



UNIVERSIDADE FEDERAL DO RIO GRANDE  
DO SUL

ENGENHARIA DE CONTROLE E AUTOMAÇÃO

CCA99003 - PROJETO II

---

SISTEMA DE GERENCIAMENTO DE PONTO  
ELETRÔNICO

---

*Autores:*

Alex Treviso

Bruno Cappellari

Guilherme Marchioro

Henrique S. Coelho

Pedro Victor H. V. Garcia

*Matrícula:*

259921

288580

291403

243627

291402

Fevereiro de 2022

<b>1</b>	<b>Introdução</b>	<b>2</b>
1.1	Objetivos . . . . .	3
1.1.1	Objetivos Gerais . . . . .	3
1.1.2	Objetivos Específicos . . . . .	3
<b>2</b>	<b>Análise de Mercado</b>	<b>4</b>
2.1	Pontomais . . . . .	4
2.2	EzPointWeb . . . . .	4
2.3	Factorial . . . . .	5
<b>3</b>	<b>Funcionalidades</b>	<b>5</b>
<b>4</b>	<b>Modelagem UML do sistema</b>	<b>6</b>
<b>5</b>	<b>Banco de dados</b>	<b>7</b>
5.1	Tipos de banco de dados . . . . .	8
5.1.1	Bancos de dados relacionais . . . . .	8
5.1.2	Bancos de dados não-relacionais . . . . .	9
5.2	Definição do gerenciador de banco de dados . . . . .	10
5.3	Diagrama entidade relacionamento . . . . .	10
<b>6</b>	<b>Próximos Passos</b>	<b>12</b>
6.1	Web App . . . . .	12
<b>7</b>	<b>Conclusão</b>	<b>13</b>

## 1 Introdução

O controle de ponto pode ser descrito como um sistema para a gestão da jornada de trabalho, permitindo que sejam obtidos registros de jornadas de trabalho de maneira precisa, confiável e transparente. Tem como principal objetivo a aquisição do registro dos horários de entrada e saída dos colaboradores de uma empresa.

Empresas que buscam excelência operacional tendem a voltar seu foco e atenção para todos os processos que compõe o seu negócio. Desta forma, faz-se extremamente necessário a busca por ferramentas capazes de tornar processos mais ágeis e menos burocráticos. A eficiência operacional está diretamente ligada à redução de custos e ao aumento da produtividade, fatores que relacionam-se diretamente com o uso de registro de ponto eletrônico.

Através de uma experiência de um dos integrantes do grupo, realizando uma consultoria para uma empresa do ramo de produção de erva mate, após instituído o controle de ponto através de planilhas pôde-se verificar uma melhor gestão das horas trabalhadas pelos funcionários, reduzindo principalmente custos adicionais com horas extras e possibilitando análise de métricas de produção por horas trabalhadas mês-a-mês, o que auxilia principalmente na tomada de decisões por parte dos gestores da empresa.

Porém as planilhas acabam por apresentar certas limitações, como a necessidade da geração de uma nova planilha todos os meses, dificuldades de integração ou visualização dos dados entre os meses, grande trabalho por parte de uma pessoa para digitar os horários de todos os funcionários da empresa.

Assim sendo, como alternativa para tais planilhas, prevê-se com este projeto o desenvolvimento de um software para gerenciamento de ponto com interface intuitiva principalmente para visualização de totalizações de horários, registro de histórico/log de informações para cada funcionário e integração com banco de dados. Em segundo plano, será analisada a viabilidade da integração do software com um hardware para facilitar a entrada de dados.

Com o desenvolvimento deste projeto espera-se um sistema capaz de melhorar a produtividade e consequentemente resultados, através das métricas e informações obtidas; auxiliar a gestão, principalmente da área de recursos humanos; favorecer o acesso dos dados, através da disponibilização do sistema via web, permitindo o acesso a qualquer momento, de qualquer dispositivo conectado à internet.

## **1.1 Objetivos**

Abaixo são apresentados os objetivos gerais e específicos do projeto.

### **1.1.1 Objetivos Gerais**

O principal objetivo desse projeto é eliminar o controle e a análise manual das horas de trabalho dos colaboradores de um empreendimento.

### **1.1.2 Objetivos Específicos**

- Criação de banco de dados para armazenamento de registros de jornadas de trabalho de usuários;
- Estabelecer a comunicação entre diferentes sistemas;
- Criar uma interface intuitiva de usuário;
- Visualização dos dados em múltiplos dispositivos pela web.

## 2 Análise de Mercado

Atualmente a empresa com mais agilidade e desempenho é aquela que acaba dominando o mercado, por isso despender tempo para gerenciar e monitorar o tempo da equipe não é mais aceitável. Logo, surgiram no mercado soluções que monitoram o ponto dos colaboradores e o desempenho da equipe. Com essas soluções, a equipe economiza tempo e pode se dedicar as operações que importam. As soluções permitem acessar e controlar as marcações de ponto, histórico de presença, consultas, lançamentos, relatórios e etc. de forma rápida e confiável. A administração dos softwares podem ser feitas de qualquer lugar de forma online via internet e operam de forma segura onde todos os dados são salvos em nuvem. As similaridades mais relevantes dentre essas soluções são: controle de ponto biométrico, com cartão e com smartphone, além de gestão de banco de horas, firewall de segurança e relatórios conforme legislação. Dentre algumas dessas soluções para o controle de ponto, pode-se destacar: Pontomais, EzPointWeb e Factorial.

### 2.1 Pontomais

A Pontomais enfatiza quatro aspectos da sua solução: as informações de jornada de trabalho são salvas diretamente na nuvem, garantindo a segurança das informações de registro de ponto, as informações ficam disponíveis em tempo real; otimiza o tempo do fechamento dos horários de ponto, de cálculo de horas extras, da folha de pagamento e férias; demonstra uma visão geral sobre a jornada de trabalho de cada colaborador, é possível acessar o desempenho dos funcionários em tempo real e de qualquer lugar, não sendo necessária a compra do relógio ponto.

Figura 1: Logo da Pontomais



Fonte: Pontomais.

### 2.2 EzPointWeb

A EzPointWeb destaca quatro vantagens da sua solução: todos os dados são salvos em nuvem, não corre o risco de perder nenhuma informação; é um sistema de controle de ponto eletrônico completo, que armazena e monitora o desempenho da sua equipe; agiliza o fechamento da folha de ponto, faz uma gestão mais apurada

da produtividade em poucos minutos e reduz o tempo no controle de ponto dos funcionários.

Figura 2: Logo da EzPointWeb



Fonte: EzPointWeb.

## 2.3 Factorial

A Factorial enfatiza quatro funcionalidades da sua solução: gestão de férias e ausências, os colaboradores podem solicitar as suas ausências em qualquer momento e saber quantos dias de férias têm a partir do seu perfil; toda a informação e documentação dos colaboradores em um único lugar, não precisa enviar os os contracheques e holerites por e-mail, nem atualizar constantemente a informação de cada colaborador; gestão documental, controle de toda a documentação dos colaboradores de forma segura e compartilhamento de documentos da empresa com quem precisa, não sendo necessário enviar folhas de pagamento, holerites e outros documentos aos colaboradores. Dispõe da criação de relatórios personalizados baseados nos dados da empresa e conhecimento, em todo momento, o estado da equipe, estrutura, a média de idades dos trabalhadores, entre outros.

Figura 3: Logo da Factorial



Fonte: Factorial.

## 3 Funcionalidades

Tendo feita a análise de mercado de soluções comerciais semelhantes com o propósito do projeto aqui apresentado, pode-se ter um melhor embasamento para a

especificação das funcionalidades do sistema.

A maior semelhança entre as soluções encontradas é principalmente a disponibilidade de um banco de dados para armazenamento da informação dos funcionários da empresa, bem como das suas jornadas de trabalho diárias, as quais podem ser registradas de diferentes formas. Dessa maneira torna-se indispensável tal funcionalidade na aplicação proposta, facilitando a manutenção dos dados e a criação de um histórico valioso, auxiliando análises do setor de recursos humanos e também do setor executivo da empresa, através de métricas que podem ser calculadas a partir dos dados.

Atentando para as necessidades específicas da ervateira é importante que os dados registrados possam ser totalizados, principalmente mês-a-mês, para que possam ser utilizados para fins do fechamento das folhas de pagamento dos trabalhadores. Assim sendo, é previsto para o software a ser projetado a capacidade de, através dos dados registrados no banco de dados, calcular as horas totais trabalhadas, horas extras ou negativas, levando em conta possíveis adicionais devido a trabalho em um turno diferente do especificado para determinado funcionário, para que o fechamento da folha de pagamento possa ser facilitado.

## 4 Modelagem UML do sistema

Dadas as funcionalidades, buscou-se representar o sistema a ser desenvolvido. Utilizou-se para tanto o paradigma de orientação a objetos, sendo possível dividir o sistema em classes, relacionadas entre si, contendo diferentes atributos e métodos para tal.

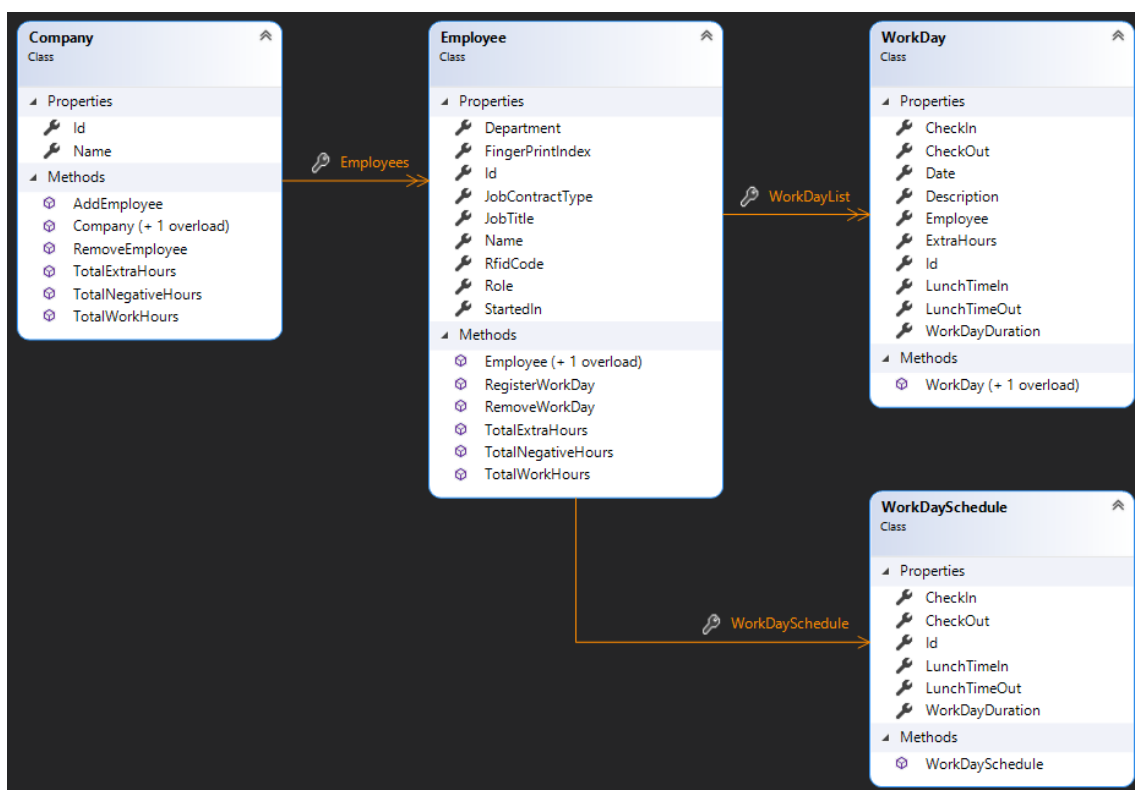
Tendo em vista o sistema aqui apresentado, idealizou-se as principais entidades envolvidas no processo do controle do ponto dos funcionários. Para isso definiu-se uma classe relacionada a empresa, denotada pela classe *Company* da Figura 4, a qual é responsável principalmente pela gestão dos funcionários da empresa, dessa maneira contendo uma lista de objetos da classe *Employee*, e contendo métodos para a adição/remoção destes funcionários. Além disso previu-se na classe *Company* métodos relacionados a totalização das horas trabalhadas pelos funcionários, dadas datas inicial e final.

A classe *Employee*, contém todas as informações a respeito do funcionário em si, úteis não só para o registro do ponto mas também para facilitar o cálculo da folha de pagamento. A cada funcionário é atribuída uma jornada de trabalho específica a ser desempenhada, modelada pela classe *WorkDaySchedule*, a qual é comparada com os registros de fato das jornadas de cada funcionário (modelados pela classe *WorkDay*), sendo assim possível identificar horas extras, negativas ou aditivos referentes a trabalho fora do turno do funcionário. Para a gestão das jornadas de trabalho são previstos métodos para adição e remoção dos registros. Além disso,

assim como na classe *Company*, são previstos métodos de totalização dos resultados das horas trabalhadas, porém específicas para o objeto do funcionário em questão.

A estrutura referente a jornada de trabalho a ser registrada é modelada pela já mencionada classe *WorkDay*, a qual é vinculada a um funcionário específico e contém propriedades quanto a horários de chegada, saída, pausa para almoço e de duração da jornada.

Figura 4: Modelo UML proposto



Fonte: Elaborada pelos autores

## 5 Banco de dados

Atualmente os dados vêm ganhando importância cada vez maior em diferentes mercados e possuir uma cultura focada em dados apresenta-se como uma boa opção para muitos negócios. Com a recente ascensão da indústria 4.0 e os dispositivos cada vez mais inteligentes e conectados, ter conhecimento e controle dos dados é de suma importância do ponto de vista do negócio.

O foco em dados, muitas vezes com a utilização de abordagens *data-driven*, que está presente há alguns anos na área de tecnologia, agora é também um termo



bastante debatido em outros segmentos. Indústrias, agronegócio, educação e mesmo os setores de comunicação, como o marketing, têm muito a ganhar com as soluções baseadas nesse recurso.

Deste modo, é de extrema importância que empresas que optem por focar em seus dados façam a utilização de bancos de dados.

Um sistema de banco de dados é, basicamente, um sistema de manutenção de registros que pode ser considerado como o equivalente de um armário de arquivamento: ou seja, é um repositório de arquivos computadorizados [1]. O banco de dados permite que o usuário manipule as informações nele registradas, através de diferentes operações como inserção de dados, consulta, exclusão, modificação e remoção. Em outras palavras, é um sistema computadorizado cuja finalidade geral é armazenar informações e permitir que os usuários busquem e atualizem essas informações quando as solicitar.

Tendo em vista o constante crescimento na área de tecnologia, presenciamos cada vez mais a utilização de bancos de dados como forma de armazenamento de dados. Ainda, a utilização de sistemas de gerenciamento de banco de dados, que fornecem a capacidade da realização de operações em banco de dados é cada vez maior. Segundo [2], um sistema de gerenciamento de banco de dados, ou SGBD, é um *software* projetado para auxiliar a manutenção e utilização de vastos conjuntos de dados.

Neste capítulo serão apresentados os diferentes tipos de bancos de dados e seus respectivos SGDBs.

### 5.1 Tipos de banco de dados

Visando um maior entendimento da utilidade do armazenamento de dados em bancos, é necessário conhecer as duas categorias das quais os banco de dados subdividem-se: relacionais e não relacionais. Nesta seção serão apresentados as duas categorias e suas definições, bem como será realizado um apanhado dos principais SGDBs para cada categoria.

#### 5.1.1 Bancos de dados relacionais

Segundo [3], o modelo de banco de dados relacional é uma coleção de múltiplos conjuntos de dados organizados em tabelas (linhas/colunas), possuindo uma relação bem definida entre as mesmas. A linguagem SQL (*Structured Query Language*) é usada por oferecer uma interface facilitada para interação com o banco.

O conceito de modelo relacional foi criado em 1970 por Edgar Frank Codd, sendo descrito no artigo "Relational Model of Data for Large Shared Data Banks", sendo considerado o sucessor do modelo hierárquico. O modelo relacional utiliza como base o conceito da relação matemática, da lógica e da teoria de conjuntos.

O modelo relacional fornece uma separação clara entre como a representação física e lógica da informação é representada [4]. Neste modelo, os dados são projetados como uma tabela ficando a cargo do sistema de gerenciamento de banco de dados determinar uma maneira de como armazenar essas informações. Entre outros elementos que tornam-se importantes em um modelo relacional, pode-se citar o esquema de controle de transações, a normalização dos dados e as linguagens de definição e consulta de dados.

Para que a integridade dos dados seja garantida é necessário que as transações mantenham as propriedades conhecidas como ACID [5]. São elas:

- Atomicidade: em uma transação envolvendo duas ou mais partes de informações, a transação será realizada em sua totalidade ou não será realizada, garantindo assim que as transações sejam atômicas
- Consistência: a transação irá gerar um novo estado no banco ou, caso alguma falha ocorra, irá retornar ao estado anterior antes da inicialização da transação
- Isolamento: uma operação em andamento e ainda não validada deverá permanecer isolada de qualquer outra operação, garantindo que não ocorra nenhuma interferência
- Durabilidade: dados validados serão registrados pelo sistema de modo a estarem disponíveis mesmo no caso de uma falha ou reinício do sistema

Dentre os principais sistemas de gerenciamento de bancos de dados relacionais atuais destacam-se o MySQL, o PostgreSQL, o Microsoft SQL Server e o OracleDB.

### 5.1.2 Bancos de dados não-relacionais

Devido ao expressivo crescimento de informações geradas diariamente, percebeu-se a necessidade de possibilitar uma maior flexibilidade na forma como os dados são armazenados, conflitando com a forma estruturada que bancos relacionais propõem. Deste modo, um banco não-relacional é uma alternativa para lidar com problemas vinculados à estruturação de banco de dados devido a sua arquitetura diferenciada, proporcionando uma forma facilitada de tratar alta escalabilidade de dados [6].

Ainda nesta linha, segundo [7], a categoria de bancos não relacionais foi proposta com o objetivo de atender aos requisitos de gerenciamento de grandes volumes de dados, semi-estruturados ou não estruturados, que necessitam de alta disponibilidade e escalabilidade. A necessidade de uma nova tecnologia de banco de dados surgiu como consequência da ineficiência dos bancos de dados relacionais em lidar com esta tarefa.

Banco de dados não relacional, também denominado NoSQL (abreviação para "*Not Only SQL*"), foi utilizado pela primeira vez em 1998 por Carlo Strozzi, com o nome de um banco de dados relacional de código aberto que não possuía um interface SQL [8].

No caso dos bancos NoSQL, toda a a informação necessária estará agrupada

no mesmo registro, ou seja, em vez de ter o relacionamento entre várias tabelas para formar uma informação, ela estará em sua totalidade no mesmo registro.

Dentre os principais sistemas de gerenciamento de dados não-relacionais, destacam-se o MongoDB, que vem ganhando destaque e o DynamoDB, bando de dados NoSQL totalmente gerenciado e oferecido pela AWS.

### 5.2 Definição do gerenciador de banco de dados

O gerenciador de banco de dados PostgreSQL no ano de 2021 foi o segundo banco de dados mais utilizado e continua sendo, entre os gerenciadores tradicionais, o que mais cresce nos últimos 5 anos [9]. Dentre os motivos que o tem tornando tão popular estão sua alta extensibilidade seguindo as definições SQL da ISO/IEC 9075-1:2016 [10], seus sistemas de replicação síncrona ou assíncrona, proteção ACID (atomicidade, consistência, isolamento e durabilidade) e grande comunidade por se tratar de um projeto de código aberto.

O modelo de dados utilizado pelo PostgreSQL é o objeto-relacional e utiliza o SQL como linguagem de pesquisa declarativa. Como o sistema de gerenciamento de ponto eletrônico irá ingerir dados a todo e instante e também realizar consultas frequentes a melhor escolha de banco de dados é PostgreSQL, tendo até mesmo um desempenho superior a banco de dados específicos para dados temporais [11].

### 5.3 Diagrama entidade relacionamento

O digrama de entidade relacionamento proposto para o banco de dados foi baseado no diagrama UML desenvolvido anteriormente e seu resultado pode ser observado na Figura 5. O diagrama foi desenvolvido com o auxílio da ferramenta *dbdiagram.io*.

Figura 5: Diagrama de entidade relacionamento do banco de dados



Fonte: Elaborada pelos autores

Definiu-se quatro tabelas principais: *employee*, *employeeDaily*, *workSchedule* e *daily*. As tabelas *rooms* e *accessRooms* são opcionais dependendo se o projeto definido pelo cliente inclui o controle de acessos a salas específicas além do tradicional ponto eletrônico.

A tabela *employee* é responsável por armazenar os dados provenientes dos objetos da classe de mesmo nome. Na tabela *daily* estão as informações de qualquer registro efetuado no ponto eletrônico. A tabela *employeeDaily* armazena os dados limpos da tabela *daily* e está relacionada com os objetos da classe *WorkDay*. Por

fim, a tabela armazena as informações relacionadas com a classe *WorkDaySchedule*.

## 6 Próximos Passos

Com a base de dados planejada, o próximo passo é a construção de ferramentas que permitem a visualização e manipulação dos dados. O método principal de interação com a base de dados se dará a partir de um *web app*. O *web app* foi escolhido porque é multi-plataforma, portanto, é possível acessar de diversos dispositivos sem a necessidade da criação de diferentes aplicativos.

Além disso, será analisada a utilização de um relógio de ponto para a inserção de dados. Esta ferramenta facilitaria a inserção de dados possibilitando, por exemplo, que cada funcionário registre seu horário de entrada ou saída apenas com um cartão de identificação pessoal.

### 6.1 Web App

O *web API* será desenvolvido utilizando o ASP.NET Core. Um dos pontos importantes para a escolha do ASP.NET Core é que ele utiliza a linguagem de programação C#, que já é de familiaridade dos integrantes do grupo. Porém, o ASP.NET Core também apresenta outros benefícios, como a flexibilidade no ambiente de desenvolvimento e simplificação no desenvolvimento *web*.

Um aspecto que será facilitado pela utilização do ASP.NET Core é na autenticação e armazenamento de senhas dos usuários. Certas informações do banco de dados só poderão ser acessadas por funcionários com autorização, para isso, haverá tipos de usuários agrupados por permissões. Logo, cada funcionário deverá ter seu usuário e senha. A autenticação e o armazenamento de senhas, que são informações sigilosas e, portanto, não devem ser armazenadas em texto comum, podem ser gerenciadas pelo ASP.NET Core Identity.

Outro benefício na utilização do ASP.NET Core é a possibilidade de utilizar bibliotecas diferentes para o desenvolvimento da interface gráfica, como Angular e React. Como será construído um *web app* com intenção de ter uma experiência satisfatória em qualquer dispositivo, será feito um *Progressive Web App* (WPA). PWAs utilizam tecnologias web e rodam no navegador, porém tem experiência de usuário que se assemelha a uma aplicação nativa, utilizando Javascript para atualizar apenas o conteúdo da página e não a página inteira, como uma *Single Page Application*. Além disso, PWAs podem ser instaladas no computador ou celular. Portanto, será utilizado o Angular para desenvolver a aplicação, que permite a construção de PWAs através da sua ferramenta de linha de comando.

## **7 Conclusão**

## Referências

- [1] C.J. DATE. **Introdução a sistemas de Banco de Dados**. 8. ed. Rio de Janeiro: Pearson, 2004.
- [2] RAMAKRISHNAN Raghu; GEHRKE, Johanned. **Sistema de Gerenciamento de Banco de Dados**. , 3. Ed. New York, Mcgraw-hill, 2011.
- [3] D. WADE, B.; CHAMBERLIN. **IBM Relational Database Systems: The Early Years**. IEEE Annals of the History of Computing, v. 34, n. 4, p. 38-48, 2012.
- [4] STRAWN G., STRAWN C. **Relational Databases: Codd, Stonebraker, and Ellison. It Professional**. v. 18, n. 2, p.63-65, mar. 2016. Institute of Electrical and Electronics Engineers (IEEE).
- [5] Henry F.; SUDARSHAN S. SILBERSCHATZ, Abraham; KORTH. **Sistema de Banco de Dados**. 5. ed. Rio de Janeiro: Elsevier, 2006.
- [6] SOARES B. E. **Uma avaliação experimental de desempenho entre Sistemas Gerenciadores de Banco de Dados Colunares e Relacionais**. Cascavel, 2012. p.102 - Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - UNOPA. Disponível em: [http://www.inf.unioeste.br/~tcc/2012/TCC\\_Bruno.pdf](http://www.inf.unioeste.br/~tcc/2012/TCC_Bruno.pdf). Acessado em: 09/05/2022.
- [7] DE OLIVEIRA B. F., LÓSCIO; H. R. **NoSQL no Desenvolvimento de Aplicações Web Colaborativas**. VIII SBSC, Paraty, RJ. Disponível em: [https://www.addlabs.uff.br/sbbsc\\_site/SBSC2011\\_NoSQL.pdf](https://www.addlabs.uff.br/sbbsc_site/SBSC2011_NoSQL.pdf). Acesso em: 07/05/2022.
- [8] NASCIMENTO J. **NoSQL – você realmente sabe do que estamos falando?** 2010. Disponível em: <https://imasters.com.br/banco-de-dados/nosql-voce-realmente-sabe-do-que-estamos-falando>. Acesso em: 09/05/2022.
- [9] Stack Overflow. **Technology Databases Professional Developers**. Disponível em: <https://insights.stackoverflow.com/survey/2021#technology-databases-professional-developers4>. Acessado em: 07/05/2022.
- [10] **Database languages — SQL**. Disponível em: <https://www.iso.org/standard/63555.html>. Acessado em: 07/05/2022.
- [11] Rodrigo Dlugokenski. **Bancos de dados para monitoramento de desempenho de grandes redes**.