

**CENTRO PAULA SOUZA  
FACULDADE DE TECNOLOGIA DE FRANCA  
“Dr. THOMAZ NOVELINO”**

**TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**DAYLTON DUTRA BATISTA**

**SISTEMA PARA CONTROLE DE PATRIMÔNIOS ELETRÔNICOS**

Trabalho de Graduação  
apresentado à Faculdade de Tecnologia de  
Franca - “Dr. Thomaz Novelino”, como parte dos  
requisitos obrigatórios para obtenção do título  
de Tecnólogo em Análise e Desenvolvimento de  
Sistemas.

Orientador: Prof. Me. Fausto Gonçalves Cintra

**FRANCA/SP**

**2020**

Ficha catalográfica

B333s          Batista, Daylton Dutra  
                 Sistema para controle de patrimônios eletrônicos / Daylton  
Dutra Batista  
                 [ s.n], 2020  
                 19f. ; 30 cm; il

Trabalho de Graduação (Curso Superior de Análise e  
Desenvolvimento de Sistemas) Fatec - Faculdade de  
Tecnologia "Dr. Thomaz Novelino".  
Orientador: Prof.Me. Fausto Gonçalves Cintra

1. Análise e desenvolvimento de sistemas. 2. Gestão de  
patrimônio. 3. Registro de mudanças.. I. Autor. II. Título.

CDD – 004

# SISTEMA PARA CONTROLE DE PATRIMÔNIOS ELETRÔNICOS

Daylton Dutra Batista<sup>1</sup>

Fausto Gonçalves Cintra<sup>2</sup>

## Resumo

O projeto foi criado para o desenvolvimento de um *software* que atendesse aos requisitos necessários para controle e gerenciamento de patrimônios eletrônicos de uma organização. Foi idealizado para auxiliar os gestores nas tomadas de decisão quanto à quantidade de equipamentos na empresa, projetos de ampliação e prevenção de perdas. O sistema conta também com uma função de RDM (registro de mudanças) automatizada no SGBD (Sistema Gerenciador de Banco de Dados), pela qual é visualizada a vida útil dos equipamentos, além dos departamentos e colaboradores por quais utilizaram.

**Palavras-chave:** Análise e desenvolvimento de sistemas. Gestão de patrimônio. Registro de mudanças.

## Abstract

*The project was created in order to develop a kind of software that would meet the necessary requirements for control and management of an organization's electronic wealth. It was designed to assist managers in making decisions regarding the amount of equipment in the company, expansion projects and loss prevention. The system also has an automated RDM (change registration) function in the DBMS (Database Management System), through which the useful life of the equipment is visualized, besides the departments and employees they used.*

**Keywords:** System analysis and development. Patrimony management. Registration of changes.

---

1 Graduando em Análise e Desenvolvimento de Sistema pela Fatec Dr. Thomaz Novelino – Franca/SP. Endereço eletrônico: daylton.batista@fatec.sp.gov.br.

2 Mestre interdisciplinar em Desenvolvimento Regional pelo Centro Universitário Municipal de Franca (Uni-FACEF). Docente da Faculdade de Tecnologia de Franca “Dr. Thomaz Novelino”. Endereço Eletrônico: fausto.cintra@fatec.sp.gov.br

## 1 Introdução

Grandes organizações têm um trabalho maior para controlar os seus equipamentos eletrônicos, sendo que muitas delas utilizam apenas planilhas. Tal forma de controle pode colocar em risco a confiabilidade dos dados e, e na maioria dos casos, não há um controle de histórico. Com isso, surge a necessidade de um sistema para controle com banco de dados com gerenciamento efetivo.

O projeto foi desenvolvido para atender a uma demanda de mercado, a qual consiste em controlar o patrimônio eletrônico de uma empresa. O projeto permite criar um *software* para gerenciar informações dos colaboradores, departamentos e equipamentos de uma empresa, a fim de levantar o inventário de ativos. E, assim, com a visualização dos dados em forma de gráficos e listas, os gestores e analistas podem ter um maior controle da quantidade de equipamentos em operação e no estoque e decidir se há a necessidade da compra ou evitar uma aquisição maior do que o necessário. Conseguem, assim, ter uma visão clara da situação dos ativos da empresa, com uma melhor compreensão da existência ou não de extravios e se a operação se mantém conforme o esperado.

## 2 Levantamento de Requisitos

Pressman (2011) considera a engenharia de requisitos como uma das principais partes do desenvolvimento do *software*. Ele exemplifica que desenvolvedores gostam de iniciar a criação do sistema sem antes levantar corretamente os requisitos, o que pode levar a problemas futuros. O levantamento de requisitos é iniciado ao se consultar as partes interessadas para estabelecer as funcionalidades, possíveis cenários e restrições. Durante o processo, é possível entender o que o cliente deseja, analisando as necessidades e avaliando a viabilidade para chegar a uma solução.

### 2.1 Elicitação e especificação dos requisitos

Na atuação diária com controle de patrimônios em planilhas eletrônicas, foi observada a necessidade de um *software* confiável com uso de banco de dados,

restrições de usuários e de dados para manipulação das informações. Na empresa em que foram levantados os requisitos, os registros estavam em planilhas espalhadas com dados redundantes, controlados apenas pelas próprias pessoas que as manipulavam. O gestor não tinha uma ideia clara do total de equipamentos do parque instalado e, com isso, não tinha controle da aquisição de equipamentos, da quantidade existente na empresa, nem ainda dos equipamentos com defeito ou obsoletos. Os requisitos foram levantados com base nas necessidades do cliente, por meio de entrevistas com as partes interessadas.

Pressman (2011) aborda o modelo de interações de *WebApps*. Ali, o autor explica as formas que a interface interage com o usuário, normalmente usando diálogos, sendo uma dessas formas a utilização dos casos de uso no projeto. Salienta também a importância de criar o protótipo de interface de usuário, fazendo com que o criador do projeto se pergunte onde será exibido o menu de navegação, os *links* e demais componentes.

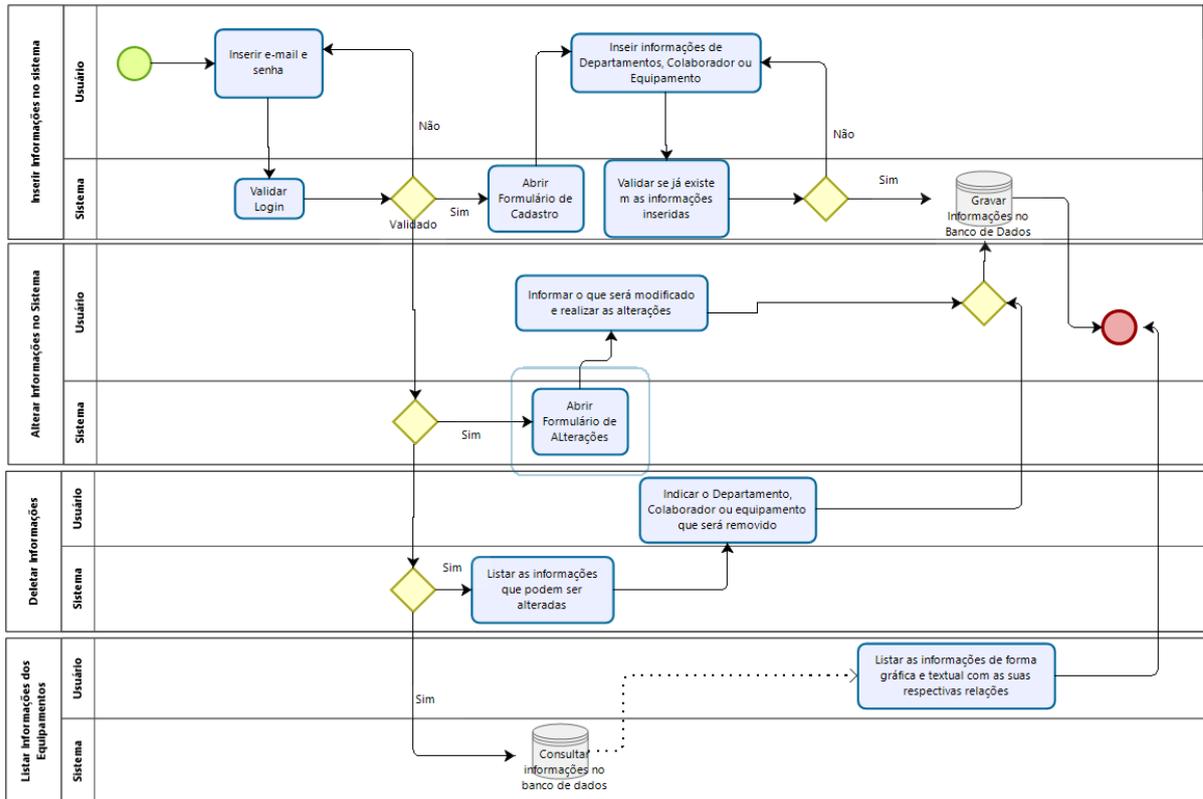
## 2.2 BPMN

Relacionado a fluxo de processos, o BPMN (*Business Process Modelling Notation*) é um diagrama que exemplifica de forma gráfica e textual as sequências de interações de usuário e sistema com as suas respectivas respostas ao acessar as funcionalidades.

Segundo Pressman (2011, p. 182),

O diagrama de fluxo de dados permite que desenvolvamos modelos do domínio de informações e domínio funcional. À medida que o DFD é refinado com níveis de detalhe cada vez maiores, realizamos uma decomposição funcional implícita do sistema. Ao mesmo tempo, o refinamento do DFD resulta em um correspondente refinamento dos dados à medida que se avança nos processos que constituem a aplicação.

Figura 1 – BPMN



Fonte: elaborado pelo autor

### 2.3 Requisitos Funcionais

Os requisitos funcionais trazem as funções individuais que o sistema deve fornecer, assim como, a sua reação ao iniciar tal funcionalidade.

Para Sommerville (2011, p. 59),

São declarações de serviços que o sistema deve fornecer, de como o sistema deve reagir a entradas específicas e de como o sistema deve se comportar em determinadas situações. Em alguns casos, os requisitos funcionais também podem explicitar o que o sistema não deve fazer.

Quadro 1 – Requisitos Funcionais do sistema

<b>RF001</b> - Cadastro de usuário	Categoria: ( ) Oculto (X) Evidente	Prioridade: (X) Altíssima ( ) Alta ( ) Média ( ) Baixa
<b>Descrição:</b> O sistema deve cadastrar o usuário no banco de dados por meio de um formulário para ter acesso às funções.		
<b>RF002</b> - Cadastro de equipamento	Categoria: ( ) Oculto (X) Evidente	Prioridade: (X) Altíssima ( ) Alta

		( ) Média ( ) Baixa
<b>Descrição:</b> O sistema deve cadastrar os equipamentos no banco de dados por meio de um formulário.		
<b>RF003</b> - Cadastro de colaboradores	Categoria: ( ) Oculto (X) Evidente	Prioridade: (X) Altíssima ( ) Alta ( ) Média ( ) Baixa
<b>Descrição:</b> O sistema deve cadastrar os colaboradores no banco de dados por meio de um formulário.		
<b>RF004</b> - Controle de RDM	Categoria: ( ) Oculto (X) Evidente	Prioridade: (X) Altíssima ( ) Alta ( ) Média ( ) Baixa
<b>Descrição:</b> O sistema deve salvar as alterações em uma tabela de histórico após os dados serem inseridos, editados ou deletados.		
<b>RF005</b> - Pesquisa do consolidado dos equipamentos	Categoria: ( ) Oculto (X) Evidente	Prioridade: (X) Altíssima ( ) Alta ( ) Média ( ) Baixa
<b>Descrição:</b> Nessa visão, o sistema deve mostrar todo o consolidado de equipamentos do parque da empresa, separando os que estão em produção e estoque.		
<b>RF006</b> – Geração de relatório	Categoria: (X) Oculto ( ) Evidente	Prioridade: (X) Altíssima ( ) Alta ( ) Média ( ) Baixa
<b>Descrição:</b> O sistema deve permitir a geração de relatórios a partir dos dados inseridos e manipulados.		

## 2.4 Requisitos Não Funcionais

Sommerville (2011) esclarece que, os requisitos não funcionais são restrições às funcionalidades do sistema, as quais podem ser impostas no controle de tempo, no processo de desenvolvimento e por normas. Informa, também, que as restrições se aplicam no sistema como um todo, e não apenas a características individuais.

**Quadro 2** – Requisitos Não Funcionais do sistema

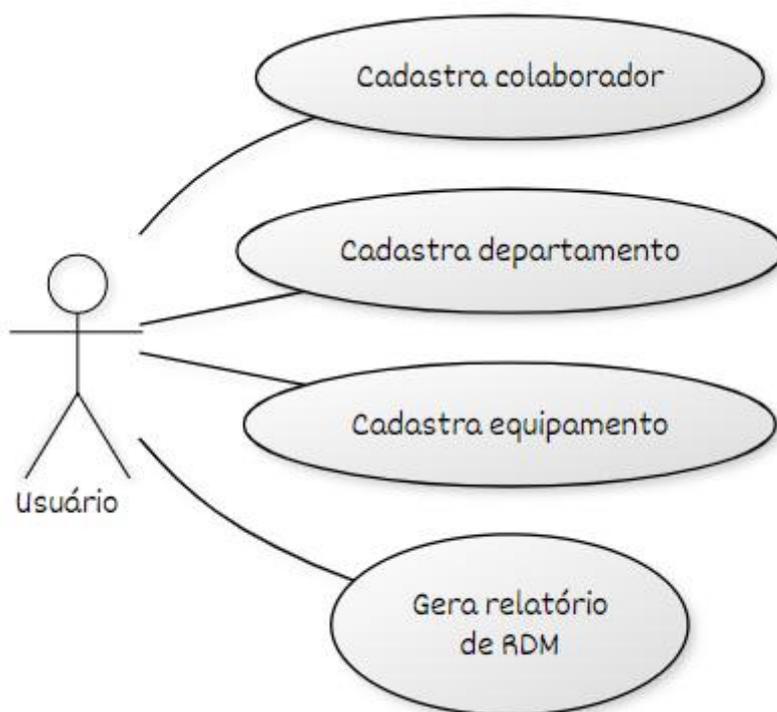
<b>RNF001</b> - Plataforma	O sistema deve funcionar na plataforma <i>web</i> , em navegadores	Tipo	( ) Desejável (X) Obrigatório	(X) Permanente ( ) Transitório
<b>RNF002</b> -Banco de dados	O sistema deve utilizar banco de dados relacional	Tipo	( ) Desejável (X) Obrigatório	(X) Permanente ( ) Transitório
<b>RNF003</b> - Autenticação para realizar login	O sistema deve utilizar um <i>token</i> para validação do <i>login</i> do usuário	Tipo	( ) Desejável (X) Obrigatório	(X) Permanente ( ) Transitório

<b>RNF004-</b> Desenvolvimento	O sistema deve ser desenvolvido em <i>Javascript</i>	Tipo	( ) Desejável (X) Obrigatório	(X) Permanente ( ) Transitório
-----------------------------------	--	------	----------------------------------	-----------------------------------

## 2.5 Casos de Uso

Os casos de uso são representações gráficas e textuais que combinam atores e funções do *software* e as descrevem. Os atores podem ser pessoas, dispositivos ou até outros sistemas que desempenham um papel à medida que interagem com o sistema. Normalmente, no momento da entrevista, o interessado informa uma função específica que deseja ver no *software* e a equipe de levantamento de requisito a transcreve de forma narrativa. Após isso, é transformada em caso de uso (PRESSMAN, 2011).

**Figura 1** - Diagrama de caso de uso



**Fonte:** elaborado pelo autor

**Quadro 3** – Use Case Login

Caso de Uso – Login	
<b>ID</b>	UC 001
<b>Descrição</b>	Este caso de uso tem por objetivo autenticar o usuário para acessar o sistema
<b>Ator Primário</b>	Usuário do sistema

<b>Pré-condição</b>	Estar cadastrado no Banco de dados
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. O <i>use case</i> inicia quando o usuário acessa o sistema e abre a tela de <i>login</i>.</li> <li>2. O sistema carrega o formulário de <i>login</i>.</li> <li>3. O sistema valida se o usuário e senha são válidos.</li> <li>4. O sistema autentica o usuário.</li> </ol>
<b>Pós-condição</b>	Nenhum
<b>Cenário Alternativo</b>	<p>3a – As informações de login estão erradas</p> <p>3a1 O sistema informa que há erro de <i>login</i></p>

**Quadro 4 – Use Case Cadastrar Colaborador**

<b>Caso de Uso – Cadastro de Colaborador</b>	
<b>ID</b>	UC 003
<b>Descrição</b>	Este caso de uso tem por objetivo cadastrar os colaboradores da empresa no sistema
<b>Ator Primário</b>	Usuário do sistema
<b>Pré-condição</b>	Ter permissão para inserir informações
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. O <i>use case</i> inicia quando o usuário seleciona a opção cadastro de colaborador.</li> <li>2. O sistema carrega o formulário de cadastro de colaborador.</li> <li>3. O sistema grava as informações.</li> </ol>
<b>Pós-condição</b>	Nenhum
<b>Cenário Alternativo</b>	<p>3a – O sistema informa que já existe um colaborador com as informações inseridas.</p> <p>3a.1 O sistema mostra o campo onde precisa ser alterado.</p>

**Quadro 5 – Use Case Cadastrar Equipamento**

<b>Caso de Uso – Cadastrar Equipamento</b>	
<b>ID</b>	UC 002
<b>Descrição</b>	Este caso de uso tem por objetivo cadastrar os equipamentos no sistema
<b>Ator Primário</b>	Usuário do sistema
<b>Pré-condição</b>	Ter permissão para inserir informações
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. O <i>use case</i> inicia quando o usuário seleciona a opção cadastro de equipamento.</li> <li>2. O sistema carrega o formulário de cadastro de equipamento.</li> <li>3. O sistema grava as informações.</li> </ol>
<b>Pós-condição</b>	Nenhum
<b>Cenário Alternativo</b>	<p>3a – O sistema informa que já existe um equipamento com os dados inseridos.</p> <p>3a.1 O sistema mostra o campo onde precisa ser alterado.</p>

**Quadro 6 – Use Case Cadastrar Departamento**

<b>Caso de Uso – Cadastrar Departamento</b>	
<b>ID</b>	UC 002
<b>Descrição</b>	Este caso de uso tem por objetivo cadastrar os departamentos no sistema
<b>Ator Primário</b>	Usuário do sistema
<b>Pré-condição</b>	Ter permissão para inserir informações
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. O <i>use case</i> inicia quando o usuário seleciona a opção cadastro de departamento.</li> <li>2. O sistema carrega o formulário de cadastro de departamento.</li> </ol>

	3. O sistema grava as informações.
<b>Pós-condição</b>	Nenhum
<b>Cenário Alternativo</b>	3a – O sistema informa que já existe um departamento com os dados inseridos 3a.1 O sistema mostra o campo onde precisa ser alterado.

**Quadro 7 – Use Case Gerar relatório de RDM**

<b>Caso de Uso – Gerar relatório de RDM (Registro De Mudança)</b>	
<b>ID</b>	UC 004
<b>Descrição</b>	Este caso de uso tem por objetivo consultar os registros de mudança
<b>Ator Primário</b>	Usuário do sistema
<b>Pré-condição</b>	Ter equipamentos e colaboradores cadastrados.
<b>Cenário Principal</b>	1. O caso de uso inicia quando o usuário abre a página de RDM. 2. O sistema gera o relatório de RDM.
<b>Pós-condição</b>	Nenhum
<b>Cenário Alternativo</b>	

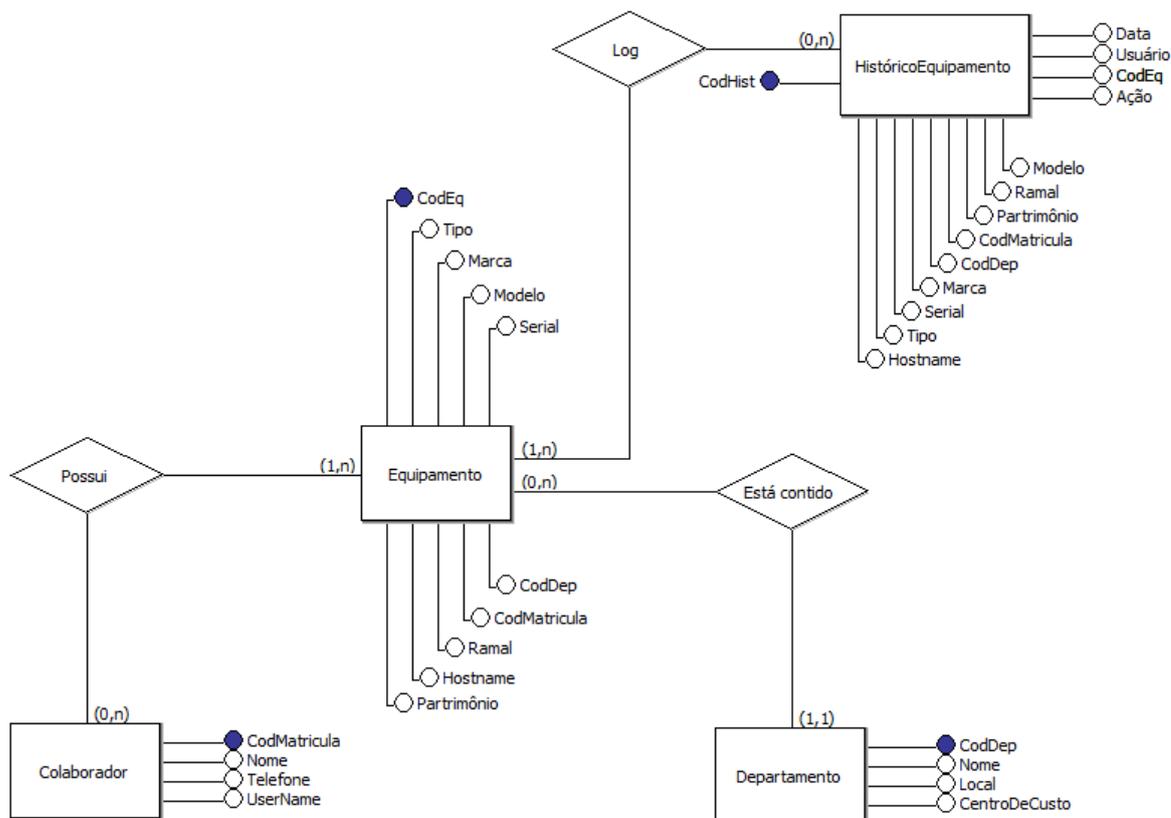
**Quadro 8 – Use Case Consulta consolidado de equipamentos**

<b>Caso de Uso – Fazer consulta do consolidado de equipamentos</b>	
<b>ID</b>	UC 005
<b>Descrição</b>	Este caso de uso tem por objetivo realizar a consulta do consolidado de equipamentos
<b>Ator Primário</b>	Usuário do sistema
<b>Pré-condição</b>	
<b>Cenário Principal</b>	1. O use case inicia quando o usuário seleciona a opção <i>Dashboard</i> . 2. O sistema gera um relatório da quantidade de equipamentos do parque distinguindo principalmente os que estão em produção e no estoque. 3. O sistema mostra um gráfico com a quantidade de equipamentos.
<b>Pós-condição</b>	Nenhum
<b>Cenário Alternativo</b>	

## 2.6 Diagrama Entidade-Relacionamento

Os Diagramas Entidade-Relacionamento ilustram como as entidades (que são todos os objetos do sistema, no nosso caso, o equipamento, colaborador e departamento) se relacionam e definem os seus atributos. Ele é utilizado para o planejamento do banco de dados.

**Figura 2 - Diagrama entidade-relacionamento conceitual**



**Fonte:** elaborado pelo autor

### 3 Ferramentas e Métodos de Desenvolvimento

Nesta seção, são descritos as ferramentas e os métodos de desenvolvimento empregados no decurso do projeto. São enfatizados os aspectos tecnológicos e os princípios de Engenharia de *Software*.

#### 3.1 Ferramentas

Para o desenvolvimento do *software*, atualmente, tem-se a vantagem de utilizar ferramentas que aumentam a produtividade, muito mais do que no início da criação dos computadores e antes da criação da Internet. A seguir, são listadas e descritas as principais ferramentas utilizadas.

O MarvelApp é um sistema *online* para prototipação de interfaces de usuário

para diversas plataformas. Ele conta com uma versão de livre utilização, com funcionalidades básicas, mas suficientes para a necessidade deste projeto (MARVEL, 2020).

O Visual Studio Code é uma ferramenta criada pela Microsoft e lançada em 2015, sob licença MIT. Traz um ambiente de desenvolvimento completo, incorporando suporte a Git e depuração, dentre outras funções. Está disponível para ambientes Windows, Linux e MacOS. Nela, é possível desenvolver em praticamente qualquer linguagem, sendo que algumas delas têm suporte por meio de extensões (MICROSOFT, 2020). Toda a parte de desenvolvimento de código foi realizada nessa ferramenta, com exceção da programação do banco de dados.

O Postman é uma ferramenta utilizada como cliente de API (*Application Programming Interface* – Interface de Programação de Aplicativos). Ele auxilia os desenvolvedores a fazer testes envios de requisições e recebimento de respostas usando o protocolo HTTP (*Hypertext Transfer Protocol* – Protocolo de Transferência de Hipertexto) (POSTMAN, INC, 2020). Durante o desenvolvimento da aplicação, o Postman foi utilizado frequentemente para testes antes de colocar a API REST (*Representational State Transfer* – Transferência de Estado Representacional) em produção na nuvem.

O MySQL é um dos sistemas gerenciadores de banco de dados (SGBD) relacionais mais populares no mundo, tanto em sua versão *open source*, de livre utilização (MySQL Community Server) quanto na sua versão paga (Enterprise Server). O SGBD é desenvolvido em C e C++ e utiliza linguagem SQL (*Structured Query Language* – Linguagem de Consulta Estruturada) (ORACLE, 2020). Por incluir as principais ferramentas de desenvolvimento e gerenciamento de bancos de dados, foi escolhida para integrar o projeto.

O MySQL Workbench é uma ferramenta com interface visual utilizada para modelagem de dados e desenvolvimento em SQL, além do gerenciamento de usuários, servidores, e *backup* do banco de dados. Tem como vantagem a facilidade da geração de código a partir da modelagem, dando produtividade ao DBA (*Database Administrator* – Administrador de Banco de Dados) (ORACLE, 2020). Por meio dela, também foi possível fazer o *deploy* (entrega) do banco de dados para nuvem.

O Node.js é um interpretador de JavaScript baseado no código que realiza a mesma função no navegador Google Chrome, chamado *engine V8*. O emprego da

ferramenta em servidores é recente; no entanto, já é muito utilizado devido ao seu alto poder de escalabilidade e leveza. Está presente em aplicações de grandes corporações, como Netflix e Uber. O Node.js tem como seu principal gerenciador de pacotes o NPM (*Node Package Manager*), que, atualmente (2020) possui o maior número de pacotes da Internet, conferindo produtividade aos desenvolvedores. Dentre tais pacotes, o *framework* Express.js é um dos mais utilizados para aplicações *web* (OPENJS FOUNDATION, 2020).

O Angular, antes conhecido como AngularJS, é um *framework* desenvolvido pela Google, tendo sido a sua primeira versão lançada em 2016. Ele combina as linguagens HTML5 e CSS com suporte a TypeScript, fazendo uso da arquitetura MVC (*Model-View-Controller* – Modelo-Visão-Controle) (GOOGLE, 2020). Sua utilização torna possível criar unidades lógicas denominadas componentes, contribuindo para o desenvolvimento de código de forma limpa e organizada.

### 3.2 Métodos ou Desenvolvimento

Ao se fazer o levantamento de requisitos e analisar as planilhas eletrônicas compartilhadas, foi observado que os dados poderiam ser facilmente corrompidos e que seu processamento era lento. A ideia inicial para o desenvolvimento do sistema partiu da experiência com o manuseio de dados através de um SGBD na disciplina de Banco de Dados, mais precisamente o Microsoft SQL Server. Foram criadas, de início, as tabelas dos principais componentes do sistema e a interligação entre elas. Foi utilizado o conhecimento obtido nas aulas para criação das *triggers*, uma função importantíssima para realização dos RDMs (Registro de Mudanças), para os quais o próprio gerenciador do banco já grava os *logs*. As RDMs também eram feitas, anteriormente, em planilhas eletrônicas compartilhadas, nas quais havia constante sobrescrita das linhas de dados pelos analistas que as utilizavam.

A parte do *back-end*, denominada, Rest API foi desenvolvida em Node.js, utilizando boas práticas de separação de rotas e controladores. Foi hospedado em nuvem em uma conta gratuita do Heroku, juntamente com o banco de dados que foi desenvolvido em MySQL (ClearDB MySQL no Heroku) e modelado no Workbench 8.0.

Por outro lado, o Angular dispõe de um conjunto de componentes já pronta para

o desenvolvimento da parte *front-end* — neste caso, o cliente HTTP — e da parte visual com as suas próprias bibliotecas. A utilização desse *framework* possibilitou o desenvolvimento ágil e funcional dos requisitos demandados pelo projeto. A parte visual resultou em uma visualização fácil e intuitiva dos dados. Foi realizada a prototipação das telas com recursos da plataforma MarvelApp. Para a criação dos gráficos, foi utilizado a biblioteca HighCharts, implementada no Angular com as suas integrações (HIGHCHARTS, 2020).

Para criar o *login*, foi utilizada a tecnologia JWT (*JSON Web Token*). O *token* é composto pelo *e-mail* do usuário, pela senha (criptografada com *bcrypt* usando *hash*) e por uma chave privada encapsulada. Em todas as sessões, o *token* gerado é diferente do anterior. Desse modo, as rotas ficam protegidas através de um *HttpInterceptor* aplicado no Angular, que só permite prosseguir para as outras telas e métodos — GET, POST, PUT, DELETE — quando recebe a resposta positiva da REST API.

#### 4 Resultados e Discussão

Segundo Pressman (2011, p. 287),

O projeto de interfaces do usuário cria um meio de comunicação efetivo entre o ser humano e o computador. Seguindo-se um conjunto de princípios de projeto de interfaces, o projeto identifica objetos e ações de interface e então cria um *layout* de tela que forma a base para um protótipo de interface do usuário.

Foi realizada a prototipação das telas com ferramentas e práticas de *design* modernos, resultando em uma interface funcional e agradável.

A Figura 4 apresenta a tela de *login* para o usuário. Nessa tela, o usuário digita o seu *e-mail* e senha para entrar no sistema e acessar as funções.

**Figura 4** – Tela de *login*

**Fonte:** elaborado pelo autor

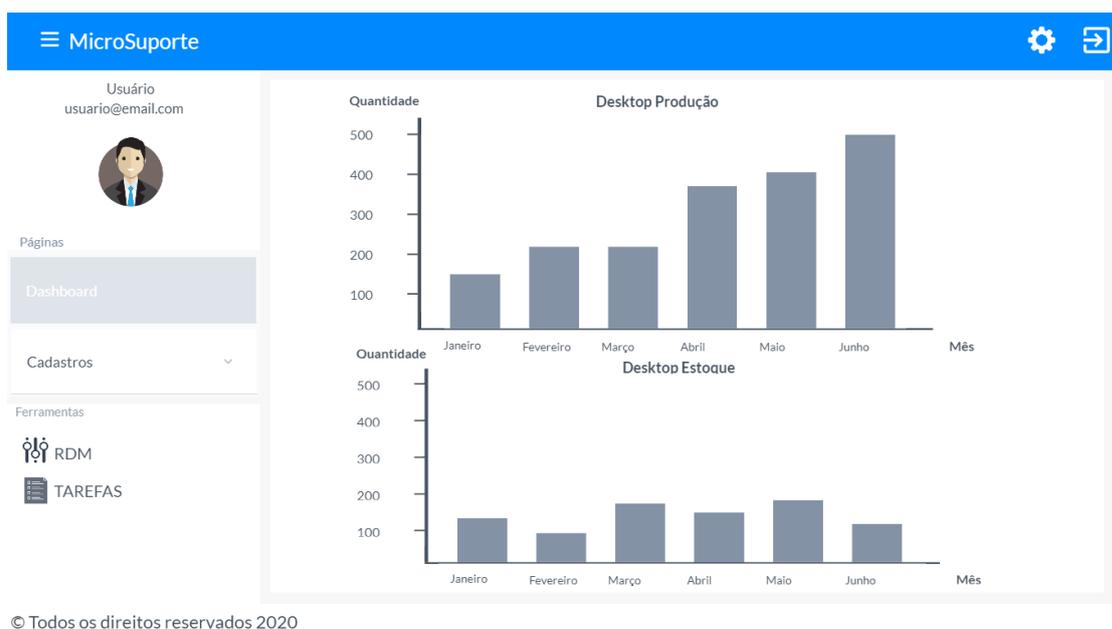
A Figura 5 apresenta a tela de boas-vindas com o menu de funções.

**Figura 5** – Tela de boas-vindas

**Fonte:** elaborado pelo autor

A Figura 6 apresenta um *dashboard* com o consolidado de equipamentos. Nessa visão, o gestor pode acompanhar o histórico da quantidade de equipamentos no parque: total, em produção e em estoque. Com isso, consegue verificar se o estoque é suficiente e se precisa adquirir mais equipamentos.

**Figura 6** – Tela de consolidado de equipamentos



**Fonte:** elaborado pelo autor

Na Figura 7, é apresentada a tela de cadastro de colaboradores. Nela, o usuário consegue cadastrar, excluir, editar e ver a lista de colaboradores já cadastrados. Os dados a serem inseridos são: o código de matrícula do colaborador fornecido pela própria empresa, o nome completo, o nome de usuário e o telefone para contato. Conta, também, com os botões de editar, excluir e salvar.

**Figura 7 – Tela de cadastro de colaboradores**

MicroSuporte

Usuário  
usuario@email.com

Dashboard

Cadastros

Colaborador

Departamento

Equipamento

Lista de Colaboradores

Procurar

Matrícula	Nome	Username	Telefone	Ações
1234	João da Silva	josilva	(12) 1234-5678	
5678	Maria da Silva	msilva	(21) 4321-5678	
9012	Pedro Mendes	pmendes	(16) 5678-1234	
9123	Ana Maria	amaria	(11) 5678-4321	

Cadastrar/Editar

Matrícula Nome User Name Telefone

© Todos os direitos reservados 2020

Fonte: elaborado pelo autor

A Figura 8 apresenta a tela de cadastro de departamentos. Nela é possível cadastrar, editar, excluir e listar os departamentos já cadastrados. Os dados a serem inseridos são: Nome do departamento, local e centro de custo. Também são disponibilizados os botões de editar, excluir e salvar.

**Figura 8 – Tela de cadastro de departamentos**

MicroSuporte

Usuário  
usuario@email.com

Dashboard

Cadastros

Colaborador

Departamento

Equipamento

Lista de Departamentos

Procurar

Código	Nome	Local	Centro de Custo	Ações
2211	TI	1º Andar	3344	
2212	RH	Térreo	3345	
2213	Qualidade	Térreo	3346	
2214	Vendas	2º Andar	3347	

Cadastrar/Editar

Código Nome Local Centro de Custo

© Todos os direitos reservados 2020

Fonte: elaborado pelo autor

Na Figura 9, é apresentada a tela de cadastro de equipamentos. Nela, é possível cadastrar, excluir, editar e listar os equipamentos já cadastrados. Os campos a serem inseridos são: tipo, modelo, marca, ramal ao qual está conectado (já que, na empresa onde foram levantados os requisitos, a rede dos computadores é conectada diretamente em uma tecnologia de telefonia VoIP (*Voice over Internet Protocol*), número de patrimônio, serial, *hostname*, usuário (da lista de colaboradores), departamentos (da lista de departamentos). Também conta com os botões de editar, excluir e salvar.

**Figura 9** – Tela de cadastro de equipamentos

The screenshot shows the 'MicroSuporte' application interface. On the left is a sidebar with a user profile (Usuário: usuario@email.com) and navigation menu items: Dashboard, Cadastros (expanded), Colaborador, Departamento, and Equipamento. The main content area is titled 'Lista de Equipamentos' and features a search bar labeled 'Procurar'. Below the search bar is a table with the following data:

Código	Tipo	Modelo	Marca	Ramal	Patrimônio	Serial	HostName	Usuário	Dep	Ações
4411	Notebook	Think Pad	Lenovo	1234	8800	5k2l	nb8800	osilvaj	TI	[Edit] [Delete]
4412	Desktop	Optiplex 10	Dell	5678	8855	6d3o	dt8855	msilva	RH	[Edit] [Delete]
4413	Thin Client	Wyse 50	Dell	9012	8844	8k0l	tc8844	n/a	RH	[Edit] [Delete]
4414	Monitor	P191	Dell	2345	8846	etf3	n/a	amaria	OP	[Edit] [Delete]

Below the table is a 'Cadastrar/Editar' form with input fields for: Código, Tipo, Modelo, Marca, Ramal, Patrimônio, Serial, HostName, Usuário, and Departamento. A save button is located to the right of the form. At the bottom left, there is a copyright notice: '© Todos os direitos reservados 2020'.

**Fonte:** elaborado pelo autor

A Figura 10 apresenta a tela de RDM (Registro de Mudança). Nela o usuário tem acesso aos *logs* de modificações gravados no banco de dados. Nessa tela são listadas todas as modificações realizadas nas demais telas de cadastro. No lado direito, há a opção de expandir para ver os detalhes do registro.

**Figura 10** – Tela de RDM (Registro de Mudança)

The screenshot displays the 'MicroSuporte' application interface. On the left, there is a user profile for 'usuário@email.com' and a sidebar with navigation options: Dashboard, Cadastros, and Ferramentas (containing RDM and TAREFAS). The main content area is titled 'Registro de Mudança' and features a search bar and a table with columns: Usuário, Tipo, Ação, Data, and Detalhes. The table contains three rows of data. A modal window is open for the first row, showing 'Informações do Equipamento' with a table of fields: Código (1122), Tipo (Notebook), Modelo (Latitude), Marca (Dell), Serial (22kb), Patrimônio (8825), and Hostname (nb8825). Below this, there is an 'OBS:' field with the text 'Equipamento obsoleto'.

Usuário	Tipo	Ação	Data	Detalhes
User1	Equipamento	Deletar	22/05/2020	(i)
User3	Departamento	Alterar	10/03/2020	(i)
User2	Colaborador	Adicionar	15/10/2019	(i)

Código	Tipo	Modelo	Marca	Serial	Patrimônio	Hostname
1122	Notebook	Latitude	Dell	22kb	8825	nb8825

OBS:  
Equipamento obsoleto

© Todos os direitos reservados 2020

**Fonte:** elaborado pelo autor

## Considerações finais

O maior desafio foi unir os conhecimentos adquiridos durante as aulas em um único projeto. Ambas as partes (*back-end* e *front-end*), foram criadas seguindo o aprendizado das aulas e tutoriais da internet. Foi buscado um projeto na internet para se basear. Porém, os encontrados não aplicavam todas as ferramentas planejadas para desenvolver o *software*, servindo, todavia de base para seguir os modelos de projetos mais atuais.

A disciplina de Banco de Dados foi algo que ajudou muito para implementação do *back-end* que está funcionando e podendo ser testada com o Postman, inclusive em nuvem. A disciplina de Segurança da Informação foi inspiradora para construção da API com autenticação JWT e da criação do próprio *software* tendo em vista a segurança dos dados. A disciplina de IHC (Interação Humano-Computador) desempenhou um importante papel para idealizar uma boa interface, trabalhando, inclusive com a prototipação com as ferramentas apresentadas em sala de aula.

O desenvolvimento do projeto fez com que buscasse mais a fundo o conhecimento para desenvolver os códigos, boas práticas, além de ressaltar a importância de fazer uma boa Engenharia de *Software*. A arquitetura do *software*,

incluindo a prototipação serviu para abstrair todas as ideias para criação do sistema final. Os passos futuros serão a criação de um componente de tarefas, a integração completa do *back-end* com o *front-end*, construir uma ferramenta para abertura de chamados com níveis de SLA (*Service Level Agreement* – Acordo de Nível de Serviço) e disponibilizar integralmente os recursos para as organizações.

### Referências

GOOGLE. **ANGULAR**, sd. Disponível em: <https://angular.io/>. Acesso em: 18 maio 2020.

HIGHCHARTS. **Interactive JavaScript Charts for your webpage | Highcharts**, sd. Disponível em: <https://www.highcharts.com/>. Acesso em: 16 maio 2020.

MICROSOFT. **Visual Studio Code**, sd. Disponível em: <https://code.visualstudio.com/>. Acesso em: 16 maio 2020.

MARVEL. **Marvel App**, sd. Disponível em: <https://marvelapp.com/>. Acesso em: 24 maio 2020.

ORACLE. **MySQL Documentation**, sd. Disponível em: <https://dev.mysql.com/doc/>. Acesso em: 16 maio 2020.

OPENJS FOUNDATION. **Node.js**, sd. Disponível em: <https://nodejs.org/en/>. Acesso em: 18 maio 2020.

POSTMAN, INC. **Postman**, sd. Disponível em: <https://www.postman.com/>. Acesso em: 16 maio 2020.

PRESSMAN. Roger S. **Engenharia de Software: Uma Abordagem Profissional**. 7. ed. São Paulo: AMGH, 2011.

SOMMERVILLE. **Engenharia de Software**. 9. ed. São Paulo: Pearson, 2011.