

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

## **PORTARIA FÁCIL: APLICAÇÃO DE CONTROLE DE ACESSO COM ALTA ESCALABILIDADE, DISPONIBILIDADE E SUSTENTABILIDADE**

Jéssica Maria Lana de Melo Kohn

Prof. Simone Erbs da Costa – Orientadora

### **1 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 **seguimento** precisa ser feito de maneira eficaz, ajustando pontos de falha na segurança.

Nesse contexto, estão a alta disponibilidade e escalabilidade. Para Satheler (2021), 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 (2019, tradução nossa), 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 ~~framework~~. Conjectura-se assim agilizar o controle de acesso de visitantes e proporcionar segurança para os presentes nas instalações de empresas e condomínios.

## 1.1 OBJETIVOS

O objetivo geral deste trabalho é desenvolver uma aplicação de controle de acesso que proporcione uma experiência diferenciada. Os objetivos específicos são:

- a) controlar e agilizar o acesso de moradores e visitantes por meio de interfaces disponibilizadas;
- b) garantir alta disponibilidade e escalabilidade com o uso de recursos e produtos disponíveis na nuvem;
- c) 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).

## 2 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).

### 2.1 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 Moraes, Silva e Dágola (2018) são: gerenciamento com interface; manter cadastro do usuário;

manter cadastro de data e hora; filtrar dados; gerar relatórios; gerar as notificações para os usuários; 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 Moraes, Silva e Dágola (2018) contém a interface gráfica do *front-end* e funciona em um computador que está montado na portaria de entrada do Instituto. O *back-end* é a interface pela qual os gestores e administradores têm acesso a **geração dos relatórios com as informações cadastradas no banco de dados**. Essa interface é responsável pela gravação dos cartões de RFID, pois no *back-end* existe a permissão de acesso ao banco de dados que permite a gravação. 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).

O *middleware* que o projeto ~~de pesquisa~~ desenvolveu é composto, basicamente, de quatro blocos principais: **fonte de alimentação para os circuitos eletrônicos, placa de circuito impresso com o microcontrolador e componentes eletrônicos montados, placa leitora dos cartões RFID e o gabinete para acondicionamento dos circuitos**. O *middleware* se comunica com o computador por meio de uma porta Universal Serial Bus (USB) e **realiza a gravação dos cartões RFID**, conforme comunicação enviada pelo *back-end*. O *middleware* também realiza a leitura das informações gravadas nos cartões RFID (MORAIS; SILVA; DÁGOLA, 2018).

Segundo Moraes, 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).

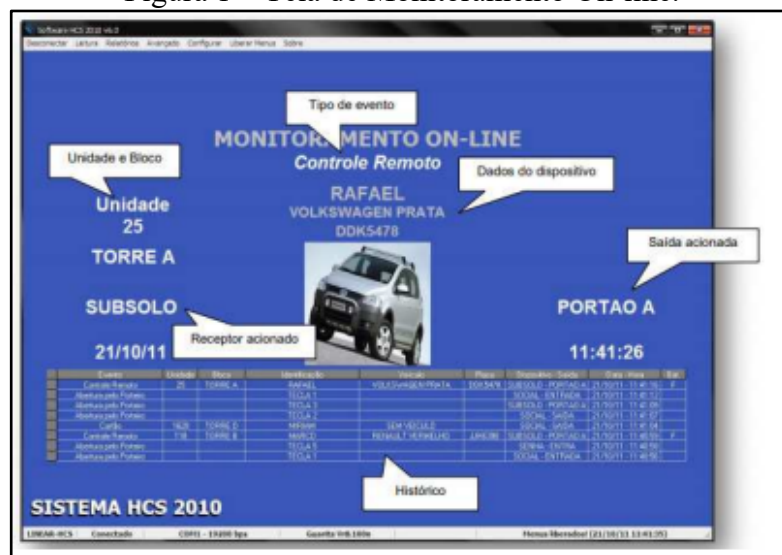
## 2.2 SISTEMA DE CONTROLE DE ACESSO PARA ESTACIONAMENTO

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 em **sua** memória interna, 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. Ao finalizar, 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 cadastro de data e hora; filtrar dados; gerar relatórios; gerar as notificações para os usuários; 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, 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 1 – Tela de Monitoramento On-line.



Fonte: Silva (2018).

## 2.3 SMARTLOCK: CONTROLE DE ACESSO ATRAVÉS DE SMART CONTRACTS E SMART PROPERTY

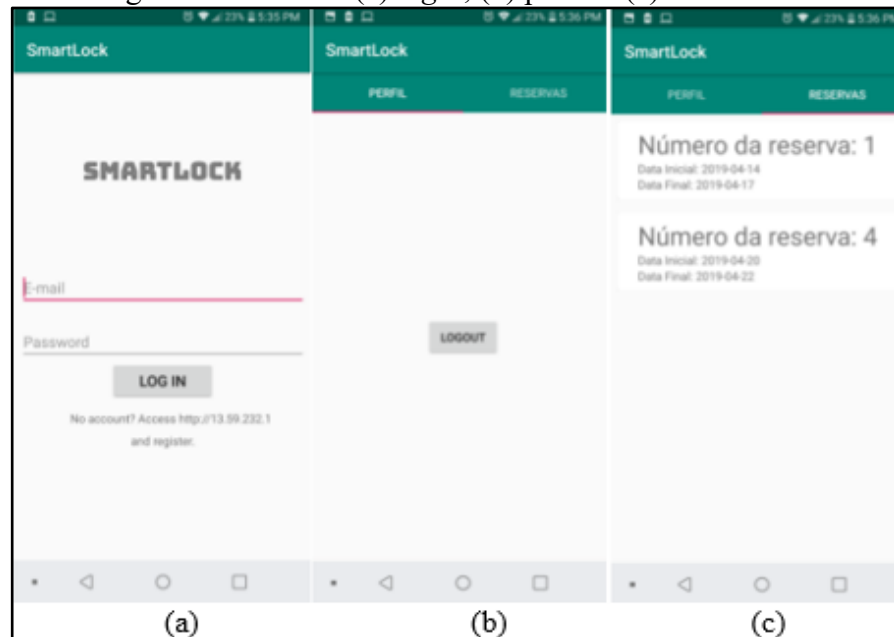
Esse trabalho propõe o desenvolvimento de uma fechadura eletrônica controlada por contratos inteligentes registrados na plataforma *blockchain* da Ethereum. A aplicação abrange

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 (ZAPAROLI, 2019).

Para Zaparoli (2019), a aplicação possui funcionalidades como: gerenciamento com interface; manter cadastro do usuário; manter cadastro de data e hora; gerar as notificações para os usuários; 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 2 – Telas de (a) *login*, (b) perfil e (c) de reservas



Fonte: Zaparoli (2019).

### 3 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).

#### 3.1 JUSTIFICATIVA

Nas seções 1 e 2 foram apresentados a relevância do trabalho proposto. 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, Moraes, 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 1 – Comparativo dos trabalhos correlatos

Trabalhos Correlatos	Morais, Silva e Dágola (2018)	Silva (2018)	Zaparoli (2019)
Características			
Gerenciamento com interface	✓	✓	✓
Manter cadastro dispositivos	X	✓	X
Manter cadastro do usuário	✓	✓	✓
Manter cadastro data e hora	✓	✓	✓
Possuir filtros	✓	✓	✓
Gerar relatórios	✓	✓	X
Notificação aos usuários	✓	✓	✓
Plataforma	Web	Web	Android

Fonte: elaborado pela autora.

Ao analisar o Quadro 1 é possível identificar que as soluções de Moraes, 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. Existe também a opção do usuário de cadastrar, editar ou apagar dispositivos sem comunicação direta em off-line, podendo executar posteriormente a função *restore* para gravar as informações no equipamento. 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 Moraes, Silva e Dágola (2018), Silva (2018) e Zaparoli (2019). Já as características de manter um cadastro de data e hora, assim como de possuir filtros também estão disponíveis nas aplicações de Moraes, 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. Já a característica de exibir notificações aos usuários, lembretes, comunicados sobre registros, informações que foram salvas, problemas que podem ocorrer com a comunicação entre os equipamentos, estão presentes Moraes, Silva e Dágola (2018), Silva (2018) e Zaparoli (2019). Por fim, cabe destacar que Moraes, Silva e Dágola (2018) e Silva (2018) disponibilizaram a aplicação na plataforma Web, enquanto Zaparoli (2019) disponibilizou sua solução para plataforma Android. A proposta aponta paridade com os três trabalhos correlatos apresentados. A aplicação enfatiza o controle de acesso, o gerenciamento das pessoas que desejam acessar o local ou empresa. A importância da segurança nos dias de hoje, o bloqueio de alguém não identificado, a entrada facilitada de quem tem horário de entrada prevista. Além disso, a proposta se destaca por meio da alta disponibilidade da aplicação, permitindo o controle de acesso a portaria a qualquer dia e horário, com um sistema sempre disponível. Ela ainda traz o diferencial de propor uma aplicação com alta escalabilidade, que é a possibilidade de muitos usuários acessarem no mesmo instante.



Com essas características e tal como as apresentadas no Quadro 1, é perceptível que o trabalho possui importância para a sociedade. A proposta concentra a premissa da importância da segurança para a comunidade, visto a necessidade de proteção em condomínios e empresas. Para a contribuição tecnológica, será desenvolvida uma aplicação tanto Web quanto móvel para Android trazendo mais possibilidades para o usuário, fazendo uso de tecnologias recentes, porém robustas com vasta documentação e consolidadas no mercado. No *back-end* Lambda com Serverless *framework*, e base PostgreSQL, no *front-end* Ionic. Além disso, fará uso de recursos e produtos Amazon Web Services (AWS), visando a alta disponibilidade e escalabilidade. 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 outros trabalhos, assim como referencial da AWS.

### 3.2 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 2 – Principais Requisitos Funcionais e Não Funcionais

O aplicativo deve:	Tipo
permitir ao usuário realizar <i>login/logout</i>	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 agendamento de visitas (CRUD)	RF
permitir ao usuário visualizar quantidades de visitas (dia, mês, ano)	RF
notificar os usuários por meio de notificações <i>push</i>	RF
ser construído utilizando um <i>framework</i> Ionic para o <i>front-end</i>	RNF
ser construído utilizando Lambdas com Serverless <i>framework</i> para o <i>back-end</i>	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
possuir acesso a câmera	RNF
possuir acesso a internet	RNF

Fonte: elaborado pela autora.

### 3.3 METODOLOGIA

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

- aprofundamento bibliográfico: realizar aprofundamento na literatura sobre os assuntos como: o que seria, como criar uma Lambda em Serverless *framework*;



- b) levantamento dos requisitos: analisar os requisitos funcionais e não-funcionais já definidos e, se necessário, especificar outros a partir da etapa do aprofundamento realizado;
- c) 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;
- d) implementação: desenvolver a aplicação utilizando no *front-end* Ionic e na Application Programming Interface (API) que será publicada utilizando recursos e produtos (AWS) como Lambdas com Serverless *framework* em conjunto com um banco de dados PostgreSQL;
- e) verificação, validação e análise: paralelamente a implementação, realizar testes das funcionalidades da aplicação, bem como da usabilidade e experiência do usuário das interfaces desenvolvidas por meio do método RURUCAg.

Quadro 3 – Cronograma

Etapas	Quinzenas		2023									
			fev.		mar.		abr.		maio		jun.	
	1	2	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.

## 4 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 sustentabilidade; e por fim, a subseção 4.3 aborda os temas de alta disponibilidade, escalabilidade.

### 4.1 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) ainda complementa 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).

#### 4.2 SUSTENTABILIDADE

Os principais consumidores de energia em um servidor, segundo Bergman (2020, tradução nossa), são a Central Processing Unit (CPU), a Graphics Processing Unit (GPU) e a memória. A quantidade de energia que cada um desses componentes consome passa uma estimativa de quanta energia seu servidor ou seu aplicativo consome (BERGMAN, 2020, tradução nossa). Para Rakhunathan (2020, tradução nossa), uma maneira de reduzir a emissão de carbono de nossas aplicações é reduzir o uso de CPU, GPU e memória dos servidores nos quais as cargas de trabalho estão sendo executadas. Bergman (2020, tradução nossa) coloca que isso vai funcionar como uma estimativa com o objetivo de começar a rastrear o consumo de energia e garantir que esteja melhorando. Rakhunathan (2020, tradução nossa) ainda complementa que quanto menos energia for consumida por esses componentes, menor será a quantidade de combustíveis fósseis necessária para alimentar o data center.

Mas isso não significa diminuir as cargas de trabalho, em vez disso, é necessário que se auxilie as organizações a executar servidores com uma taxa de utilização mais alta e as arquiteturas *serverless* tem esse objetivo (RAKHUNATHAN, 2020, tradução nossa). Para Amazon (2022b, tradução nossa), além das emissões do uso de energia, existe uma ampla gama de emissões indiretas, incluindo aquelas associadas à construção de data centers. Nesse sentido, as empresas estão reduzindo essas emissões trabalhando com usinas que usam processos de produção de fornos elétricos (AMAZON, 2022b, tradução nossa). Hogue (2020, tradução nossa) coloca um caso que aconteceu com o Azure que concluiu com êxito a migração de dezenas de pipelines de dados de *WebJobs* para funções *serverless*. Hogue (2020, tradução nossa) ainda destaca que essa refatoração arquitetônica reduziu não apenas o Azure Spend, mas também o consumo de carbono em 10 vezes, pois não foram gastos mais ciclos desnecessários em operações ou atualizando com o mesmo conjunto de dados.

Essa abordagem é consistente com o princípio de sustentabilidade de proporcionalidade de energia, especificamente maximizar os níveis de utilização, usando computação baseada em consumo elástico (HOGUE, 2020, tradução nossa). Nesse sentido, a Amazon (2022b, tradução

nossa) investe em sustentabilidade em toda sua infraestrutura, alimenta os data centers com energia renovável e possuem o compromisso de atingir zero carbono em operações até 2040. Além disso, Amazon (2020b, tradução nossa) coloca que para quem usa recursos AWS foi desenvolvida a ferramenta Customer Carbon Footprint Tool com o objetivo de medir as emissões de carbono.

### 4.3 ALTA DISPONIBILIDADE E ESCALABILIDADE

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. Para Cloudflare (2022, tradução nossa), 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, tradução nossa).

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). Dessa forma, Zanelato e Fontana Filho (2019) 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 de tempo. Já para Amazon (2022a, tradução nossa), as tecnologias *serverless* contam com escalabilidade automática, alta disponibilidade integrada, modelo de faturamento pago por utilização. Além disso, segundo Amazon (2022a, tradução nossa), as aplicações *Serverless framework* iniciam com AWS Lambda, um serviço orientado a eventos integrado nativamente com outros serviços AWS.

### 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](https://aws.amazon.com/serverless/?nc1=h_ls). Acesso em: 17 set. 2022.

AMAZON. **Sustainability in the Cloud**. AWS, 2022b. Disponível em: <https://sustainability.aboutamazon.com/environment/the-cloud?energyType=true>. Acesso em: 24 set. 2022.

BERGMAN, Sara. **How to measure the power consumption of your backend service.** Microsoft, 2020. Disponível em: [https://devblogs.microsoft.com/sustainable-software/how-to-measure-the-power-consumption-of-your-backend-service/?WT.mc\\_id=-green-8647-cxa#what-if-i-run-in-the-cloud](https://devblogs.microsoft.com/sustainable-software/how-to-measure-the-power-consumption-of-your-backend-service/?WT.mc_id=-green-8647-cxa#what-if-i-run-in-the-cloud). Acesso em: 6 set. 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.

HOGUE, Aaron. **How Azure.com uses Serverless Functions for Consumption-based utilization and reduced always-on electric footprint.** Microsoft, 2020. Disponível em: <https://devblogs.microsoft.com/sustainable-software/how-azure-com-uses-serverless-functions-for-consumption-based-utilization-and-reduced-always-on-electric-footprint/>. Acesso em: 6 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, 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.

RAKHUNATHAN, Srinivasan. **Adopting Azure serverless architectures to help reduce CO2 emissions – Part 1.** Microsoft, 2020. Disponível em: <https://devblogs.microsoft.com/sustainable-software/adopting-azure-serverless-architectures-to-help-reduce-co2-emissions-part-1/>. Acesso em: 11 set. 2022.

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.

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.

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.

# FORMULÁRIO DE AVALIAÇÃO SIS ACADÊMICO PROFESSOR AVALIADOR – PRÉ-PROJETO

Avaliador(a): Marcel Hugo

Atenção: quando o avaliador marcar algum item como atende parcialmente ou não atende, deve obrigatoriamente indicar os motivos no texto, para que o aluno saiba o porquê da avaliação.

ASPECTOS AVALIADOS		atende	atende parcialmente	não atende
ASPECTOS TÉCNICOS	1. INTRODUÇÃO O tema de pesquisa está devidamente contextualizado/delimitado?		X	
	O problema está claramente formulado?		X	
	2. OBJETIVOS O objetivo principal está claramente definido e é passível de ser alcançado?		X	
	Os objetivos específicos são coerentes com o objetivo principal?		X	
	3. TRABALHOS CORRELATOS São apresentados trabalhos correlatos, bem como descritas as principais funcionalidades e os pontos fortes e fracos?		X	
	4. JUSTIFICATIVA Foi apresentado e discutido um quadro relacionando os trabalhos correlatos e suas principais funcionalidades com a proposta apresentada?	X		
	São apresentados argumentos científicos, técnicos ou metodológicos que justificam a proposta?		X	
	São apresentadas as contribuições teóricas, práticas ou sociais que justificam a proposta?	X		
	5. REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO Os requisitos funcionais e não funcionais foram claramente descritos?		X	
	6. METODOLOGIA Foram relacionadas todas as etapas necessárias para o desenvolvimento do TCC?	X		
	Os métodos, recursos e o cronograma estão devidamente apresentados e são compatíveis com a metodologia proposta?	X		
	7. REVISÃO BIBLIOGRÁFICA Os assuntos apresentados são suficientes e têm relação com o tema do TCC?		X	
	As referências contemplam adequadamente os assuntos abordados (são indicadas obras atualizadas e as mais importantes da área)?	X		
ASPECTOS METODOLÓGICOS	8. LINGUAGEM USADA (redação) O texto completo é coerente e redigido corretamente em língua portuguesa, usando linguagem formal/científica?	X		
	A exposição do assunto é ordenada (as ideias estão bem encadeadas e a linguagem utilizada é clara)?	X		