

**DESENVOLVIMENTO DE SOFTWARE PARA O GERENCIAMENTO DE NEGÓCIO,  
VENDAS E ESTOQUE POR MEIO DE RELATORIOS E DASHBOARDS***DEVELOPMENT OF SOFTWARE FOR BUSINESS, SALES AND INVENTORY MANAGEMENT  
THROUGH REPORTS AND DASHBOARDS**DESARROLLO DE SOFTWARE PARA LA GESTIÓN DE NEGOCIOS, VENTAS E INVENTARIO  
MEDIANTE REPORTE Y TABLEROS*

*Abiratan Lopes de Sousa<sup>1</sup> (abi.lopes.sousa@gmail.com)*  
*Ana Elisa da Silva Motta Aguiar<sup>1</sup> (elisamguiar@gmail.com)*  
*Antonio Paulo Garcia de Castro<sup>1</sup> (antonio.paulo9@gmail.com)*  
*Bruno Donizeti da Silva<sup>1</sup> (bruno@fatecguaratingueta.edu.br)*  
*Camila Martinelli Rocha<sup>1</sup> (camila@fatecguaratingueta.edu.br)*  
*Claudemir Santos Pinto<sup>1</sup> (profdemir@yahoo.com.br)*  
*Matheus Borges Tereza<sup>1</sup> (borgesloth@hotmail.com)*

*<sup>1</sup>Faculdade de Tecnologia de Guaratinguetá*

**Resumo**

Este projeto tem como finalidade solucionar um problema recorrente que acontece há tempos dentro do negócio “Donuts 51”. Situado em Guaratinguetá, o estabelecimento desde o princípio utiliza de ferramentas antigas como planilhas físicas (caderno, caneta, calculadora) para organizar e contabilizar suas vendas para de calcular o lucro, o que ocasiona uma demanda muito grande de tempo e esforço, resultando em uma noção rasa e baixo controle quando se compara a algum aplicativo ou ferramenta de gerenciamento digital. Esta pesquisa visa arquitetar uma aplicação que consiga realizar um gerenciamento de todo o setor de vendas de uma maneira simples, de fácil entendimento e manuseio que consiga calcular os gastos, produzir Dashboards com diferentes gráficos contendo todos os dados pertinentes do negócio e montar uma projeção com base nos dados de como serão as futuras vendas dos próximos meses, trimestres, baseado nos dados já cadastrados pelo usuário.

**Palavras-chave:** Aplicação, Gerenciamento, Dashboards.

**Abstract**

This project aims to solve a recurring problem that has been occurring for some time within the “Donuts 51” business. Located in Guaratinguetá, the establishment has since the beginning used old tools such as physical spreadsheets (notebook, pen, calculator) to organize and account for its sales and calculate profit, which causes a very large demand for time and effort, resulting in a shallow notion and low control when compared to any application or digital management tool. This research aims to design an application that can manage the entire sales sector in a simple, easy to understand and handle way that can calculate expenses, produce Dashboards with different graphs containing all pertinent business data and set up a projection with based on data on what future sales will be like in the coming months and quarters, based on data already registered by the user.

**Keywords:** Application, Management, Dashboards

**Resumen**

Este proyecto pretende solucionar un problema recurrente que viene ocurriendo desde hace algún tiempo dentro del negocio de “Donuts 51”. Ubicado en Guaratinguetá, el establecimiento utiliza desde sus inicios herramientas antiguas como hojas de cálculo físicas (cuaderno, bolígrafo, calculadora) para organizar y contabilizar sus ventas y calcular ganancias, lo que provoca una demanda muy grande de tiempo y esfuerzo, resultando en una escasa noción y bajo control en comparación con cualquier aplicación o herramienta de gestión digital. Esta investigación tiene

como objetivo diseñar una aplicación que pueda gestionar todo el sector de ventas de una manera simple, fácil de entender y manejar, que pueda calcular gastos, producir Dashboards con diferentes gráficos que contengan todos los datos comerciales pertinentes y configurar una proyección basada en datos sobre lo que Las ventas futuras serán como las de los próximos meses y trimestres, en base a los datos ya registrados por el usuario.

**Palabras clave:** Aplicación, Gestión, Tableros.

## Introdução

Na sociedade contemporânea, o avanço tecnológico permitiu que seu uso se conectasse diretamente nas dimensões sociais, profissionais e afetivas. No âmbito dos negócios, tais conexões não poderiam ser diferente, e seus desdobramentos ligados aos impactos desses avanços se efetiva como processo estrutural e de resignificação sociocultural com impactos diretamente ligados no meio empresarial. Neste contexto, os aparelhos tecnológicos dirigem nossas atividades e condicionam o pensar, o agir, o sentir, o raciocínio bem como mediam as relações com as pessoas (DORIGONI, 2007).

Desta forma, com o advento das novas Tecnologias da Informação e Comunicação (TICs), cria-se um complexo contexto de inter-relações e interação comunicativa que por sua vez alteram o modo de ser e de agir das pessoas, ao mesmo tempo em que influenciam a maneira de abordar atividades ligadas ao ramo do negócio. Segundo Kenski (2003), “velocidade” e “tempo” são termos-sínteses do conhecimento no mundo contemporâneo. Aprimorar qualquer tipo de atividade quando se trata de aumentar a eficiência e produtividade de um negócio é sempre bem-vindo, afinal “tempo é dinheiro”.

O uso de celular e aparelhos outros tecnológicos estão inerentes na vida de todos, então usá-los para aumentar o rendimento faz total sentido. Aproveitar de uma tecnologia poderosa e robusta como um computador para acessar uma aplicação que ajude na gestão do seu negócio, já é uma realidade presente nas maiores, e, em uma parcela significativa das empresas, das mais variadas possíveis ao redor do mundo. Usar de cadernos e meios físicos para controle é quase que um retrocesso, seja por falta de conhecimento por parte do usuário, por receio devido a interfaces pouco amigáveis e até mesmo por sistemas pouco intuitivos.

Este projeto tem, portanto, a finalidade de arquitetar um sistema, inicialmente web, em que o usuário consiga gerir seu negócio de uma forma ampla, controlar seu estoque em um nível geral, controlar suas vendas, gerar relatórios, e ter a opção de gerar Dashboards com diferentes gráficos de acordo com a necessidade da análise e filtrar as informações conforme a necessidade e gosto pessoal, aumentando significativamente o nível de detalhe do negócio.

Pensando na parte da integridade das informações, ao inserir e armazená-las em um banco de dados hospedado em um servidor, é fácil de afirmar que há uma discrepância entre uma gestão com aparatos físicos como cadernos e planilhas ao se comparar com dados em nuvem.

Além da velocidade e praticidade ao manusear grandes volumes de dados no meio digital, é mais fácil fazer backups e filtrar as informações do que com algum outro tipo de gerenciamento.

## Desenvolvimento

No primeiro momento, o projeto possuiu um caráter mais exploratório, então, se tratou de uma pesquisa de natureza bibliográfica e de campo. Foram utilizadas técnicas de pesquisas por meio de vídeos e artigos a respeito de quais tecnologias de programação usar, para que melhor atendesse as necessidades na parte a posteriori do desenvolvimento, fase essa que se encontra o projeto atualmente.

A partir das pesquisas realizadas, foi decidido quais linguagens de programação e banco de dados que será utilizado. Das diversas opções que o mercado e a internet disponibilizam, optamos por seguir com o MERN Stack (*MongoDB, Express, React, Node*), pois para o projeto era o que mais atendia as necessidades.

Durante esse processo de criação do software, foi levado em consideração o levantamento de requisitos do sistema, de maneira a entender as necessidades do cliente em busca de uma melhor arquitetura das funcionalidades do projeto. Esses requisitos podem ser separados em funcionais, que descrevem as funcionalidades do sistema, e os não funcionais, que tratam da qualidade do software (SOMMERVILLE, 2011, p. 18).

Um pequeno esboço já foi criado de base no que tange a programação em scripts e banco de dados, como o projeto se encontra em fase de desenvolvimento, os códigos são passíveis a mudanças, mas pelo que já foi desenvolvido, a linha a ser seguida continuará bem próxima do que já se foi produzido de controladores de usuário (Apêndice E), e na criação de um usuário na criação do banco de dados (Apêndice F).

A metodologia adotada pelo grupo para a inicialização do desenvolvimento do código foi a XP (*Extreme Programming*), em que foi dividido e criado um fluxo de trabalho em várias etapas com o intuito de obter um grau de excelência e produtividade maior, devido a uma fácil validação do código já fragmentado e uma flexibilidade maior ao se falar de prazos e metas a serem entregues.

## Considerações Finais

Como o projeto encontra-se em fase de desenvolvimento, ainda está em processo de construção, levando sempre em conta toda a parte teórica das matérias usadas como uma fundação para ele. Utilizando da metodologia XP, o grupo está programando em par, um computador para dois programadores, e está mostrando-se muito promissor esse tipo de prática.

Na parte da engenharia por de trás do software, foi desenvolvido a parte do diagrama de caso de uso (Apêndice A), diagrama de atividade (Apêndice B), Descrições de caso de uso,

diagrama de classe (Apêndice C) e requisitos funcionais e não funcionais (Apêndice D) dos quais foi usado de base para a parte de modelagem do banco de dados. Todos os processos e informações da engenharia estão sendo levados em consideração para com todas as etapas de programação.

Em síntese, é necessário que os softwares sigam normas e padrões durante seu desenvolvimento para melhor performance e manutenção, sendo essencial o conhecimento das etapas prescritas pela engenharia de software para a criação de um bom sistema. Aspectos esses que foram e estão sendo utilizados em todas as *sprints* que designadas.

Todo o conteúdo pensado, planejado e produzido gira em torno do objetivo de desenvolver um sistema intuitivo, com uma boa maneabilidade do software ao tratar dos dados e observá-los em forma de Dashboards, o que se espera ser o diferencial da aplicação em relação aos demais. Embora a sprint não se encontre na fase da produção da Dashboard, todo o caminho até ela já está sendo construído com o objetivo claro.

## Referências

CARVALHO, José Oscar. **O papel da interação humano-computador na inclusão digital.**

Campinas, 2003. Disponível em:

<https://www.scielo.br/j/tinf/a/Swf9dHT3KPYS6WgnSgz9btG/?lang=pt&format=pdf>. Acesso em: 17 out. 2022.

DORIGONI, G. M. L.; SILVA, J. C. **Mídia e Educação: o uso das novas tecnologias no espaço escolar.** [2007]

KENSKI, Vani Moreira. **Tecnologias e ensino presencial e a distância.** Campinas, SP: Papyrus, 2003.

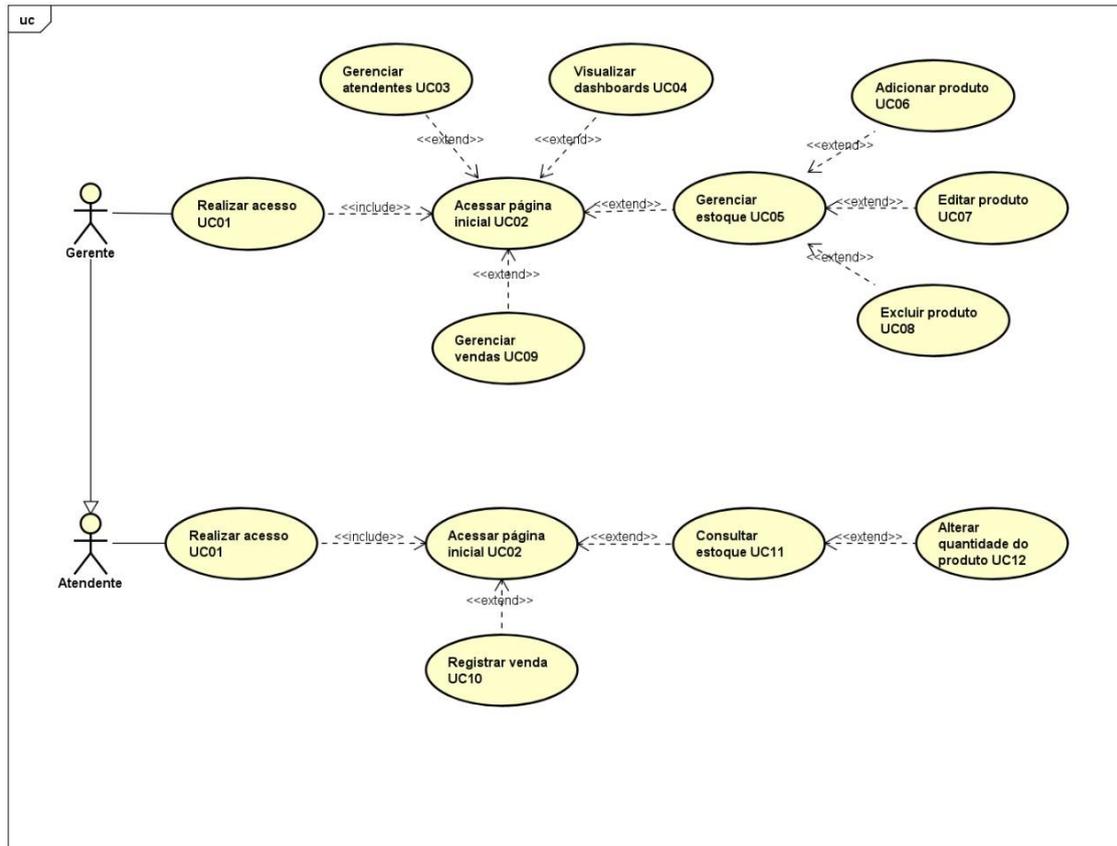
SOMMERVILLE, Ian. **Engenharia de Software**, 9. Ed. São Paulo: Pearson Prentice Hall, 2011.

Disponível em: <https://www.facom.ufu.br/~william/Disciplinas%202018-2/BSI-GSI030-EngenhariaSoftware/Livro/engenhariaSoftwareSommerville.pdf>. Acesso em: 17 out. 2022.

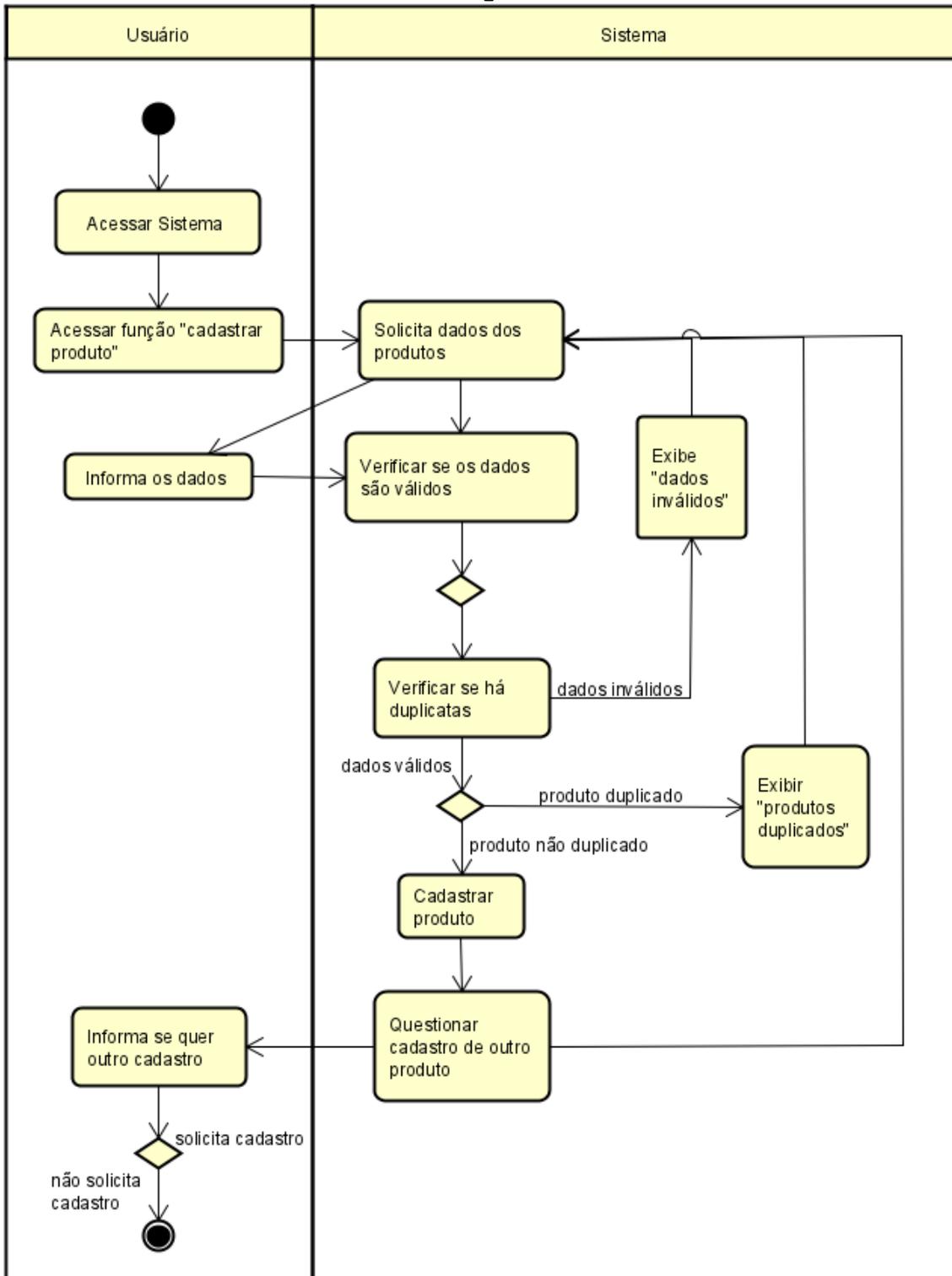
VALENTE, Marco Tulio. **Engenharia de Software Moderna: Princípios e práticas para desenvolvimento de software com produtividade.** Minas Gerais, s.d. Disponível em:

<https://engsoftmoderna.info/>. Acesso em: 17 out. 2022.

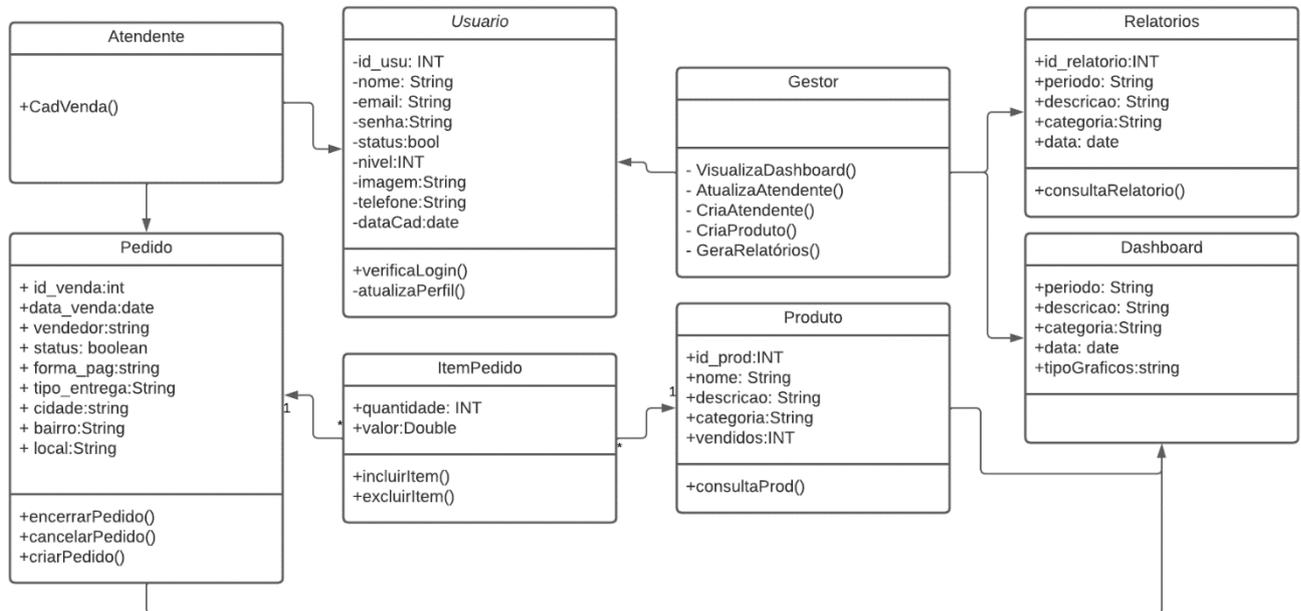
### APÊNDICE A - Diagrama de caso de uso



APÊNDICE B - Diagrama de atividade



### APÊNDICE C - Diagrama de classe



## APÊNDICE D - Requisitos funcionais e não funcionais

Código	Descrição
RF01	O sistema deve possuir mecanismo de login, com dois níveis de usuários: Gerente e atendente.
RF02	O sistema deve possibilitar cadastro de vendas de maneira intuitiva, aos usuários: Gerente e atendente
RF03	O sistema deve possibilitar cadastro de produtos de maneira intuitiva, aos usuários: Gerente e atendente
RF04	O sistema deve possibilitar cadastro de matéria prima no estoque de maneira intuitiva, aos usuários: Gerente e atendente
RF05	O sistema deve possibilitar cadastro de usuário de maneira intuitiva, ao usuário: Gerente
RF06	O sistema deve permitir correção e remoção de dados inseridos incorretamente, mediante aprovação do gerente.
RF07	O sistema deve fornecer uma tabela para visualização de vendas, produtos, estoque e usuários
RF08	As tabelas de visualização de vendas, produtos, estoque e usuários devem possuir filtros de categoria bem como de ordenação
RF09	O sistema deve possuir uma tela “Dashboard” que exiba gráficos de acordo com a preferência do usuário
RF10	O sistema deverá permitir ao gerente realizar levantamentos estatísticos
RF11	O sistema deverá permitir ao gerente gerar relatórios diários, semanais, mensais e anuais.
RF12	O sistema deverá permitir ao usuário alterar sua senha
RF13	O sistema deverá permitir ao usuário alterar suas informações da conta

### Requisitos não funcionais

Código	Descrição
RNF01	O sistema deverá ser implementado na Web
RNF02	O sistema deverá possuir design responsivo, que se adapte aos diferentes tamanhos de dispositivos

RNF03	O sistema deverá ser desenvolvido utilizando linguagens de programação web
RNF04	O sistema deverá ser desenvolvido utilizando linguagens de marcação web
RNF05	O sistema deverá carregar página em menos de três segundos.
RNF06	O sistema deverá utilizar-se de poucas ou nenhuma informação sensível, adequando-se a LGPD.
RNF07	O sistema deverá ser desenvolvido utilizando a metodologia ágil Extreme Programming (XP)
RNF08	O sistema deverá possuir diretiva para consumo de dados em API's

## APÊNDICE E – Controladores de usuários

```
const User = require('../models/User')
const asyncHandler = require('express-async-handler')
const bcrypt = require('bcrypt')

const getAllUsers = asyncHandler(async (req, res) => {
  const users = await User.find().select('-usu_senha').lean()
  if (!users?.length) {
    return res.status(400).json({message: 'Nenhum usuário encontrado.'})
  }
  res.json(users)
})

const postNewUser = asyncHandler(async (req, res) => {
  const {usu_nome, usu_email, usu_senha, usu_telefone} = req.body

  if(!usu_nome || !usu_senha || !usu_email.length || !usu_telefone.length) {
    return res.status(400).json({message: 'Preencha todos os campos.'})
  }

  const duplicate = await User.findOne({usu_email}).lean().exec()
  if (duplicate) {
    return res.status(409).json({message: 'Email já cadastrado.'})
  }

  const hashedPwd = await bcrypt.hash(usu_senha, 10)

  const userObject = {usu_nome, "usu_senha": hashedPwd, usu_email, usu_telefone}

  const user = await User.create(userObject)

  if (user) {
    res.status(201).json({message: `Novo usuário ${usu_nome} criado.`})
  } else {
    res.status(400).json({message: 'Usuário inválido.'})
  }
})

const patchUser = asyncHandler(async (req, res) => {
  const {id, usu_nome, usu_email, usu_senha, usu_telefone} = req.body

  if (!usu_nome || !usu_email.length) {
    return res.status(400).json({message: 'Preencha todos os campos'})
  }

  const user = await User.findOne(id).exec()

  if(!user) {
    return res.status(400).json({message: 'Usuário não encontrado.'})
  }

  const duplicate = await User.findOne({usu_email}).lean().exec()
  if (duplicate && duplicate?._id.toString !== id) {
    return res.status(409).json({message: 'Email já cadastrado.'})
  }
})
```

```
}  
  
user.usu_nome = usu_nome  
user.usu_email = usu_email  
user.usu_telefone = usu_telefone  
  
if (usu_senha) {  
  user.usu_senha = await bcrypt.hash(usu_senha, 10)  
}  
  
const updatedUser = await user.save()  
  
res.json({message: `${updatedUser.usu_email} atualizado.`})  
})  
  
const deleteUser = asyncHandler(async (req, res) => {  
  const {usu_email} = req.body  
  
  if(!usu_email) {  
    return res.status(400).json({message: 'O email precisa ser válido.'})  
  }  
  
  const user = await User.findOne({usu_email}).exec()  
  
  if (!user) {  
    return res.status(400).json({message: 'Usuário não encontrado.'})  
  }  
  
  const result = await user.deleteOne()  
  
  const reply = `Usuário ${result.usu_nome} com o email ${result.usu_email} deletado.`  
  
  res.json(reply)  
})  
  
module.exports = {  
  getAllUsers,  
  postNewUser,  
  patchUser,  
  deleteUser  
}
```

**APÊNDICE F – Criação de um usuário no banco de dados**

```
const mongoose = require('mongoose')
const AutoIncrement = require('mongoose-sequence')(mongoose)

const userSchema = mongoose.Schema (
  {
    usu_nome: {
      type: String,
      required: true,
      trim: true
    },
    usu_email: {
      type: String,
      required: true,
      trim: true,
      validate(value) {
        if (!value.match(/^[^@ ]+@[^@ ]+.[^@ ]{2,}$/) ) {
          throw new Error('O email não é válido.');        }
      }
    },
    usu_senha: {
      type: String,
      required: true,
      trim: true,
      minlength: 8
    },
    usu_status: {
      type: Boolean,
      default: true
    },
    usu_telefone: {
      type: String,
      required: true,
      trim: true,
      maxlength: 13,
      minlength: 9
    }
  }
)

userSchema.plugin(AutoIncrement, {
  inc_field: 'id',
  start_seq: 1
})

module.exports = mongoose.model('User', userSchema)
```