



FACULDADE DE TECNOLOGIA ALFA DE UMUARAMA

Autorizada pela Portaria Nº. 1.390 de 14 de novembro de 2008
Rua Desembargador Antônio Franco Ferreira da Costa nº. 3678 - Umuarama – PR
Fone: (44) 3622-2562 CNPJ: 81236267/0001-84

**Desenvolvimento do Trabalho de Conclusão de Curso de
Sistemas para Internet**

RADAR 4.0

HÉLISON RODRIGUES DE SOUZA

**ORIENTADOR:
MARCIO ALVES DA COSTA**

Umuarama
2020

SISTEMA RADAR 4.0

HÉLISON RODRIGUES DE SOUZA

Relatório Final do Trabalho de Conclusão de Curso (TCC), apresentado ao Curso de Tecnologia em Sistemas para Internet, na Faculdade Alfa de Umuarama, como requisito para a obtenção da certificação de Tecnólogo em Sistemas para Internet.

Umuarama
2019

RESUMO

O trabalho a seguir é o desenvolvimento de um sistema denominado Radar 4.0, que gerencia informações de linhas de produção de indústria de manufatura. Esta aplicação é baseada em um gateway de campo, que coleta os dados de um conjunto de sensores que identifica e valida o fluxo de produção de uma linha. Estes dados são transmitidos em tempo real ao servidor da aplicação que disponibiliza aos usuários relatórios e gráficos detalhados sobre a execução da programação do dia.

Resumo da parte de introdução

O sistema disponibiliza as telas de cadastros de usuários, de cadastro de programação, os relatórios de produção, a estrutura do banco de dados e a implementação dos servidores de aplicação e de dados.

Palavras-Chaves: OEE, agendamento, controle, relatórios, horários.

LISTA DE FIGURAS

Figura 1: Protótipo de formulário de consulta de Produtos.....	15
Figura 2: Protótipo de lista de relatório de Produtos	16
Figura 3: Protótipo de formulário de consulta de Pausas.....	16
Figura 4: Protótipo de lista de relatório de Pausas.....	17
Figura 5: Protótipo de formulário de consulta de Produção.....	18
Figura 6: Protótipo de lista de relatório de Produção.....	19
Figura 7: Protótipo de formulário de consulta de Paradas.....	20
Figura 8: Protótipo de lista de relatório de Paradas.....	20
Figura 9: Protótipo de formulário de consulta de Motivos.....	21
Figura 10: Protótipo lista de relatório de motivo.....	21
Figura 11: Protótipo de formulário de consulta de Usuários.....	22
Figura 12: Protótipo de lista de relatório de Usuários.....	23
Figura 13: Modelo do diagrama de entidade e relacionamento do sistema.....	24
Figura 14: Diagrama de Caso de Uso.....	25
Figura 15: Protótipo Login de Usuário.....	27
Figura 16: Protótipo Página Home.....	29
Figura 17: Protótipo tabela de motivos cadastrados.....	31
Figura 18: Protótipo formulário de cadastro de motivo de parada.....	31
Figura 19: Protótipo formulário de edição de motivo de parada.....	31
Figura 20: Protótipo lista de paradas com motivo não identificado.....	33
Figura 21: Protótipo lista de motivo para atribuir a parada selecionada.....	33
Figura 22: Protótipo tabela de produções cadastradas.....	36
Figura 23: Protótipo formulário de cadastro de produção.....	36
Figura 24: Protótipo formulário de edição de produção.....	37
Figura 25: Protótipo formulário de atribuição de pausas a produção.....	37
Figura 26: Protótipo tabela de produtos cadastrados.....	39
Figura 27: Protótipo formulário de cadastro de produtos.....	40
Figura 28: Protótipo formulário de edição de produto.....	40
Figura 29: Protótipo tabela de pausas cadastradas.....	42
Figura 30: Protótipo formulário de cadastro de pausa.....	43
Figura 31: Protótipo formulário de edição do cadastro de pausa.....	43
Figura 32: Protótipo tabela de usuários cadastrados.....	45

Figura 33: Protótipo formulário de cadastro de usuário.....	46
Figura 34: Protótipo formulário de edição do cadastro de usuário.....	46
Figura 35: Diagrama de Atividades Gerenciar Motivos.....	48
Figura 36: Diagrama de Atividades Gerenciar Paradas.....	49
Figura 37: Diagrama de Atividades Gerenciar Produção.....	50
Figura 38: Diagrama de Atividades Gerenciar Produtos.....	51
Figura 39: Diagrama de Atividades Gerenciar Pausas.....	52
Figura 40: Diagrama de Atividades Gerenciar Usuários.....	53
Figura 41: Console de falha do teste 01.....	54
Figura 42: Trecho de código da correção do teste 01.....	54
Figura 43: Trecho de código do erro do teste 02.....	55
Figura 44: Console do erro do teste 02.....	55
Figura 45: Trecho de código da correção do erro do teste 02.....	56
Figura 46: Trecho de código do erro do teste 03.....	57
Figura 47: Trecho de código da correção do erro do teste 03.....	57
Figura 48: Trecho de código do erro do teste 04.....	57
Figura 49: Trecho de código da correção do erro do teste 04.....	56
Figura 50: Trecho de código do erro do teste 05.....	57
Figura 51: Trecho de código da correção do erro do teste 05.....	57

LISTA DE ABREVIATURAS

UML	Linguagem de Modelagem Unificada - <i>Unified Modeling Language</i>
PHP	<i>Hypertext Preprocessor</i>
IBGE	Instituto Brasileiro de Geografia e Estatística
JS	<i>JavaScript</i>
ECMA	<i>European Computer Manufacturers Association</i>
DOM	<i>Document Object Model</i>
SQL	Linguagem de Consulta Estruturada - <i>Structure Query Language</i>
XML	<i>Extensible Markup Language</i>
AJAX	JavaScript e XML Assíncronos - <i>Asynchronous JavaScript and XML</i>
PCP	Programação e Controle de Produção
RAM	Memória de acesso aleatório - <i>Random Access Memory</i>
OEE	Eficácia geral do equipamento - <i>Overall Equipment Effectiveness</i>

SUMÁRIO

1 INTRODUÇÃO.....	9
2 DESCRIÇÃO DO SISTEMA PROPOSTO.....	9
2.1 Recursos de Software.....	10
2.1.1 JavaScript.....	10
2.1.2 Node.js.....	11
2.1.3 Socket-io.....	11
2.1.4 Express.....	12
2.1.5 Mysql.....	12
2.1.5 Sequelize.....	12
2.2 Recursos de Hardware.....	12
2.3 Descrição dos Cadastros.....	13
2.3.1 Cadastro de usuários.....	13
2.3.2 Cadastro de produção.....	13
2.3.3 Cadastro de produto.....	13
2.3.4 Cadastro pausas.....	13
2.3.5 Cadastro de parada.....	14
2.3.6 Cadastro de motivo.....	14
2.4 Descrição dos Processos / Movimentos.....	14
2.4.1 Gerenciamento de Produção.....	14
2.5 Descrição dos Relatórios.....	15
2.5.1 Descrição do relatório de Produtos.....	15
2.5.2 Descrição do relatório de Pausas.....	16
2.5.3 Descrição do relatório de Produções.....	17
2.5.4 Descrição do relatório de Paradas.....	19
2.5.5 Descrição do relatório de Motivos.....	21
2.5.6 Descrição do relatório de Usuários.....	22
3 DIAGRAMA ENTIDADE-RELACIONAMENTO.....	24
4 CASOS DE USO.....	25
4.2 ESPECIFICAÇÃO DE CASO DE USO: EFETUAR LOGIN.....	26
4.2.1 Descrição.....	26
4.2.2 Fluxo básico.....	26
4.2.3 Fluxos alternativos.....	26
4.2.4 Precondições.....	26
4.2.5 Pós-condições.....	26
4.2.6 Pontos de inclusão ou extensão.....	27
4.2.7 Dicionário de dados.....	27
4.2.8 Protótipo de interface.....	27
4.3 Especificação de Caso de Uso: Visualizar Indicadores.....	28

4.3.1 Descrição.....	28
4.3.2 Fluxo básico.....	28
4.3.3 Fluxos alternativos.....	28
4.3.4 Precondições.....	28
4.3.5 Pós-condições.....	28
4.3.6 Pontos de inclusão ou extensão.....	28
4.3.7 Dicionário de dados.....	28
4.3.8 Protótipo de interface.....	29
4.4 Especificação de Caso de Uso: Gerenciar Motivos.....	29
4.4.7 Dicionário de dados.....	30
4.4.8 Protótipo de interface.....	31
4.5 Especificação de Caso de Uso: Gerenciar Paradas.....	32
4.5.7 Dicionário de dados.....	33
4.4.8 Protótipo de interface.....	33
4.5 Especificação de Caso de Uso: Gerenciar Produção.....	34
4.5.7 Dicionário de dados.....	35
4.5.8 Protótipo de interface.....	36
4.6 Especificação de Caso de Uso: Gerenciar Produtos.....	38
4.6.7 Dicionário de dados.....	39
4.6.8 Protótipo de interface.....	39
4.7 Especificação de Caso de Uso: Gerenciar Pausas.....	41
4.7.7 Dicionário de dados.....	42
4.7.8 Protótipo de interface.....	42
4.7 Especificação de Caso de Uso: Gerenciar Usuários.....	44
4.7.7 Dicionário de dados.....	45
4.7.8 Protótipo de interface.....	45
5 DIAGRAMA DE ATIVIDADES.....	47
6 RELATÓRIO DE TESTES.....	54
7 CONSIDERAÇÕES FINAIS.....	58

1 INTRODUÇÃO

De acordo com números do IBGE, nos dois primeiros meses do ano a produção industrial paranaense cresceu 10,3%. Foi o melhor desempenho do Brasil. Para comparar, o segundo melhor estado neste recorte, Rio Grande do Sul, teve alta de 6,7%. O resultado é bom também no acumulado dos 12 meses (de março de 2018 a fevereiro de 2019): 3,4% de aumento, um dos cinco melhores resultados entre os estados no período [1].

Atualmente a coleta de dados na maioria das indústrias é manual e de baixa segurança. São tratados hoje, dados da produção de ontem, impossibilitando corrigir a produção já perdida. A transformação digital é um conceito que expressa a necessidade de uma empresa de utilizar tecnologias para auxiliar, automatizar ou monitorar processos que agregam valor ao negócio.

O sistema Radar 4.0 será o elo de ligação para dar visibilidade às ações de uma linhas de produção, entregando aos gestores dados seguros em tempo real, possibilitando assim a tomada de decisão para corrigir erros que impactam diretamente no resultado da organização.

Desta forma a produtividade, qualidade, performance e disponibilidade da produção será apontada e as falhas, desvios e perdas de tempo serão monitoradas e visualizadas para que gestão tome ações corretivas e preventivas para aumentar a eficiência da linha de produção.

Este trabalho está dividido da seguinte forma: no Capítulo 2 é apresentada a Descrição do Sistema Proposto com todos os itens que farão parte do sistema, no Capítulo 3 será apresentado o diagrama entidade-relacionamento que compõe a documentação do sistema, no Capítulo 4 são apresentados os requisitos do projeto através dos Casos de Uso do sistema desenvolvido. No Capítulo 5 são apresentados os Diagramas de Atividades com a sequência de atividades que o sistema executa, no Capítulo 6 é apresentado o relatório de testes executados para verificar a conformidade e a qualidade do sistema, e finalmente no Capítulo 7 são apresentadas as conclusões e considerações finais.

2 DESCRIÇÃO DO SISTEMA PROPOSTO

O sistema Radar 4.0 coleta dados da linha de produção e transmite a qualquer dispositivo conectado à internet em tempo real. Com indicadores de gestão a equipe de chão de fábrica é direcionada a corrigir falhas e recuperar a produção do turno, garantindo as metas do mês. As lideranças terão em mãos os resultados de cada linha de produção e o histórico de ações podendo verificar os erros atuais para evitar erros futuros.

2.1 Recursos de Software

O sistema será desenvolvido utilizando a linguagem de programação JavaScript e Banco de Dados MySQL. Tecnologias de desenvolvimento como Socket-io, NodeJS, Express, Socket-io e Sequelize também poderão ser implementadas.

2.1.1 JavaScript

JavaScript (JS) é uma linguagem de programação utilizada principalmente no lado do cliente para scripts de páginas web dinâmicas, mas também é utilizado no lado do servidor.

Concebida como uma linguagem server-side por Brendan Eich (então funcionário da Corporação Netscape), Javascript logo chegou ao Navegador Netscape 2.0 em Setembro de 1995. JavaScript foi um sucesso imediato e o Internet Explorer 3.0 introduziu suporte ao JavaScript sob o nome JScript em Agosto de 1996.

Em Novembro de 1996, a Netscape começou a trabalhar com a ECMA para tornar o JavaScript um padrão da indústria. Desde então, o JavaScript padronizado é chamado de ECMAScript e é especificado em ECMA-262. A oitava e mais recente versão do padrão foi disponibilizada em Junho de 2017 com o nome ECMAScript 2017.

JavaScript é usado principalmente no navegador, permitindo que desenvolvedores manipulem o conteúdo da página web através do DOM, manipulem dados com AJAX e IndexedDB, desenhem gráficos com canvas, interajam com o dispositivo que executa o navegador através de várias APIs, e

assim por diante. JavaScript é uma das linguagens de programação mais utilizadas no mundo, devido ao recente crescimento e melhoria de desempenho das APIs disponíveis nos navegadores.

JavaScript é uma linguagem de programação que permite implementar funcionalidades mais complexas em páginas web. A cada momento uma página web faz mais do que apenas mostrar informações estáticas para você - elas mostram em tempo real conteúdos atualizados, ou mapas interativos, animações gráficas em 2D/3D, vídeos, etc., você pode apostar que o Javascript provavelmente está envolvido [2].

2.1.2 Node.js

O Node.js pode ser definido como um ambiente de execução Javascript server-side. Isso significa que com o Node.js é possível criar aplicações Javascript para rodar como uma aplicação standalone em uma máquina, não dependendo de um browser para a execução, como estamos acostumados.

Apesar de recente, o Node.js já é utilizado por grandes empresas no mercado de tecnologia, como Netflix, Uber e LinkedIn.

O principal motivo de sua adoção é a sua alta capacidade de escala. Além disso, sua arquitetura, flexibilidade e baixo custo, o tornam uma boa escolha para implementação de Microsserviços e componentes da arquitetura Serverless. Inclusive, os principais fornecedores de produtos e serviços Cloud já têm suporte para desenvolvimento de soluções escaláveis utilizando o Node.js.[3].

2.1.3 Socket-io

O Socket.IO oferece uma API de JavaScript simples, baseada em eventos que te permite comunicar entre o servidor e o cliente sem esforço e em tempo real. Seu mecanismo padrão é o WebSockets, mas se não for implementado no navegador do usuário, ele recorrerá a fallbacks, como Flash e AJAX. Isso o torna disponível para um grande número de navegadores.

A biblioteca Socket.IO convencional é escrita em JavaScript tanto para front end, quanto para back-end, por isso foi projetada para rodar em um servidor Node.js [5].

2.1.4 Express

O Express é um framework para aplicativo da web do Node.js mínimo e flexível que fornece um conjunto robusto de recursos para aplicativos web e mobile. Com uma miríade de métodos utilitários HTTP e middleware a seu dispor, criar uma API robusta é rápido e fácil.

O Express fornece uma camada fina de recursos fundamentais para aplicativos da web, sem obscurecer os recursos do Node.js [8].

2.1.5 Mysql

MySQL é um sistema de gerenciamento de banco de dados (SGBD), gratuito e livre gerenciado pela Oracle. Criado inicialmente em 1995, foi sofrendo evoluções com o tempo, e atualmente é a plataforma mais utilizada no mundo. Trata-se de um serviço estável, seguro e confiável. Uma ferramenta bastante poderosa. O MySQL foi criado por David Axmark e Michael Widenius, dois engenheiros da Suécia. [9]

2.1.5 Sequelize

O Sequelize é um ORM (Object-Relational Mapper) para Node.js, que tem suporte aos bancos de dados PostgreSQL, MariaDB, MySQL, SQLite e MSSQL, como ORM ele faz o mapeamento de dados relacionais (tabelas, colunas e linhas) para objetos Javascript.

Ele permite criar, buscar, alterar e remover dados do banco de dados utilizando métodos JS, além de permitir a modificação da estrutura das tabelas, com isso temos muita facilidade tanto na criação, população e migração de banco de dados [10].

2.2 Recursos de Hardware

A aplicação deverá ser hospedada em um servidor com processador Intel i7 ou similar, com mínimo de 16 GB de memória RAM e serão necessários 2TB de espaço em disco para o sistema.

2.3 Descrição dos Cadastros

2.3.1 Cadastro de usuários

O cadastro de usuários será gerenciado pelo administrador do sistema que acessará no menu a opção Adm. Na página de cadastro terá um lista com usuários já cadastrados com opção de editar ou desativar o usuário, ao lado um formulário para cadastro de um novo usuário onde será necessário informar o nome, e-mail, cargo, tipo de permissões de acesso e a senha .Os campos deverão ser validados, e o sistema não deixará que dados incorretos sejam cadastrados, para evitar futuros problemas.

2.3.2 Cadastro de produção

O cadastro de produção é gerenciado por usuário da equipe de PCP - Programação e Controle de Produção. Para executar um cadastro de programação o usuário deverá acessar o formulário no menu clicando PCP e depois em Produção. Na página de cadastro terá uma lista com o histórico dos últimos cadastros realizados, ao lado um formulário para cadastro de uma nova programação com os dados pertinentes à produção.

2.3.3 Cadastro de produto

O cadastro de produtos será gerenciado por usuário da equipe de PCP, acessando o menu e clicando em PCP e depois em produtos. Os produtos cadastrados e com status de ativo serão listados na tela com opção de editar o cadastro e de desativar o produto. Na lateral direita da tela fica o formulário de cadastro com os campos pertinentes ao cadastro de um novo produto.

2.3.4 Cadastro pausas

O cadastro de pausas será gerenciado por usuário da equipe de PCP, acessando o menu e clicando em PCP e depois em pausas. As pausas cadastrados e com status de ativo serão listados na tela com opção de editar o cadastro e de desativar a pausa. Na lateral direita da tela fica o formulário

de cadastro com os campos pertinentes ao cadastro de uma nova pausa.

2.3.5 Cadastro de parada

O cadastro de parada é automático e é executado quando o sistema recebe uma notificação de um evento de parada de produção. Sendo assim o sistema grava a hora de início e fim da parada e vincula a produção que estiver com status de iniciada no momento. A lista de paradas pode ser gerenciada por usuário do grupo de acesso Operação, onde o operador pode registrar o motivo da parada da linha de produção.

2.3.6 Cadastro de motivo

Os motivos de paradas são gerenciados por usuários do grupo de acesso Operação, podem ser acessados no menu clicando em Operação e depois em Motivos que serão listados na tela com opção de editar o cadastro e de desativar o produto. Na lateral direita da tela fica o formulário de cadastro com os campos pertinentes ao cadastro de um novo produto.

2.4 Descrição dos Processos / Movimentos

2.4.1 Gerenciamento de Produção

O sistema Radar 4.0 conta com uma série de sensores de campo que automatizam a coleta de dados ligados a aplicação por um gateway com web server que fornece os dados via protocolos web. A aplicação gerencia as informações e persiste os dados da produção no banco de dados. Tais dados ao serem tratados geram informações que serão expostos aos usuários em formas gráficas e de indicadores de fácil interpretação. Os usuários cadastrados possuem níveis de acesso organizados por grupos, e cada grupo terá permissão restrita gerenciada pelo administrador.

O sistema conta com um formulário de programação da produção que é a base que referencia o tratamento dos dados em função da quantidade de itens a serem produzidos, a velocidade nominal da linha e as pausas programadas a serem realizadas. A aplicação será responsiva e poderá ser usada em desktop ou dispositivos móveis.

2.5 Descrição dos Relatórios

2.5.1 Descrição do relatório de Produtos

Este relatório lista o cadastro de produtos podendo filtrar por nome do produto, período cadastrado entre duas datas informadas ou status do cadastro. Exemplo: A consulta a seguir lista os produtos cadastrados entre os dias 07/10/2020 a 08/10/2020, sem filtro de nome e status Ativo.



Consulta de Cadastro de Produtos:

Nome do Produto

Cadastrado no Período entre:

Status do Cadastro

Figura 1: Protótipo de formulário de consulta de Produtos

Relatórios de Produtos

Nome	Descrição	Velocidade Produção	Data de Cadastro	Status
Produto 02	Descrição produto 02	500	07-10-2020 19:51	Ativo
Produto 03	Descrição do produto 03	830	07-10-2020 19:56	Ativo
Produto 04	Descrição do produto 04	150	07-10-2020 19:58	Ativo
Produto 05	Descrição do produto 05	220	07-10-2020 20:07	Ativo

« 1 »

Figura 2: Protótipo de lista de relatório de Produtos.

2.5.2 Descrição do relatório de Pausas

Este relatório lista o cadastro de pausas podendo filtrar por nome, período cadastrado entre duas datas informadas ou status do cadastro. Exemplo: A consulta a seguir lista as pausas cadastrados com status ativo, sem filtro de nome ou data.

Consulta de Cadastro de Pausas:

Nome da Pausa

Cadastrado no Período entre:

Status do Cadastro

Figura 3: Protótipo de formulário de consulta de Pausas

Relatórios de Pausas

Nome	Inico	Fim	Duração	Data de Cadastro	Status
Café da Manhã	06:00	06:30	30 minutos	12-11-2020 22:06	Ativo
Almoço	12:00	13:00	1 hora	12-11-2020 22:08	Ativo
1ª Pausa ergonômica	08:00	08:15	15 minutos	12-11-2020 22:08	Ativo
2ª Pausa ergonômica	14:15	14:35	20 minutos	12-11-2020 22:09	Ativo

« 1 »

Figura 4: Protótipo de lista de relatório de Pausas.

2.5.3 Descrição do relatório de Produções

Este relatório lista o cadastro de produções podendo filtrar períodos cadastrados entre duas datas informadas, situação da produção e status do cadastro. Exemplo: A consulta a seguir lista as produções finalizadas entre 27/10/2020 e 02/12/2020, com status ativo .

Consulta de Produções:

Período iniciando em:

27/10/2020 

Período finalizando em:

02/12/2020 

Status da produção:

Planejada

Finalizada

Em Execução

Incluir:

Apenas produções ativas

Apenas produções desativadas

Figura 5: Protótipo de formulário de consulta de Produção

Relatórios de Produção

Produções de 27/10/2020, até 02/12/2020, com status finalizada, incluindo apenas produções ativas

Data de Início	Lote	Produto	Planejado	Produzido	Defeito
29-10-2020 22:56	003	Produto 02	850	0	0
29-10-2020 23:00	003	Produto 02	50	0	0
02-12-2020 22:20	05	Produto 04	30	55	10
02-12-2020 22:30	05	Produto 04	30	30	14
02-12-2020 22:30	5	Produto 04	30	20	10
02-12-2020 22:40	05	Produto 04	30	20	10
02-12-2020 23:00	05	Produto 04	30	19	11
02-12-2020 23:10	05	Produto 04	30	42	9
02-12-2020 23:10	05	Produto 04	30	31	0
02-12-2020 23:15	05	Produto 04	30	31	0

Figura 6: Protótipo de lista de relatório de Produção.

2.5.4 Descrição do relatório de Paradas

Este relatório lista o cadastro de paradas podendo filtrar por motivo, nome do usuário que atribuiu o motivo a parada e período cadastrado entre duas datas informadas. Exemplo: A consulta a seguir lista as paradas cadastrados entre 26/11/2020 e 29/11/2020, sem outros filtros .

Consulta de Cadastro de Paradas:

Cadastrado no período inicial entre:

26/11/2020 

29/11/2020 

Producao

Motivo

Figura 7: Protótipo de formulário de consulta de Paradas

Relatórios de Paradas

Produção	Inico	Fim	Duração	Motivo
1	26/11/2020 21:52	26/11/2020 21:52	0 segundos	Falta de matéria prima
3	26/11/2020 22:24	26/11/2020 22:24	4 segundos	Falha 002
3	26/11/2020 22:24	26/11/2020 22:24	11 segundos	Falha 002
3	28/11/2020 12:02	28/11/2020 12:02	5 segundos	Falha 002
3	28/11/2020 12:05	28/11/2020 12:06	8 segundos	Falha 002
3	28/11/2020 12:13	28/11/2020 12:13	5 segundos	Falha 002
3	28/11/2020 12:13	28/11/2020 12:13	5 segundos	Falha 002
3	28/11/2020 12:14	28/11/2020 12:14	3 segundos	Falha 002
3	28/11/2020 12:15	28/11/2020 12:15	1 segundo	Falha 002
3	28/11/2020 12:19	28/11/2020 12:19	2 segundos	Falta de matéria prima

« 1 »

Figura 8: Protótipo de lista de relatório de Paradas.

2.5.5 Descrição do relatório de Motivos

Este relatório lista o cadastro de motivos podendo filtrar por nome, período cadastrado entre duas datas informadas ou status do cadastro. Exemplo: A consulta a seguir lista o motivo cadastrado com nome contendo a palavra “Falta”, sem outros filtros.

Consulta de Cadastro de Motivos:

Descrição do Motivo

Cadastrado no Período entre:

Status do Cadastro

Figura 9: Protótipo de formulário de consulta de Motivos

Relatórios de Motivos

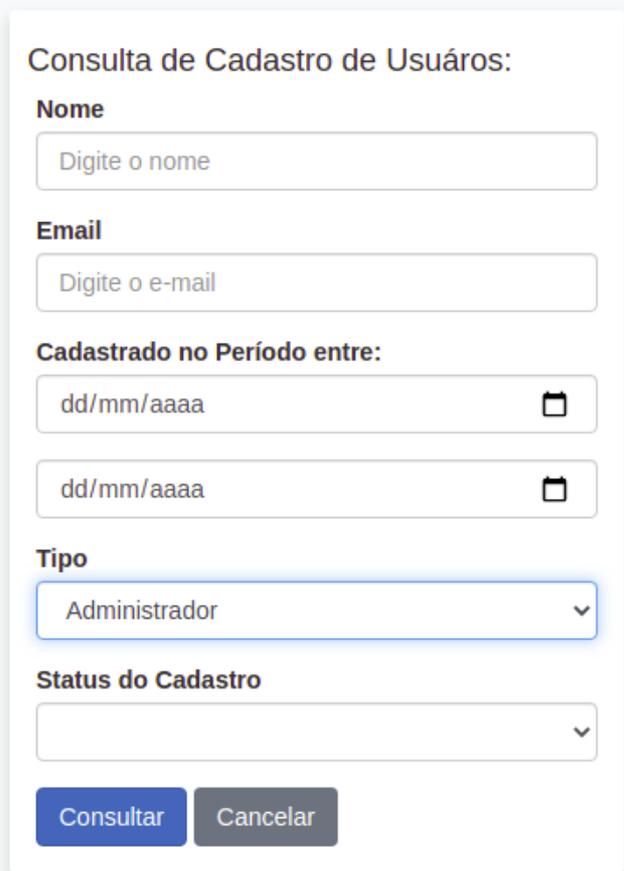
Descrição	Data de Cadastro	Status
Falta de matéria prima	04-11-2020 22:59	Ativo
Falta de energia elétrica	15-11-2020 19:04	Ativo

« 1 »

Figura 10: Protótipo de lista de relatório de Motivos.

2.5.6 Descrição do relatório de Usuários

Este relatório lista o cadastro de usuários podendo filtrar por nome, período cadastrado entre duas datas informadas, tipo ou status do cadastro. Exemplo: A consulta a seguir lista os usuários cadastrados com tipo Administrador, sem outros filtros.



Consulta de Cadastro de Usuários:

Nome

Email

Cadastrado no Período entre:

Tipo

Status do Cadastro

Figura 11: Protótipo de formulário de consulta de Usuários

Relatórios de Usuários

Nome	Email	Tipo	Data de Cadastro	Status
Lenovo	lenovo@mail.com	Administrador	25-10-2020 00:10	Desativado
Adm	adm@mail.com	Administrador	28-10-2020 20:20	Desativado
Fulano da Silva	fulano@mail.com	Administrador	03-11-2020 23:04	Ativo

« 1 »

Figura 12: Protótipo de lista de relatório de Usuários.

3 DIAGRAMA ENTIDADE-RELACIONAMENTO

O Modelo Entidade Relacionamento (também chamado Modelo ER, ou simplesmente MER), como o nome sugere, é um modelo conceitual utilizado na Engenharia de Software para descrever os objetos (entidades) envolvidos em um domínio de negócios, com suas características (atributos) e como elas se relacionam entre si (relacionamentos).

Em geral, este modelo representa de forma abstrata a estrutura que possuirá o banco de dados da aplicação. Obviamente, o banco de dados poderá conter várias outras entidades, tais como chaves e tabelas intermediárias, que podem só fazer sentido no contexto de bases de dados relacionais [6].

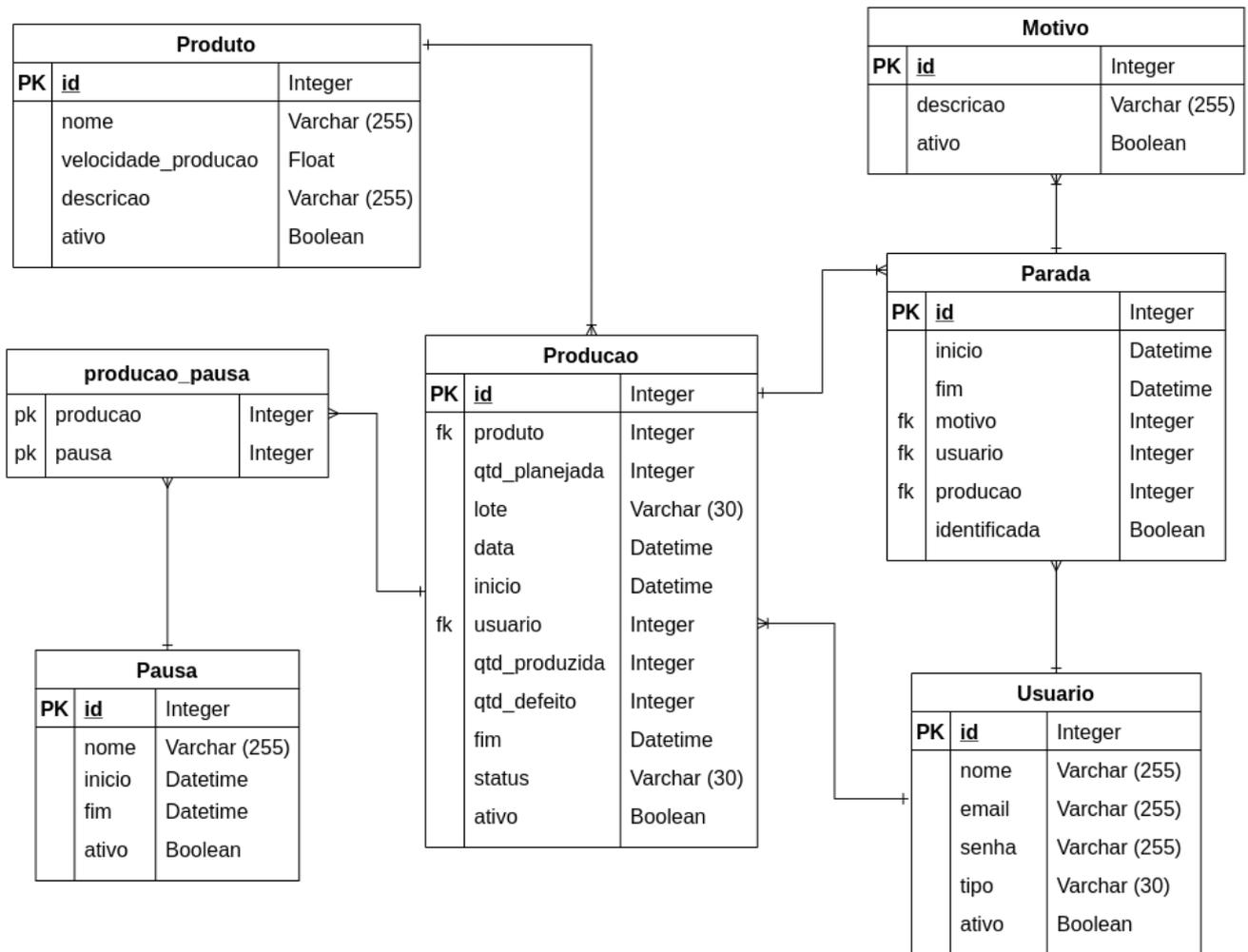


Figura 13: Modelo do diagrama de entidade e relacionamento do sistema.

4 CASOS DE USO

4.1 Diagrama de Casos de Uso

Esse diagrama documenta o que o sistema faz do ponto de vista do usuário. Em outras palavras, ele descreve as principais funcionalidades do sistema e a interação dessas funcionalidades com os usuários do mesmo sistema. Neste diagrama não nos aprofundamos em detalhes técnicos que dizem como o sistema faz [4].

A seguir, apresentamos a Figura 3 que mostra todos os Casos de Uso utilizados no desenvolvimento do Sistema.

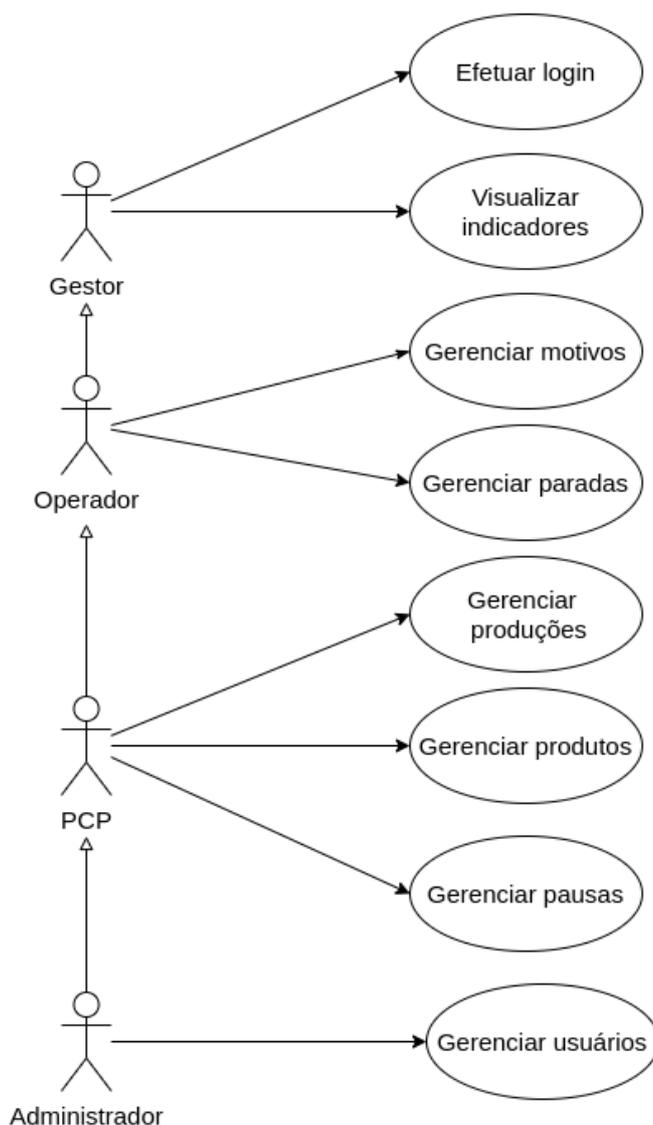


Figura 14: Diagrama de Caso de Uso

4.2 Especificação de Caso de Uso: Efetuar Login

4.2.1 Descrição

Esse caso de uso destina-se a permitir o acesso dos usuários de cada perfil do sistema desenvolvido.

4.2.2 Fluxo básico

O usuário deve preencher os campos de “usuário” e “senha” para efetuar o login, após isso o usuário seleciona a opção “Entrar”, o sistema valida os campos do formulário e acessa a tela inicial home.

4.2.3 Fluxos alternativos

Usuário ou Senha Inválidos: O usuário pode digitar errado um “usuário” ou “senha” e em seguida clicar no botão “Entrar”. O sistema mostrará uma mensagem dizendo “Usuário ou senha incorretos”. Após isso, o usuário deve corrigir os campos e selecionar novamente a opção “Entrar”. Caso estejam corretos o sistema acessa a tela inicial.

Esqueceu usuário ou senha: O usuário pode não lembrar seu “usuário” ou “senha”. Assim, ele pode clicar no link “Esqueci minha Senha” e aparecerá a mensagem “Para alterar a senha ou visualizar a senha atual, entre em contato com o administrador do sistema”.

4.2.4 Precondições

Nenhum.

4.2.5 Pós-condições

Depois da execução deste caso de uso o sistema poderá executar um dos casos de uso descritos a seguir: **Gestor:** Visualizar relatórios. **Operador:** Visualizar relatórios e Gerência paradas. **PCP:** Gerência programação da produção. **Administrador:** Gerência usuários

4.2.6 Pontos de inclusão ou extensão

Nenhum.

4.2.7 Dicionário de dados

TABELA USUÁRIO

ATRIBUTO	TIPO	OBRIGATÓRIO	REPETIDO	DESCRIÇÃO
ID	NUMBER	SIM	NÃO	CÓDIGO DO USUÁRIO
EMAIL	VARCHAR	SIM	NÃO	EMAIL DO USUÁRIO
SENHA	VARCHAR	SIM	SIM	SENHA DE ACESSO
TIPO	VARCHAR	SIM	SIM	PERMISSÃO DE ACESSO
ATIVO	BOOLEAN	SIM	SIM	STATUS

4.2.8 Protótipo de interface



O protótipo de interface de login de usuário apresenta um layout centralizado em um fundo cinza claro. No topo, há um ícone de engrenagem com um cubo amarelo no centro. Abaixo do ícone, o texto "Faça seu login" é exibido em uma fonte grande e escura. Seguem dois campos de entrada de texto: o primeiro rotulado "Endereço de E-mail" e o segundo rotulado "Senha". Abaixo dos campos, há um botão azul com o texto "Entrar" em branco. Na base do formulário, o link "Esqueci a senha..." é exibido em uma cor roxa.

Figura 15: Protótipo Login de Usuário

4.3 Especificação de Caso de Uso: Visualizar Indicadores

4.3.1 Descrição

Esse caso de uso destina-se a visualizar os dados de forma gráfica para auxiliar os gestores a monitorar o desempenho das atividades e na tomada de decisões.

4.3.2 Fluxo básico

O usuário deve executar o login com tipo de acesso “Gestor”, após isso o sistema direciona para tela Home onde os dados são apresentados de forma gráfica. Clicando no botão “Gestão” no menu, pode ser acessado os relatórios gerenciais.

4.3.3 Fluxos alternativos

Nenhum.

4.3.4 Precondições

Tem executado o login.

4.3.5 Pós-condições

Depois da execução deste caso de uso o sistema poderá executar um dos casos de uso descritos a seguir: **Operador**: Visualizar relatórios e Gerência paradas. **PCP**: Gerência programação da produção. **Administrador**: Gerência usuários

4.3.6 Pontos de inclusão ou extensão

Nenhum.

4.3.7 Dicionário de dados

Não se aplica.

4.3.8 Protótipo de interface

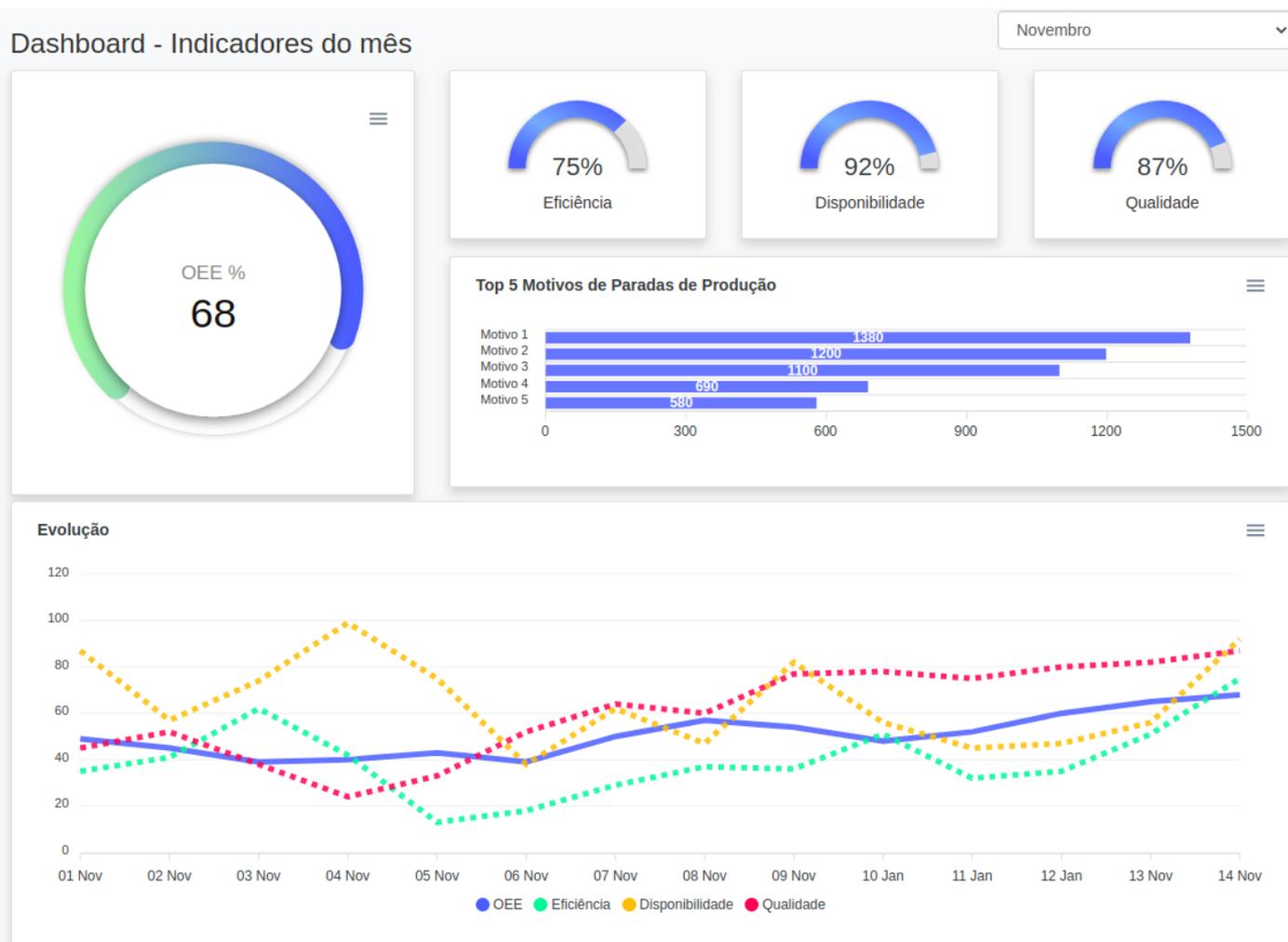


Figura 16: Protótipo Página Home

4.4 Especificação de Caso de Uso: Gerenciar Motivos

4.4.1 Descrição

Esse caso de uso destina-se a cadastrar, manter atualizado e inativar motivos de paradas.

4.4.2 Fluxo básico

O usuário do sistema preenche todos os campos necessários para cadastrar no sistema um novo motivo, após isso seleciona a opção "Salvar". O sistema então valida os campos necessários do formulário, salva e atualiza a lista de motivos.

4.4.3 Fluxos alternativos

Alterar um cadastro: Clicando no botão “Editar”. O sistema retornará os dados ao formulário de edição, e estes poderão ser alterados, após isso seleciona-se a opção “Salvar” e o sistema atualiza a lista de motivos e mostra a mensagem “Dados editados com sucesso”.

Formulário não validado: O sistema pode não validar o formulário de dados. O usuário então deverá corrigir os erros mostrados e selecionar a opção “Salvar”. O sistema valida novamente os campos do formulário, faz as ações de salvar ou alterar, atualizando a lista de motivos.

Inativar um cadastrado: Clicando no botão “desativar” o sistema emite um alerta para confirmação da ação clicando no botão “Sim”, após isso o sistema atualiza a lista de motivos e mostra a mensagem “Cadastro de motivo desativado com Sucesso”.

4.4.4 Precondições

Antes da execução deste caso de uso o sistema deverá ter executado um dos casos de usos descritos a seguir: **Operador:** Efetuar Login.

4.4.5 Pós-condições

Depois da execução deste caso de uso o sistema poderá executar um dos casos de uso descritos a seguir: **Operador:** Gerência paradas. **PCP:** Gerência programação da produção. **Administrador:** Gerência usuários

4.4.6 Pontos de inclusão ou extensão

Nenhum.

4.4.7 Dicionário de dados

TABELA MOTIVOS

ATRIBUTO	TIPO	OBRIGATÓRIO	REPETIÇÃO	DESCRIÇÃO
ID	NUMBER	SIM	NÃO	CÓDIGO DO MOTIVO
DESCRICAO	VARCHAR	SIM	NÃO	DESCRIÇÃO DO MOTIVO
ATIVO	BOOLEAN	SIM	SIM	STATUS DO CADASTRO

4.4.8 Protótipo de interface

Motivos de paradas

id	Descrição		
1	Falha no transportador de alimentação da linha 1		
3	Falha 002		
4	Falta de matéria prima		

Figura 17: Protótipo tabela de motivos cadastrados

Cadastrar motivo de parada:

Descrição

Figura 18: Protótipo formulário de cadastro de motivo de parada

Editar motivo de parada:

Descrição

Figura 19: Protótipo formulário de edição de motivo de parada

4.5 Especificação de Caso de Uso: Gerenciar Paradas

4.5.1 Descrição

Esse caso de uso destina-se a atribuir motivos as paradas de produção. Visto que as paradas de produção são cadastradas automaticamente com um evento sem um motivo pré definido. Este motivo deve ser especificado pelo usuário.

4.5.2 Fluxo básico

O usuário do sistema seleciona uma parada cadastrada clicando do check “Atribuir Motivo”, depois seleciona o motivo no campo de seleção onde previamente estarão listados os motivos previamente cadastrados no sistema. Clicando em “Atribuir” o sistema atribui o motivo selecionado a parada.

4.5.3 Fluxos alternativos

Alterar um cadastro: Não se aplica.

Formulário não validado: O sistema pode não validar o formulário de dados. O usuário então deverá corrigir os erros mostrados e selecionar a opção “Atribuir”. O sistema valida novamente os campos do formulário, faz as ações de salvar ou alterar, atualizando a lista de paradas.

Inativar um cadastrado: Não se aplica.

4.5.4 Precondições

Antes da execução deste caso de uso o sistema deverá ter executado um dos casos de usos descritos a seguir: **Operador:** Efetuar Login.

4.5.5 Pós-condições

Depois da execução deste caso de uso o sistema poderá executar um dos casos de uso descritos a seguir: **Operador:** Gerência paradas. **PCP:** Gerência programação da produção. **Administrador:** Gerência usuários

4.5.6 Pontos de inclusão ou extensão

Nenhum.

4.5.7 Dicionário de dados

TABELA PARADAS

ATRIBUTO	TIPO	OBRIGATÓRIO	REPETIÇÃO	DESCRIÇÃO
ID	NUMBER	SIM	NÃO	CÓDIGO DA PARADA
INÍCIO	DATETIME	SIM	NÃO	DATA E HORA DO INÍCIO DO EVENTO DE PARADA
FIM	DATETIME	NÃO	NÃO	DATA E HORA DO FIM DO EVENTO DE PARADA
IDENTIFICADA	BOOLEAN	SIM	SIM	DEFINIÇÃO SE O EVENTO FOI IDENTIFICADO COM UM MOTIVO
USUARIO_ID	NUMBER	NÃO	SIM	ID DO USUÁRIO QUE EXECUTA A IDENTIFICAÇÃO DO MOTIVO DO EVENTO
MOTIVO_ID	NUMBER	NÃO	SIM	ID DO MOTIVO ATRIBUÍDO

4.4.8 Protótipo de interface

Paradas com motivo não identificado				
id	Início	Fim	Duração	Atribuir Motivo
1	27/08/2020 00:31:50	27/08/2020 00:31:50	00:00	<input checked="" type="checkbox"/>
2	27/08/2020 00:31:52	27/08/2020 00:31:52	00:00	<input type="checkbox"/>
6	06/11/2020 16:00:00	06/11/2020 16:02:30	2 minutos 30 segundos	<input type="checkbox"/>

Figura 20: Protótipo lista de paradas com motivo não identificado

Selecione um motivo:

Falta de matéria prima
▼

Atribuir
Cancelar

Figura 21: Protótipo lista de motivo para atribuir a parada selecionada

4.5 Especificação de Caso de Uso: Gerenciar Produção

4.5.1 Descrição

Esse caso de uso destina-se a cadastrar, manter atualizado e desativar um cadastro de produção.

4.5.2 Fluxo básico

O usuário do sistema preenche todos os campos necessários para cadastrar no sistema uma nova produção, após isso seleciona a opção “Cadastrar”. O sistema então valida os campos necessários do formulário, salva e atualiza a lista de produções.

4.5.3 Fluxos alternativos

Alterar um cadastro: Clicando no botão “Editar”. O sistema retornará os dados ao formulário de edição, e estes poderão ser alterados, após isso seleciona-se a opção “Salvar” e o sistema atualiza a lista de produções e mostra a mensagem “Dados editados com sucesso”.

Formulário não validado: O sistema pode não validar o formulário de dados. O usuário então deverá corrigir os erros mostrados e selecionar a opção “Cadastrar”. O sistema valida novamente os campos do formulário, faz as ações de salvar ou alterar, atualizando a lista de produções.

Inativar um cadastrado: Clicando no botão “desativar” o sistema emite um alerta para confirmação da ação, clicando no botão “Sim”, o sistema atualiza a lista de produções e mostra a mensagem “Cadastro de produção desativado com Sucesso”.

Atribuir Pausa: Clicando no botão “Atribuir Pausa” o sistema retornará os dados ao formulário da página de atribuição de pausas. Na lateral aparecerá as pausas previamente cadastradas. O usuário atribui a pausa a produção clicando no botão atribuir.

4.5.4 Precondições

Antes da execução deste caso de uso, o sistema deverá ter executado um dos casos de usos descritos a seguir: **PCP:** Efetuar Login.

4.5.5 Pós-condições

Depois da execução deste caso de uso o sistema poderá executar um dos casos de uso descritos a seguir: **Operador**: Gerência paradas. **PCP**: Gerência programação da produção. **Administrador**: Gerência usuários

4.5.6 Pontos de inclusão ou extensão

Nenhum.

4.5.7 Dicionário de dados

TABELA PRODUÇÃO

ATRIBUTO	TIPO	OBRIGATÓRIO	REPETIÇÃO	DESCRIÇÃO
ID	NUMBER	SIM	NÃO	CÓDIGO DA PRODUÇÃO
LOTE	VARCHAR	SIM	NÃO	LOTE DA PRODUÇÃO
DATA	DATETIME	SIM	NÃO	DATA E HORA DE INÍCIO DA PRODUÇÃO
ID_PRODUTO	NUMBER	SIM	SIM	IDENTIFICADOR DO PRODUTO A SER PRODUZIDO
QTD_PLANEJADA	NUMBER	SIM	SIM	QUANTIDADE DE ITENS A SEREM PRODUZIDOS
QTD_DEFEITO	NUMBER	NÃO	SIM	QUANTIDADE DE ITENS PRODUZIDOS COM DEFEITO
ID_USUARIO	NUMBER	SIM	SIM	IDENTIFICADOR DO USUÁRIO QUE CADASTROU OU EDITOU OS DADOS DA PRODUÇÃO
FINALIZADAEM	DATETIME	NÃO	SIM	DATA DE FINALIZAÇÃO DA PRODUÇÃO
STATUS	VARCHAR	SIM	SIM	SITUAÇÃO DA PRODUÇÃO PODENDO SER PROGRAMADA, EM EXECUÇÃO, FINALIZADA
ATIVO	BOOLEAN	SIM	SIM	STATUS DO CADASTRO

4.5.8 Protótipo de interface

Produções Cadastradas							
id	Data de Inicio	Lote	Produto	Planejado	Atribuir Pausa	Desativar	Editar
1	02-09-2020 06:00	001	Produto 01	100			
2	10-10-2020 00:00	002	Produto 02	50			
3	12-12-2020 00:00	003	Abate	850			
4	17-10-2020 00:00	004	Produto 02	150			
5	18-10-2020 00:00	005	Produto 02	200			

Figura 22: Protótipo tabela de produções cadastradas

Cadastrar Produção:

Data

Horário de Inicio

Lote

Produto

Quantidade

Figura 23: Protótipo formulário de cadastro de produção

Editar Produção:

Data

02/09/2020 

Horário de Início

06:00 

Lote

001

Produto

Produto 01

Salvar **Cancelar**

Figura 24: Protótipo formulário de edição de produção

Atribuir pausas a produção:

Café da Manhã
das 06:00 a 06:30 com duração de 30 minutos

Atribuir

Almoço
das 12:00 a 13:00 com duração de 1 hora

Atribuir

1ª Pausa ergonômica
das 08:00 a 08:15 com duração de 15 minutos

Atribuir

Figura 25: Protótipo formulário de atribuição de pausas a produção

4.6 Especificação de Caso de Uso: Gerenciar Produtos

4.6.1 Descrição

Esse caso de uso destina-se a cadastrar, manter atualizado e desativar um cadastro de produto.

4.6.2 Fluxo básico

O usuário do sistema preenche todos os campos necessários para cadastrar no sistema um novo produto, após isso seleciona a opção “Cadastrar”. O sistema então valida os campos necessários do formulário, salva e atualiza a lista de produtos.

4.6.3 Fluxos alternativos

Alterar um cadastro: Clicando no botão “Editar”. O sistema retornará os dados ao formulário de edição, e estes poderão ser alterados, após isso seleciona-se a opção “Salvar” e o sistema atualiza a lista de produtos e mostra a mensagem “Dados editados com sucesso”.

Formulário não validado: O sistema pode não validar o formulário de dados. O usuário então deverá corrigir os erros mostrados e selecionar a opção “Cadastrar”. O sistema valida novamente os campos do formulário, faz as ações de salvar ou alterar, atualizando a lista de produtos.

Inativar um cadastrado: Clicando no botão “desativar” o sistema emite um alerta para confirmação da ação, clicando no botão “Sim”, o sistema atualiza a lista de produtos e mostra a mensagem “Cadastro de produto desativado com Sucesso”.

4.6.4 Precondições

Antes da execução deste caso de uso, o sistema deverá ter executado um dos casos de usos descritos a seguir: **PCP:** Efetuar Login.

4.6.5 Pós-condições

Depois da execução deste caso de uso o sistema poderá executar um dos casos de uso descritos a seguir: **Operador:** Gerência paradas. **PCP:** Gerência programação da produção. **Administrador:** Gerência usuários

4.6.6 Pontos de inclusão ou extensão

Nenhum.

4.6.7 Dicionário de dados

TABELA PRODUTO

ATRIBUTO	TIPO	OBRIGATÓRIO	REPETIÇÃO	DESCRIÇÃO
ID	NUMBER	SIM	NÃO	CÓDIGO DA PRODUÇÃO
NOME	VARCHAR	SIM	NÃO	NOME DO PRODUTO
VELOCIDADE	FLOAT	SIM	SIM	VELOCIDADE DE PRODUÇÃO DE ITENS POR HORA
DESCRICAÇÃO	VARCHAR	NÃO	SIM	DESCRIÇÃO DO PRODUTO
ATIVO	BOOLEAN	SIM	SIM	STATUS DO CADASTRO

4.6.8 Protótipo de interface

Produtos Cadastrados					
id	Nome	Descrição	Velocidade de Produção [und / h]		
1	Produto 01	Descrição do produto 01	120		
2	Produto 02	Descrição produto 02	500		
3	Produto 03	Descrição do produto 03	830		
4	Produto 04	Descrição do produto 04	150		
5	Produto 05	Descrição do produto 05	220		

Figura 26: Protótipo tabela de produtos cadastrados

Cadastrar Produto:

Nome

Descrição
Velocidade de Produção

Figura 27: Protótipo formulário de cadastro de produtos

Editar Produto:

Nome

Descrição
Velocidade de Produção

Figura 28: Protótipo formulário de edição de produto

4.7 Especificação de Caso de Uso: Gerenciar Pausas

4.7.1 Descrição

Esse caso de uso destina-se a cadastrar, manter atualizado e inativar uma pausa.

4.7.2 Fluxo básico

O administrador do sistema preenche todos os campos necessários para cadastrar no sistema uma nova pausa, após isso seleciona a opção “Salvar”. O sistema então valida os campos necessários do formulário e salva os dados e atualiza a lista de pausas.

4.7.3 Fluxos alternativos

Alterar um cadastro: Apenas usuário do tipo PCP pode alterar os dados de uma pausa já cadastrada, clicando no botão “Editar”. O sistema retornará os dados ao formulário de edição, e estes poderão ser alterados, após isso seleciona-se a opção “Salvar” e o sistema atualiza a lista de pausas e mostra a mensagem “Cadastro de pausa editada com Sucesso”.

Formulário não validado: O sistema pode não validar o formulário de dados. O administrador então deverá corrigir os erros mostrados e selecionar a opção “Salvar”. O sistema valida novamente os campos do formulário, faz as ações de salvar ou alterar, atualizando a lista de pausas.

Inativar um cadastrado: Apenas usuário do tipo PCP pode desativar um cadastro clicando no botão “desativar” que é mostrado na lista de pausas. O sistema emite um alerta para confirmação da ação clicando no botão “Sim”, após isso o sistema atualiza a lista de pausas e mostra a mensagem “Cadastro de pausa desativado com Sucesso”.

4.7.4 Precondições

Antes da execução deste caso de uso, o sistema deverá ter executado um dos casos de usos descritos a seguir: **PCP:** Efetuar Login.

4.7.5 Pós-condições

PCP: Gerência programação da produção. **Administrador:** Gerência usuários

4.7.6 Pontos de inclusão ou extensão

Nenhum.

4.7.7 Dicionário de dados

TABELA PAUSAS

ATRIBUTO	TIPO	OBRIGATÓRIO	REPETIÇÃO	DESCRIÇÃO
ID	NUMBER	SIM	NÃO	CÓDIGO DA PAUSA
NOME	VARCHAR	SIM	NÃO	NOME DA PAUSA
INICIO	DATETIME	SIM	SIM	DATA E HORA DO FIM DA PAUSA
FIM	DATETIME	SIM	NÃO	DATA E HORA DO FIM DA PAUSA
ATIVO	BOOLEAN	SIM	SIM	STATUS DO CADASTRO

4.7.8 Protótipo de interface

Pausas Cadastradas

id	Nome	Inicio	Fim	Duração		
7	Café da Manhã	06:00	06:30	30 minutos		
8	Almoço	12:00	13:00	1 hora		
9	1ª Pausa ergonômica	08:00	08:15	15 minutos		
10	2ª Pausa ergonômica	14:15	14:35	20 minutos		

Figura 29: Protótipo tabela de pausas cadastradas

Cadastrar uma Pausa:

Nome

Início

Fim

Cadastrar **Cancelar**

Figura 30: Protótipo formulário de cadastro de pausa

Editar Pausa:

Nome

Início

Fim

Salvar **Cancelar**

Figura 31: Protótipo formulário de edição do cadastro de pausa

4.7 Especificação de Caso de Uso: Gerenciar Usuários

4.7.1 Descrição

Esse caso de uso destina-se a cadastrar, manter atualizado e inativar um usuário.

4.7.2 Fluxo básico

O administrador do sistema preenche todos os campos necessários para cadastrar no sistema um novo usuário, após isso seleciona a opção “Cadastrar”. O sistema então valida os campos necessários do formulário e salva os dados e atualiza a lista de usuários.

4.7.3 Fluxos alternativos

Alterar um cadastro: Apenas usuário do tipo Administrador pode alterar os dados de um usuário já cadastrado a, clicando no botão “Editar”. O sistema retornará os dados ao formulário de edição, e estes poderão ser alterados, após isso seleciona-se a opção “Salvar” e o sistema atualiza a lista de usuários e mostra a mensagem “Cadastro de usuário editado com Sucesso”.

Formulário não validado: O sistema pode não validar os dados do formulário. O administrador então deverá corrigir os erros mostrados e selecionar a opção “Salvar”. O sistema valida novamente os campos do formulário, faz as ações de salvar ou alterar, atualizando a lista de usuários.

Inativar um cadastro: Apenas usuários do tipo Administrador podem desativar um cadastro, clicando no botão “desativar” que é mostrado na lista de usuários. O sistema emite um alerta para confirmação da ação clicando no botão “Sim”, após isso o sistema atualiza a lista de usuários e mostra a mensagem “Cadastro de usuário desativado com Sucesso”.

4.7.4 Precondições

Antes da execução deste caso de uso o sistema deverá ter executado um dos casos de usos descritos a seguir: **Administrador:** Efetuar Login.

4.7.5 Pós-condições

Administrador: Gerência usuários

4.7.6 Pontos de inclusão ou extensão

Nenhum.

4.7.7 Dicionário de dados

TABELA USUÁRIO

ATRIBUTO	TIPO	OBRIGATÓRIO	REPETIDO	DESCRIÇÃO
ID	NUMBER	SIM	NÃO	CÓDIGO DO USUÁRIO
EMAIL	VARCHAR	SIM	NÃO	EMAIL DO USUÁRIO
SENHA	VARCHAR	SIM	SIM	SENHA DE ACESSO
TIPO	VARCHAR	SIM	SIM	PERMISSÃO DE ACESSO
ATIVO	BOOLEAN	SIM	SIM	STATUS

4.7.8 Protótipo de interface

Usuários Cadastrados

id	Nome	E-mail	Tipo		
19	Hélison Souza	helison.rs@gmail.com	Gestor		
20	Fulano da Silva	fulano@mail.com	Administrador		
21	Ciclano dos Santos	ciclano@mail.com	PCP		
22	Beltrano Soares	beltrano@mail.com.br	Operador		

Figura 32: Protótipo tabela de usuários cadastrados

Cadastrar Usuário:

Nome

Email
Tipo de permissão
Senha

Figura 33: Protótipo formulário de cadastro de usuário

Editar Usuario:

Nome

Email
Senha
Tipo de permissão

Figura 34: Protótipo formulário de edição do cadastro de usuário

5 DIAGRAMA DE ATIVIDADES

Um diagrama de atividade ilustra a natureza dinâmica de um sistema pela modelagem do fluxo de controle de atividade à atividade. Uma atividade representa uma operação em alguma classe no sistema que resulta em uma mudança no estado do sistema. Tipicamente, diagramas de atividades são usados para modelar fluxos de processos, processos de negócios ou operações internas. o diagrama de atividades é similar a uma máquina de estados, mas tem um propósito diferente, o qual envolve capturar ações e seus resultados em termos de mudanças do estado do objeto. O diagrama de atividades é representado por um gráfico de atividades que mostram o fluxo de uma atividade para outra. Esse fluxo é mostrado através de transições, que são setas direcionadas, mostrando o caminho entre os estados de atividade (ação) [7].

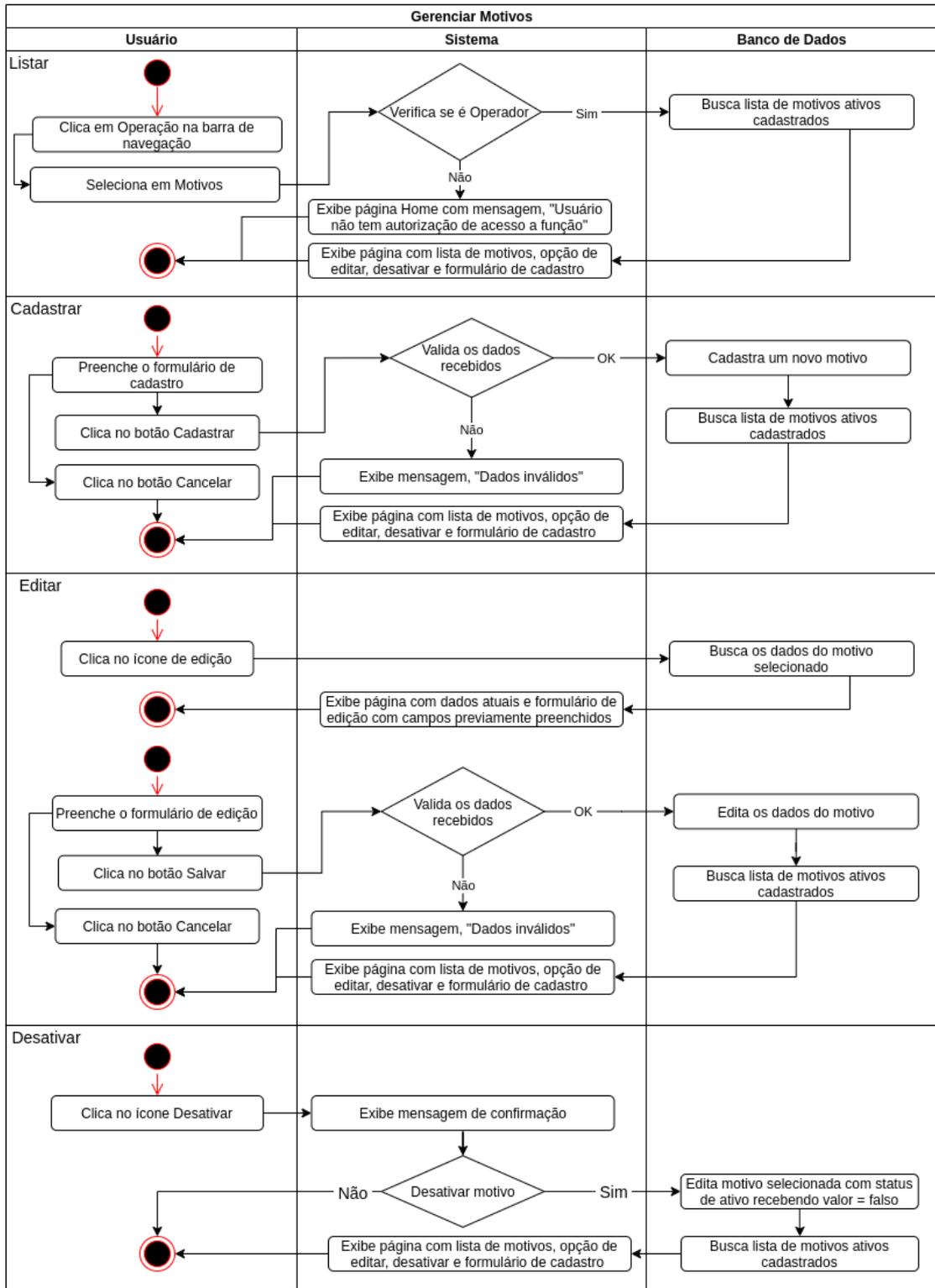


Figura 35: Diagrama de Atividades Gerenciar Motivos

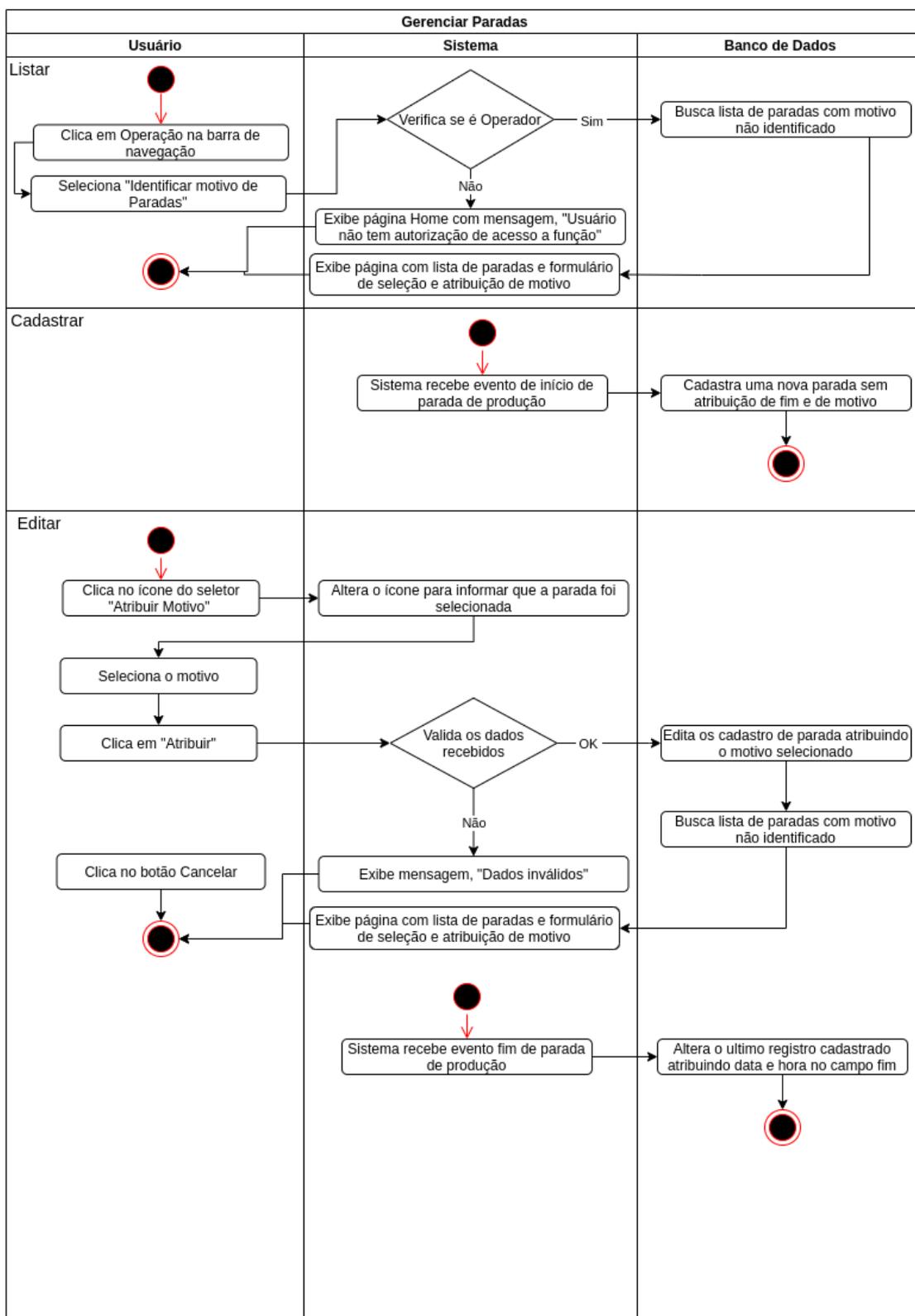


Figura 36: Diagrama de Atividades Gerenciar Paradas

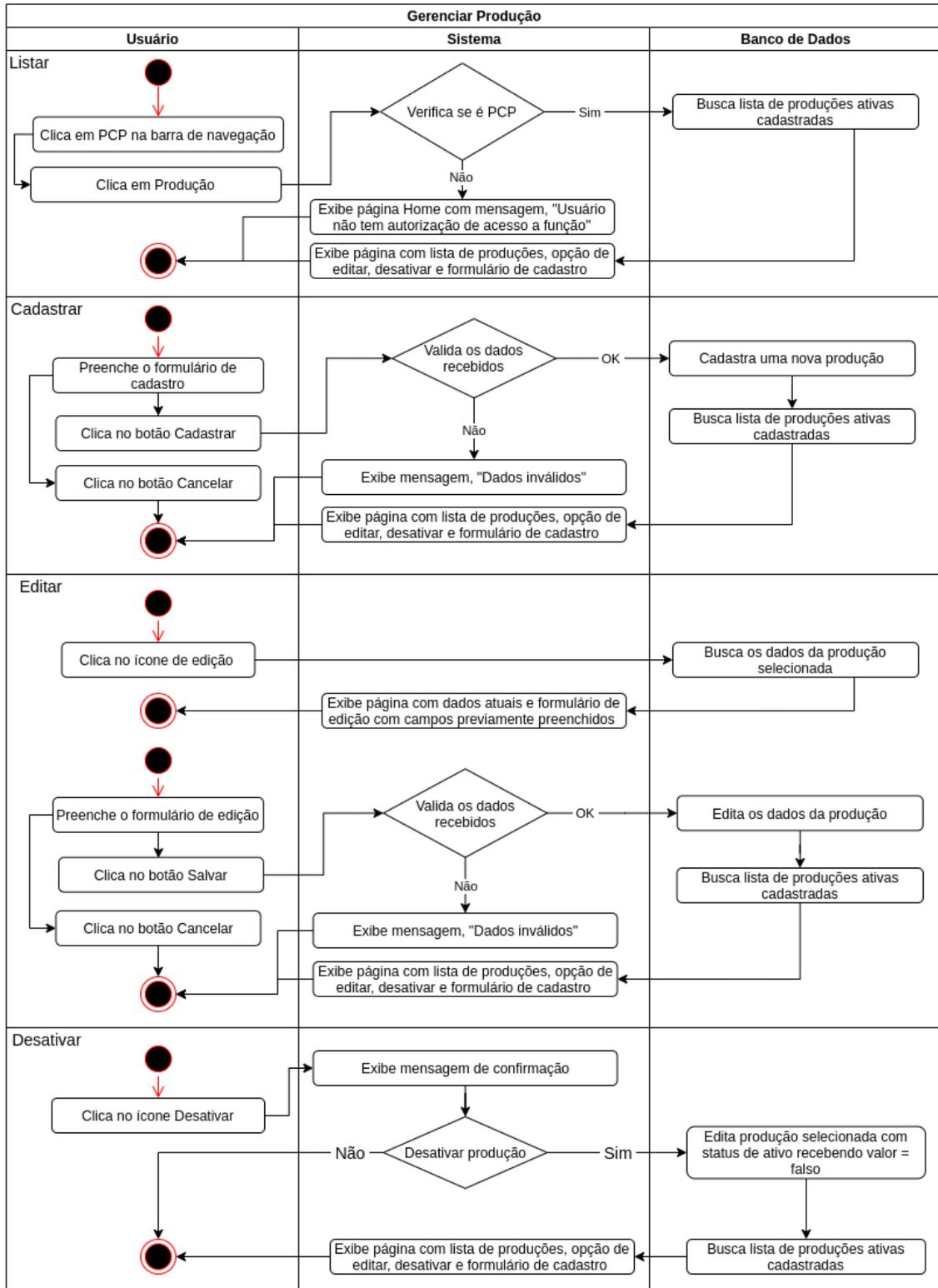


Figura 37: Diagrama de Atividades Gerenciar Produção

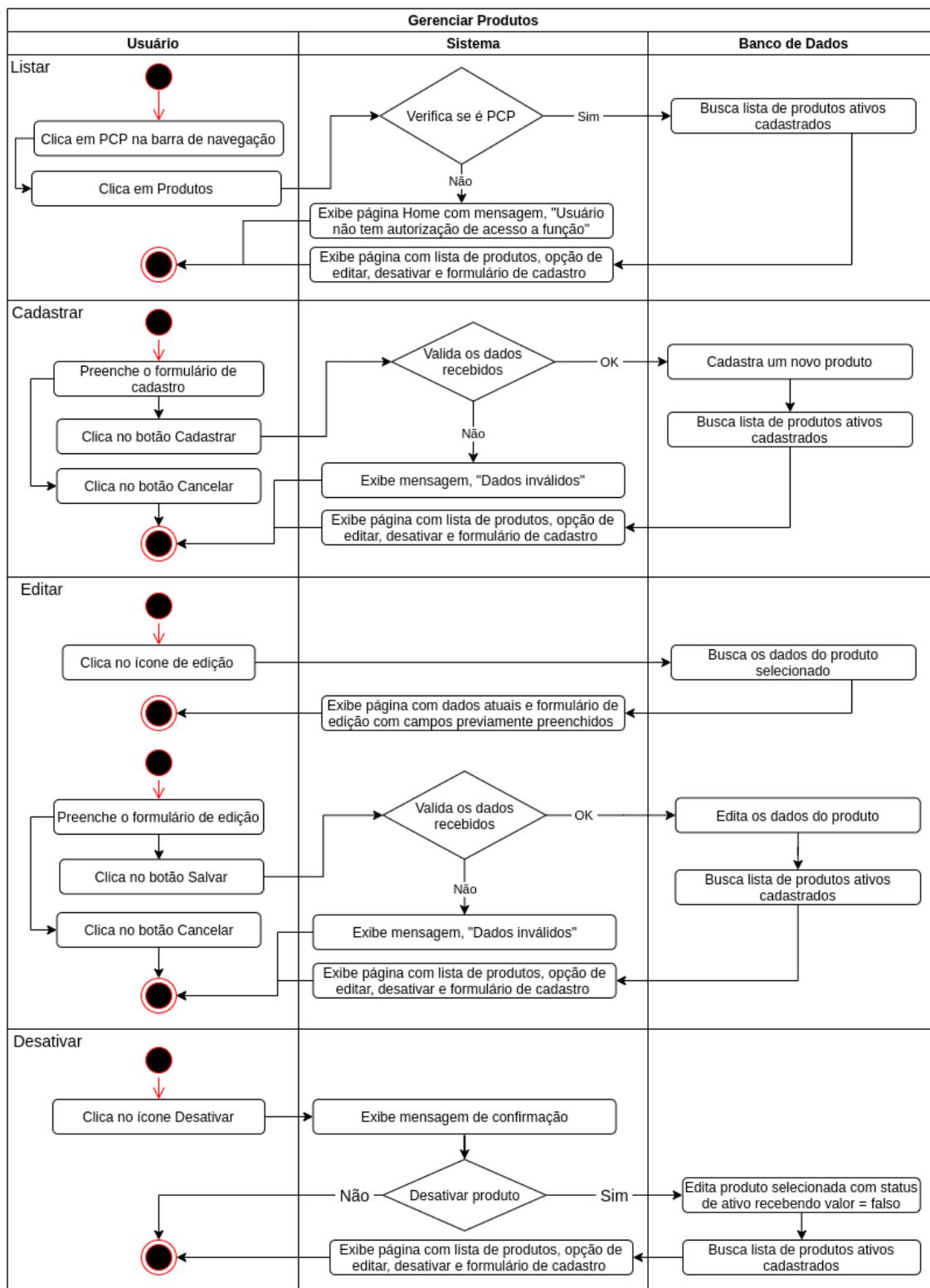


Figura 38: Diagrama de Atividades Gerenciar Produtos

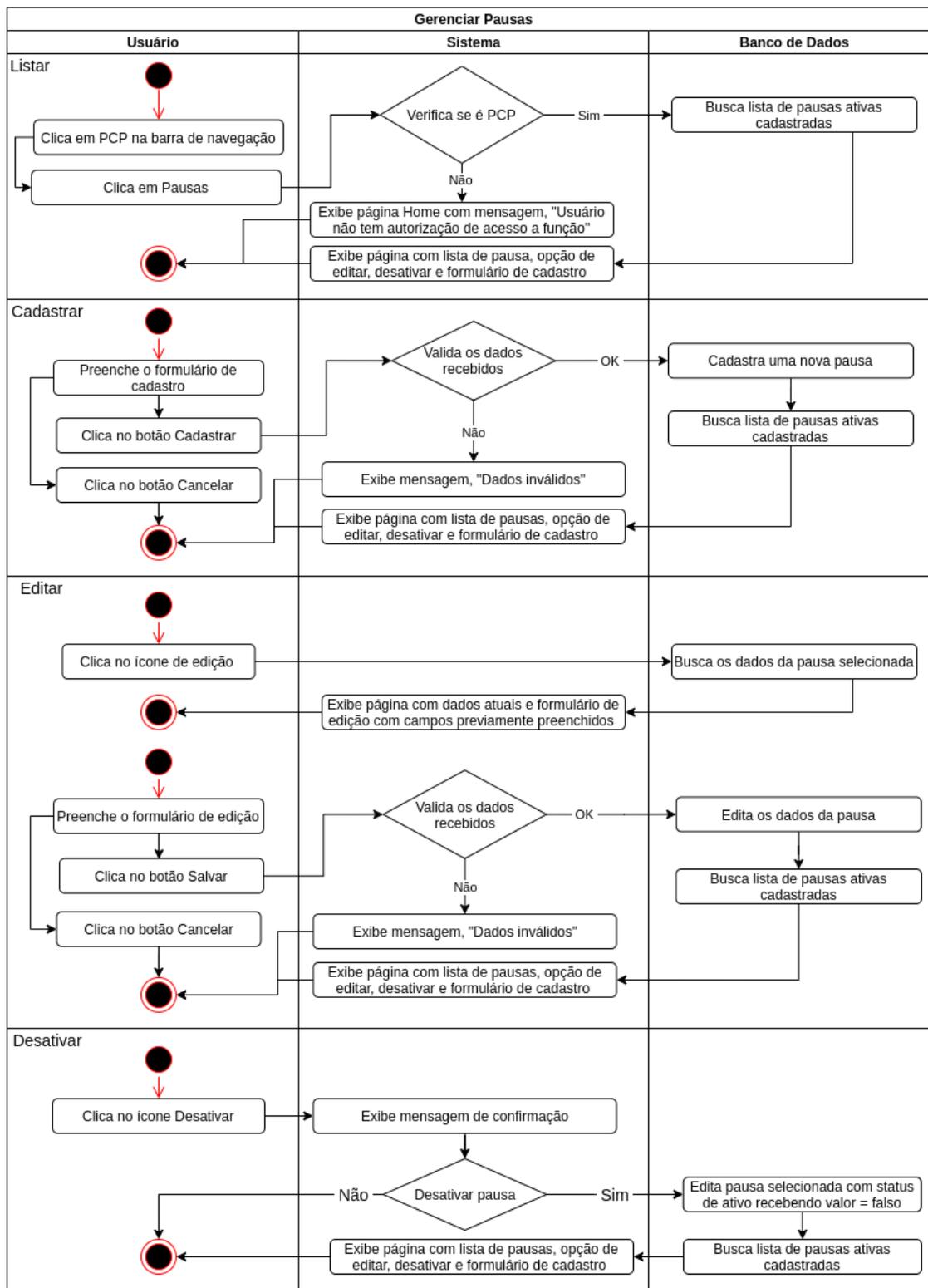


Figura 39: Diagrama de Atividades Gerenciar Pausas

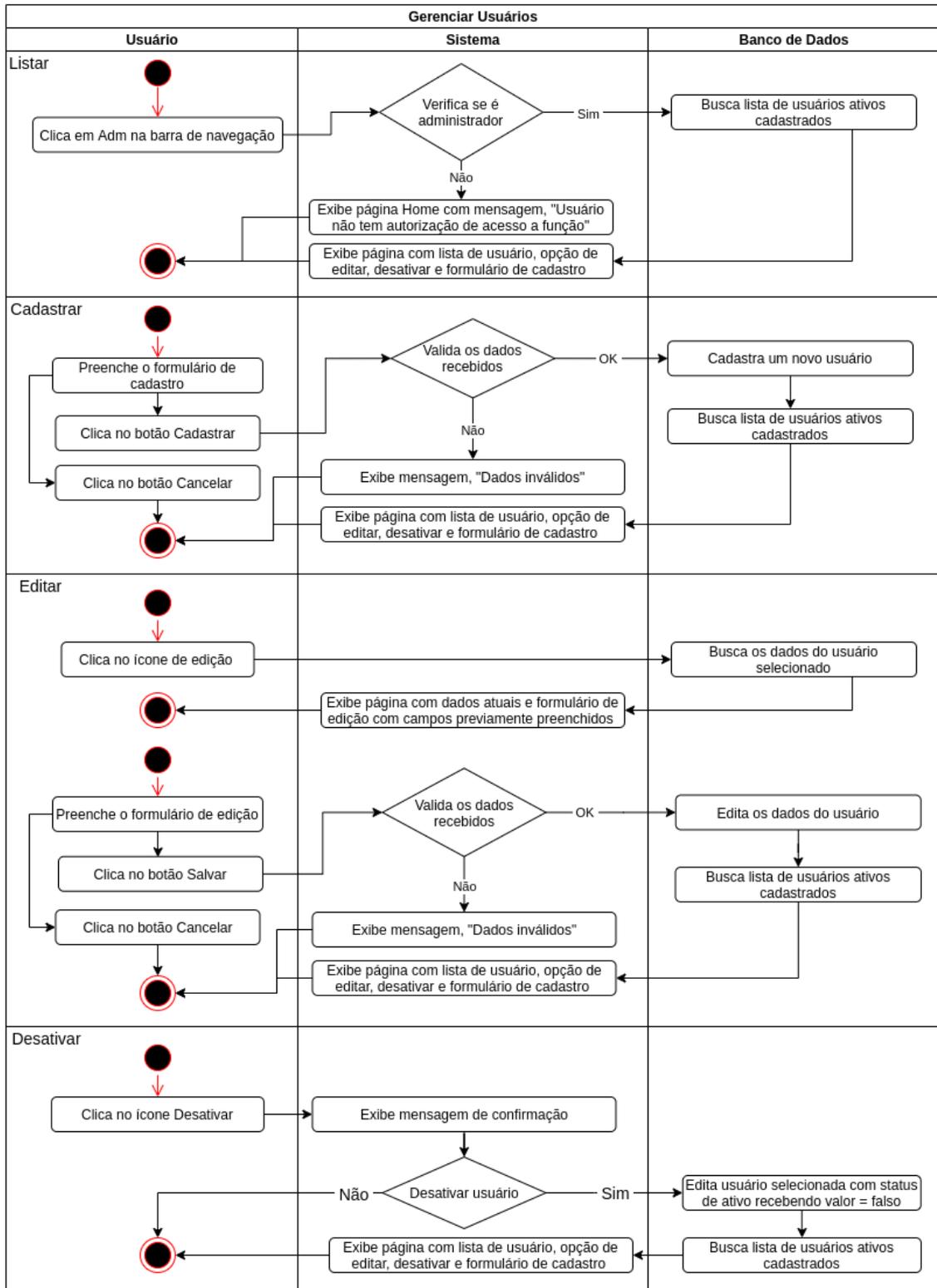


Figura 40: Diagrama de Atividades Gerenciar Usuários

6 RELATÓRIO DE TESTES

O relatório de testes traz os principais testes executados com a apresentação dos resultados de falhas e posteriormente a especificação da correção. Tem como objetivo documentar as ações tomadas para solucionar falhas especificadas de forma que correções possam ser utilizadas futuramente para atender outras aplicações ou partes de desta mesma aplicação em ampliações futuras sistema ou manutenções de rotina.

Tabela 1: Teste_01

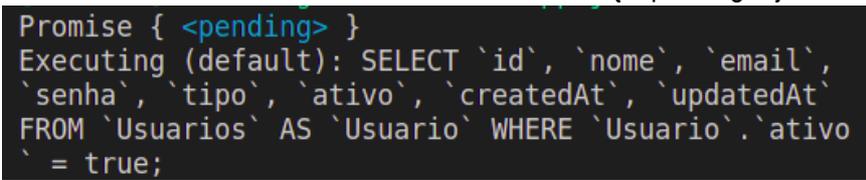
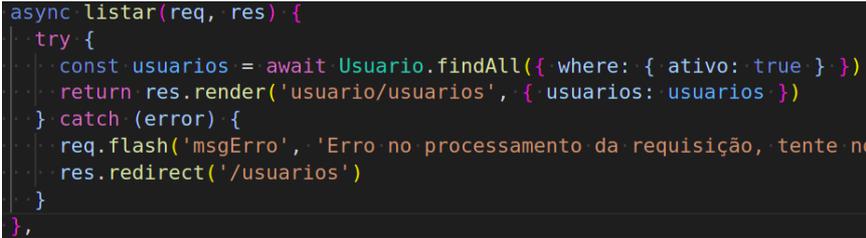
Identificação	Teste_01
Teste executado	<p>Teste da listagem de usuários ativos cadastrados</p> <p>Acessar a rota /usuarios</p> <p>Visualizar a página de usuários com lista de usuários ativos</p>
Requisito necessário	Servidor local rodando na porta 3000
Erro apresentado	<p>O sistema executa a query “SELECT” de forma correta mais não retornava os dados e no console retornava Promise { < pending > }</p>  <p style="text-align: center;">Figura 41: Console de falha do teste 1</p>
Correção	<p>Como a consulta é assíncrona, o sistema retornava o status da promise e não os dados da query, para corrigir foi implementado o método “await” antes da atribuição do objeto de retorno da query ser enviado a view.</p>  <p style="text-align: center;">Figura 42: Trecho de código da correção do teste 01</p>

Tabela 2: Teste_02

Identificação	Teste_02
Teste executado	<p>Teste de execução do relatório de paradas</p> <p>Preencher os campos no formulário de consulta de paradas.</p> <p>Clicar em consultar.</p> <p>Visualiza a lista de paradas na página de relatório de paradas</p>
Requisito necessário	Nenhum
Erro apresentado	<p>A tabela de paradas tem duas associações FK, sendo que o sistema identificava apenas uma e não retornava os dados da outra</p> <pre data-bbox="616 685 1305 1229"> const paradas = await Parada.findAll({ where: { motivo_id: { [Op.eq]: 4 } }, include: { association: 'motivo', association: 'usuario' } }) </pre> <p data-bbox="699 1240 1220 1272">Figura 43: Trecho de código do erro do teste 02</p> <pre data-bbox="572 1317 1342 1391"> TypeError: Cannot read property 'descricao' of undefined </pre> <p data-bbox="751 1402 1166 1433">Figura 44: Console do erro do teste 02</p>
Correção	<p>Foi identificado que a cláusula include do Sequelize deve receber um objeto association, e nos casos de necessidade de mais de uma associação deve receber um array com os objetos association.</p> <pre data-bbox="600 1547 1318 2002"> const paradas = await Parada.findAll({ where: { motivo_id: { [Op.eq]: 4 } }, include: [{ association: 'motivo' }, { association: 'usuario' }] }) </pre> <p data-bbox="632 2013 1286 2045">Figura 45: Trecho de código da correção do erro do teste 02</p>

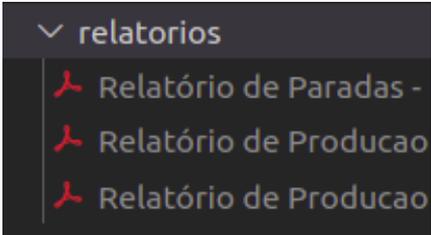
Tabela 3: Teste_03

Identificação	Teste_03
Teste executado	<p>Teste de transmissão de dados via socket-io</p> <p>Socket do servidor recebe um evento de um cliente indicando que por exemplo um item foi produzido. Então a aplicação persiste os dados de produção e reenvia os dados de produção atualizados aos clientes.</p> <p>No socket cliente, deve receber um objeto com os dados da produção atualizada.</p>
Requisito necessário	Nenhum
Erro apresentado	<p>Os dados não chegavam para o socket cliente que emitia o evento de “item produzido”, mas chegava para o cliente da view do dashboard.</p> <pre>//envia os dados para socket da view start enviaDados = (atualizado) => { socket.broadcast.emit('producaoDados', atualizado) console.log('socket enviou : ' + JSON.stringify(atualizado)) }</pre> <p>Figura 46: Trecho de código do erro do teste 03</p>
Correção	<p>Foi identificado que o método broadcast não envia um evento para o emissor de um evento antecessor, para corrigir isso foi implementado um emit direto da implementação do socket “io.emit”.</p> <pre>//envia os dados para socket da view start enviaDados = (atualizado) => { io.emit('producaoDados', atualizado) console.log('socket enviou : ' + JSON.stringify(atualizado)) }</pre> <p>Figura 47: Trecho de código da correção do erro do teste 03</p>

Tabela 4: Teste_04

Identificação	Teste_04
Teste executado	<p>Teste de formatação de datas</p> <p>Para formatação das datas do sistema foi implementado o módulo Date-fns, e amplamente utilizado para retornar a view as datas já formatadas.</p>
Requisito necessário	instalação do módulo Date-fns via NPM (npm install date-fns --save)
Erro apresentado	<p>O parâmetro que define a localidade do fuso horário não era compreendida e no console retornada “ReferenceError> ptBR is not defined”</p> <pre>datefns.format(valor.inicio, "HH:mm", { locale: ptBR })</pre> <p>Figura 48: Trecho de código do erro do teste 04</p>
Correção	<p>Foi identificado que é necessário fazer a importação da função ptBR</p> <pre>const { ptBR } = require('date-fns/locale');</pre> <p>Figura 49: Trecho de código da correção do erro do teste 04</p>

Tabela 5: Teste_05

Identificação	Teste_05
Teste executado	<p>Teste de emissão de relatórios em PDF</p> <p>Para emissão dos arquivos de relatórios em PDF foi criada uma pasta pública para armazenar os arquivos emitidos.</p>
Requisito necessário	
Erro apresentado	<p>Ao emitir o segundo relatório de uma consulta o arquivo era sobrescrito porque o título do PDF erra o mesmo do arquivo anterior</p>  <p>Figura 50: Trecho de código do erro do teste 05</p>
Correção	<p>Foi implementada uma função para acrescentar ao título do PDF um carimbo com o timestamp do momento da emissão do arquivo</p> <pre> let dataAgora = new Date() let carimbo = datefns.getTime(dataAgora) const pdf = printer.createPdfKitDocument(docDefiniton) pdf.pipe(fs.createWriteStream(path.resolve(` public/relatorios/Relatório de Paradas - \${carimbo}.pdf`))) pdf.end() </pre> <p>Figura 51: Trecho de código da correção do erro do teste 05</p>

7 CONSIDERAÇÕES FINAIS

O sistema Radar 4.0 é uma aplicação que automatiza a coleta de dados de linhas de produção e auxilia a liderança de chão de fábrica a tomar decisões para corrigir a produção do dia. Assim como fornece a gerência e diretoria dados sobre a saúde do negócio com a visualização em tempo real dos resultados obtidos. [1]

Foi desenvolvido toda estrutura de regra de negócio, testadas as tecnologias a serem aplicadas com protótipos de projetos operacionais mas não integrados, desenvolvida os formulários de gerenciamento de usuários com programação orientada a objetos para facilitar a replicação dos códigos para os outro crud's, montado o modelamento e a estruturação do banco de dados não relacional, implementado servidor de aplicação e desenvolvido uma api rest para tratar as rotas.

Será desenvolvido e implementado no próximo semestre, os demais crud's com formulários específicos, o gateway de campo que fará a leitura dos sensores que transmitem os dados com comunicação utilizado websocket, o dashboard com os gráficos de desempenho e página de login.

BIBLIOGRAFIA

- [1] Gazeta do Povo, url: <https://www.gazetadopovo.com.br/parana/industria-parana-crescimento/>, último acesso em 16/09/2019.
- [2] Developer Mozilla.org, url: <https://developer.mozilla.org/pt-BR/docs/Aprender/JavaScript>, último acesso em 16/09/2019.
- [3] Código Simples.net url : <https://www.opus-software.com.br/node-js/>, último acesso em 16/09/2019.
- [4] Lenon, Opus Software url: <https://www.devmedia.com.br/o-que-e-uml-e-diagramas-d-e-caso-de-uso-introduca-o-pratica-a-uml/23408>, último acesso em 13/11/2020.
- [5] Joe Zim, Conectando no Socket.IO, url: <https://imasters.com.br/devsecops/conectando-no-socket-io-o-basico>, último acesso em 13/10/2020.
- [6] Devmedia, url: <https://www.devmedia.com.br/modelo-entidade-relacionamento-mer-e-diagrama-entidade-relacionamento-der/14332> , último acesso em 16/09/2019.
- [7] Eduardo Figueiredo, Diagrama de Atividades DCC / ICEx / UFMG , url: https://homepages.dcc.ufmg.br/~amendes/GlossarioUML/glossario/conteudo/atividades/diagrama_de_atividades.html, último acesso em 07/10/2019.
- [8] Express, url: <https://expressjs.com/pt-br/>, ultimo acesso em 13/11/2020.
- [9] Home Host, O que é Mysql, url : <https://www.homehost.com.br/blog/tutoriais/mysql/o-que-e-mysql/> ultimo acesso em 13/11/2020,
- [10] RocketSite, Configurando o ORM Sequelize, url: [https://blog.rocketseat.com.br/nodejs-express-sequelize/#:~:text=O%20Sequelize%20%C3%A9%20um%20ORM,e%20linhas\)%20para%20objetos%20Javascript.](https://blog.rocketseat.com.br/nodejs-express-sequelize/#:~:text=O%20Sequelize%20%C3%A9%20um%20ORM,e%20linhas)%20para%20objetos%20Javascript.) , último acesso em 13/11/2020.