desafio importante no projeto de plataformas FaaS é garantir a disponibilidade das funções implantadas. Já para Cloudflare (2022), o código FaaS é inerentemente escalável, os desenvolvedores não precisam se preocupar em criar contingências para alto tráfego ou uso pesado. O provedor sem servidor lidará com todas as preocupações de dimensionamento (CLOUDFLARE, 2022).

A arquitetura *serverless*, para Zanelato e Fontana Filho (2019), veio para acabar com a complexidade do gerenciamento de servidores. Com ela o desenvolvedor não precisa se preocupar com a maioria dos aspectos da infraestrutura em que sua aplicação será executada (ZANELATO; FONTANA FILHO, 2019). Zanelato e Fontana Filho (2019) ainda afirmam que ao mesmo tempo que essa arquitetura oferece maior escalabilidade, flexibilidade, diminuição no tempo de liberação de versões e custo reduzido, ela ajuda os desenvolvedores a entregar muito mais software em um mesmo período.

Escalabilidade para Rocha *et al.* (2021) é um termo utilizado em sistemas, que diz respeito à capacidade de um sistema crescer, tendo como intenção atender mais usuários ou adicionar mais funcionalidades. Sendo assim, um sistema é dito escalável quando o seu desempenho aumenta proporcionalmente com o seu poder computacional (ROCHA *et al*., 2021). Rocha *et al.* (2021) ainda complementam que pode se dividir escalabilidade em duas vertentes, escalabilidade horizontal e escalabilidade vertical. Que pode ser visualizada na Figura 4. Na escalabilidade horizontal se aumenta o poder computacional do sistema adicionando mais nós ao sistema, ou seja, adicionando uma nova máquina. Já na escalabilidade vertical se aumenta o poder computacional do sistema melhorando um nó existente, como, adicionando mais memória Random Access Memory (RAM).

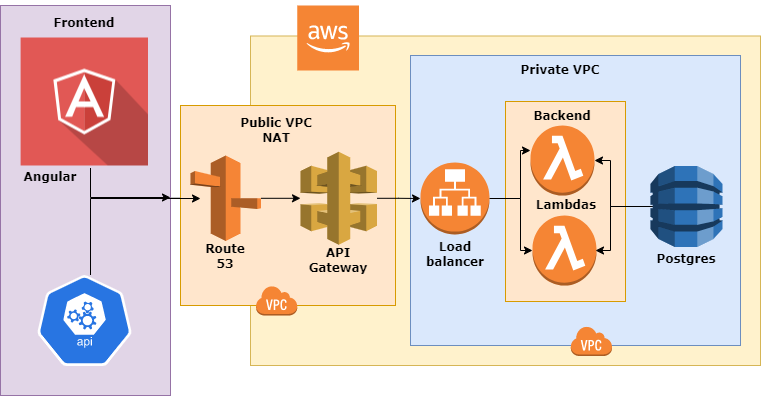
Figura – As vertentes da escalabilidade



Fonte: Silva (2018).

Amazon (2022a) ainda destaca que quando se usa tecnologias *serverless* não se preocupa com o tipo de escalabilidade pois contam com escalabilidade automática, alta disponibilidade integrada e modelo de faturamento pago por utilização. Além disso, segundo Amazon (2022a), as aplicações Serverless *framework* iniciam com AWS Lambda, um serviço orientado a eventos integrado nativamente com outros serviços AWS. A Figura 5 traz o protótipo da arquitetura serverless para a proposta, de como será desenvolvida a aplicação, na Amazon Virtual Private Cloud (VPC). Pela referida figura é possível perceber as tecnologias de escalabilidade e disponibilidade automática, como as Lambdas que serão duplicadas pelo balanceador de cargas *load balancer*, conforme a quantidade de usuários estiverem acessando a aplicação. Na Network Address Translation (NAT) o Route 53 que é o domínio e a API Gateway receptor dessas requisições (AMAZON, 2022a).

Figura – Protótipo da arquitetura *serverless* para a proposta



Fonte: elaborado pela autora.

Shafiei *et al.* (2020) complementam que a computação *serverless* se difere da computação em nuvem tradicional no sentido em que a infraestrutura e a plataforma às quais os serviços estão executando são de fato transparentes para o usuário da nuvem. Nessa abordagem o desenvolvedor se preocupa exclusivamente com as funcionalidades demandadas por suas aplicações e todo o resto é delegado ao provedor de serviço em nuvem. Vieira *et al.* (2020) ainda destacam que *serverless* é uma camada de serviço que realmente abstrai a infraestrutura de nuvem subjacente, minimizando a curva de aprendizado para que desenvolvedores e projetos utilizem programação em nuvem. Potencialmente, por conseguir contabilizar o uso de forma mais granular, reduz os custos com infraestrutura de nuvem para corporações além de discriminar melhor onde os custos foram alocados.

Referências

AMARAL, Marcus Vinícius do. **Desenvolvimento de um software para controle de entrada e saída de portaria**. 2020. Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Software), Faculdade de Tecnologia de Americana, Americana.

AMAZON. **Serverless on AWS**. AWS, 2022a. Disponível em: https://aws.amazon.com/serverless/?nc1=h\_ls. Acesso em: 17 set. 2022.

AMAZON. **O que é a AWS Lambda?** AWS, 2022b. Disponível em: https://docs.aws.amazon.com/pt\_br/lambda/latest/dg/welcome.html. Acesso em: 14 nov. 2022.

AMAZON. **Amazon API Gateway**.AWS, 2022c. Disponível em: https://aws.amazon.com/pt/api-gateway. Acesso em: 26 nov. 2022.

CLOUDFLARE. **What is Function-as-a-Service (FaaS)?** Built-in scalability. Cloudflare, 2022. Disponível em: https://www.cloudflare.com/pt-br/learning/serverless/glossary/function-as-a-service-faas/. Acesso em: 11 set. 2022.

MARIANO, Gabriel Felipe Lopes; MONTANHA, Gustavo Kimura. Desenvolvimento de sistema web para gerenciamento de controle de acesso. **Tekhne e Logos**, Revista da Fatec Botucatu, v. 10, n. 2, p. 112-125, 2019.

MORAIS, Gabriel Ferreira; SILVA, Pedro Henrique Braga da; DÁGOLA, Pedro Henrique Castello Branco. Sistema de controle de acesso por RFID. **Século 21**: o papel da escola para diferentes gerações de alunos, Coninf, v.1, 2018.

OLIVEIRA, Aryeverton Fortes de. **Empresas de vigilância no sistema de prestação de serviços de segurança patrimonial privada**: Uma avaliação da estrutura de governança. 2004. Tese (Doutorado), Universidade de São Paulo, São Paulo.

OLIVEIRA, William Cristiano de *et al.* Automatização do controle de acesso a portaria na entrada e saída de veículos nas dependências da empresa. **Tekhne e Logos**, Revista da Fatec Botucatu, v.11, n. 1, p. 60-68, 2020.

PINHEIRO, José Maurício. **Biometria nos Sistemas Computacionais**. 1ªEdição. Ciência Moderna, 2008.

ROCHA, Antonio A. de A. *et al.* Sociedade Brasileira de Computação.**Segurança e Escalabilidade em Sharding Blockchain**. Belém, 2021.

ROVANI, Esequiel. **Sistema para controle de acesso integrado com central de alarme**. 2019. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Elétrica), Universidade de Passo Fundo, Passo Fundo.

SATHELER, Gustavo Bittencourt. **Alta disponibilidade de funções como serviço em ambiente de múltiplas nuvens de computação**. 2021. Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Software), Universidade Federal do Pampa, Alegrete.

SERVERLESS. **A maneira fácil e aberta de criar aplicativos sem servidor.** Serverless, 2022. Disponível em: https://www.serverless.com/framework. Acesso em: 14 nov. 2022.

SHAFIEI, Hossein; KHONSARI, Ahmad; MOUSAVI, Payam. **Serverless Computing**: A Survey of Opportunities, Challenges and Applications**.** Iran,2019. Disponível em: https://www.researchgate.net/publication/338108229\_Serverless\_Computing\_A\_Survey\_of\_Opportunities\_Challenges\_and\_Applications. Acesso em: 14 nov. 2022.

SILVA, Fernando Léo Bueno de Oliveira e. **Sistema de controle de acesso para estacionamento**. 2018. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Elétrica), Universidade de Taubaté, Taubaté.

SOUZA, Marcelo Barboza. **Controle de Acesso**: Conceitos, Tecnologias e Benefícios. Editora Sicurezza, 2010.

VIEIRA, André G. *et al.* Sociedade Brasileira de Computação. **Computação Serverless**: Conceitos, Aplicações e Desafios. Porto Alegre, 2020.

ZANELATO, Alexandre Davi; FONTANA FILHO, Neri Burato Bez. **Arquitetura Serverless baseada em eventos para aplicações web utilizando AWS**. 2019. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação), Universidade do Sul de Santa Catarina, Florianópolis.

ZAPAROLI, Maurício Xavier. **SmartLock**: Controle de acesso através de Smart Contracts e Smart Property. 2019. Dissertação (Mestrado em Ciências e Tecnologia da Computação), Universidade Federal de Itajubá, Itajubá.