



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE AUTOMAÇÃO E SISTEMAS
CURSO DE GRADUAÇÃO DE ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Jhonata Vinicius Tridapalli

**Desenvolvimento de solução para monitoramento das atividades de negócio
de processos automatizados da Universidade Federal de Santa Catarina**

Florianópolis

2021

Jhonata Vinicius Tridapalli

**Desenvolvimento de solução para monitoramento das atividades de negócio
de processos automatizados da Universidade Federal de Santa Catarina**

Relatório final da disciplina DAS5511 (Projeto de Fim de Curso) como Trabalho Conclusão do Curso de Graduação em Engenharia de Controle e Automação da Universidade Federal de Santa Catarina em Florianópolis.

Orientador: Prof. Maurício Floriano Galimberti, Dr.
Supervisor: Prof. Maurício Floriano Galimberti, Dr.

Florianópolis

2021

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Tridapalli, Jhonata Vinicius
Desenvolvimento de solução para monitoramento das
atividades de negócio de processos automatizados da
Universidade Federal de Santa Catarina / Jhonata Vinicius
Tridapalli ; orientador, Maurício Floriano Galimberti, 2021.
94 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Centro Tecnológico,
Graduação em Engenharia de Controle e Automação,
Florianópolis, 2021.

Inclui referências.

1. Engenharia de Controle e Automação. 2. Monitoramento
de atividades de negócio. 3. Gerenciamento de processos de
negócio. I. Galimberti, Maurício Floriano. II. Universidade
Federal de Santa Catarina. Graduação em Engenharia de
Controle e Automação. III. Título.

Jhonata Vinicius Tridapalli

**Desenvolvimento de solução para monitoramento das atividades de negócio
de processos automatizados da Universidade Federal de Santa Catarina**

Esta monografia foi julgada no contexto da disciplina DAS5511 (Projeto de Fim de Curso) e aprovada em sua forma final pelo Curso de Graduação em Engenharia de Controle e Automação

Florianópolis, 12 de Julho de 2021.

Prof. Hector Bessa Silveira, Dr.
Coordenador do Curso

Banca Examinadora:

Prof. Maurício Floriano Galimberti, Dr.
Orientador e Supervisor
UFSC/CTC/INE

Prof. Ricardo José Rabelo, Dr.
Avaliador
UFSC/CTC/DAS

Prof. Marcelo De Lellis Costa de Oliveira, Dr.
Presidente da Banca
UFSC/CTC/DAS

Dedico este trabalho à minha família, colegas de curso e de vida e aos futuros membros do EAPn.

AGRADECIMENTOS

Primeiramente, à minha família e namorada, pelos incentivos constantes, oferecendo o suporte necessário durante as dificuldades, apoio e força ao longo de todo o período e principalmente, por confiarem e acreditarem no meu potencial quando a incerteza batia a porta. Um agradecimento especial a Deus, por tudo que já superei e alcancei.

Agradeço aos colegas de curso, que ao decorrer da graduação se tornaram amigos, compartilhando momentos de alegria durante as árduas maratonas de estudo e trabalho.

Agradeço aos meus amigos de infância, que se mantiveram comigo em todos os momentos mesmo com a longa distância que nos separa. Aqueles que proporcionaram momentos de diversão ao relembrar o passado e entusiasmo ao falar do futuro que nos aguarda.

Agradeço a Universidade Federal de Santa Catarina, Departamento de Automação e Sistemas, Departamento de Informática e Estatística, pelos ótimos profissionais, estrutura e ensinamentos.

Agradeço ao meu orientador Maurício Floriano Galimberti pelas oportunidades de estágio junto ao EAPn, por todas as experiências que contribuíram para meu crescimento profissional. Além do professor Antonio Carlos Mariani pela colaboração ao longo do trabalho e aos estagiários do EAPn.

RESUMO

O Escritório de Automação de Processos de Negócio (EAPn) é resultado de um projeto de pesquisa e extensão destinado à automação dos processos de negócio acadêmicos, fomentando na universidade a cultura de gestão por processos segundo a disciplina gerencial BPM (*Business Process Management*). As práticas previstas pela disciplina BPM determinam as atividades que garantem a gestão completa do processo visando sua máxima eficiência operacional, entre elas, o monitoramento. Motivado pela necessidade de aperfeiçoamento constante, é necessário identificar e avaliar constantemente o desempenho dos processos, avaliando sua execução e os resultados alcançados para melhorias e refinamentos. Para isso, ferramentas de monitoramento de processos são essenciais, principalmente para monitoramento de atividades de negócio, possibilitando aos gestores uma visão geral das atividades dos processos em execução. Com o objetivo de propor e desenvolver uma solução BAM (*Business Activity Monitoring*), em conformidade com a disciplina BPM, é originado o presente trabalho. A solução é caracterizada por um ambiente para acompanhamento de indicadores chaves de desempenho e métricas através de painéis gráficos, exibindo resumos e detalhamentos das atividades. Para isso, é realizada a integração com os processos automatizados e com os atuais sistemas utilizados no EAPn. Os resultados obtidos compreendem o desenvolvimento de uma página web utilizando o módulo de desenvolvimento de interfaces gráficas UI Designer do software Bonita.

Palavras-chave: Escritório de Automação de Processos de Negócio. EAPn. Gerenciamento de processos de negócio. BPM. Monitoramento de atividades de negócio. BAM. Monitoramento. Bonitasoft. Bonita. UI Designer.

ABSTRACT

The Business Process Automation Office (BPAO) is the result of a research and extension project aimed at the automation of academic business processes, fostering a culture of process management in the university according to the BPM (Business Process Management) management discipline. The practices foreseen by the BPM discipline determine the activities that guarantee the complete management of the process aiming at its maximum operational efficiency, among them, the monitoring. Motivated by the need for constant improvement, it is necessary to constantly identify and evaluate the performance of the processes, evaluating their execution and the results achieved for improvements and refinements. For this, process monitoring tools are essential, mainly for business activity monitoring, allowing managers an overview of the activities of the processes in execution. In order to propose and develop a BAM (Business Activity Monitoring) solution, in accordance with the BPM discipline, this work originated. The solution is characterized by an environment for monitoring key performance indicators and metrics through graphic panels, showing summaries and details of activities. For this, the integration with the automated processes and with the current systems used in the BPAO is carried out. The results obtained include the development of a web page using the UI Designer graphical user interface development module of the Bonita software.

Key-words: Business Process Automation Office. BPAO. Business process management. BPM. Business Activity Monitoring. BAM. Monitoring. Bonitasoft. Bonita. UI Designer.

LISTA DE FIGURAS

Figura 1 – Ferramentas de monitoramento em relação ao foco de análise.	16
Figura 2 – Página do Portal de Processos da UFSC para acompanhamento dos casos de processos abertos.	18
Figura 3 – Página do Portal de Processos da UFSC para acompanhamento das tarefas de casos de processos abertos.	19
Figura 4 – Estrutura de componentes do software Bonita.	26
Figura 5 – Fases do ciclo de vida BPM.	31
Figura 6 – Peso e classificação de indicadores de maturidade.	35
Figura 7 – Principais funções de sistemas BPM.	37
Figura 8 – Módulos de arquitetura BAM.	39
Figura 9 – Serviços e mecanismo de uma solução BAM.	41
Figura 10 – Modelo de melhoria contínua segundo a disciplina BPM através de BAM.	43
Figura 11 – Diagrama de casos de uso para o sistema proposto.	54
Figura 12 – Visão geral da arquitetura de componentes do software Bonita.	56
Figura 13 – Ambiente de desenvolvimento do UI Designer.	58
Figura 14 – Arquitetura de comunicação entre os componentes do Bonita e Metabase.	61
Figura 15 – Função para identificação de parâmetro de solicitação da extensão de API REST.	66
Figura 16 – Requisição HTTP POST para autenticação de usuário.	66
Figura 17 – Requisição HTTP GET para retorno do resultado das perguntas SQL.	67
Figura 18 – Exemplo de conteúdo de resposta da requisição da extensão de API.	68
Figura 19 – Arquivo de propriedades da Extensão de API.	69
Figura 20 – Interface da página definida no UI Designer Bonita.	71
Figura 21 – Exemplo de definição das propriedades dos elementos container, texto, .. título e tabela.	72
Figura 22 – Estrutura de elemento customizável do tipo gráfico de barras.	74
Figura 23 – Estrutura de elemento botão para operação de minimizar e maximizar janela.	76
Figura 24 – Estrutura JSON resultante do agrupamento dos casos de processo.	79
Figura 25 – Estrutura JSON de resposta do agrupamento das tarefas de processo.	80

Figura 26 – Estrutura JSON de resposta do agrupamento dos usuários para suas..... tarefas atribuídas.....	81
Figura 27 – Estrutura de navegação do Portal de Processos da UFSC.....	82
Figura 28 – Menu do Portal de Processos da UFSC.....	83
Figura 29 – Página de monitoramento BAM no Portal de Processos da UFSC.....	84
Figura 30 – Resultado do módulo de monitoramento de casos de processos.	85
Figura 31 – Resultado do módulo de monitoramento de tarefas.....	86
Figura 32 – Resultado do módulo de monitoramento de usuário.	86

LISTA DE TABELAS

Tabela 1 – Requisitos funcionais e não funcionais.....	53
Tabela 2 – <i>Endpoints</i> padrão de acesso aos recursos do Metabase	63
Tabela 3 – <i>Endpoints</i> específicos para acesso aos recursos desenvolvidos	64
Tabela 4 – <i>Endpoints</i> utilizados na página	78

LISTA DE ABREVIATURAS E SIGLAS

ABPMP	Association of Business Process Management Professionals
API	Application Programming Interface
BAM	Business Activity Monitoring
BDM	Business Data Model
BI	Business Intelligence
BPAO	Business Process Automation Office
BPM	Business Process Management
BPMM	Business Process Maturity Model
BPMS	Business Process Management System
CBOK	Common Body of Knowledge
CPM	Corporate Performance Management
CRM	Customer Relationship Management
DPA	Digital Process Automation
EAPn	Escritório de Automação de Processos de Negócio
ERP	Enterprise Resource Planning
HTTP	Hypertext Transfer Protocol
iBPMS	intelligent Business Process Management Suites
IDE	Integrated Development Environment
JS	JavaScript
JSON	JavaScript Object Notation
JDBC	Java DataBase Connectivity
KPI	Key Performance Indicator
REST	Representational State Transfer
SCM	Supply Chain Management
SQL	Standard Query Language
TI	Tecnologia da Informação
UI	User Interface
UML	Unified Modeling Language
URL	Uniform Resource Locator

SUMÁRIO

1	INTRODUÇÃO	15
1.1	DESCRIÇÃO GERAL DO PROBLEMA.....	17
1.2	MOTIVAÇÃO E JUSTIFICATIVA	19
1.3	OBJETIVOS	20
1.4	DELIMITAÇÃO.....	21
1.5	ESTRUTURA DO DOCUMENTO	21
2	ESCRITÓRIO DE AUTOMAÇÃO DE PROCESSOS DE NEGÓCIO	23
	(EAPN).....	23
2.1	AUTOMAÇÃO DE PROCESSOS	23
2.1.1	Bonita BPM	24
2.1.2	Portal de Processos UFSC	26
2.1.3	Monitoramento: Motivação sob a perspectiva da automação.....	27
2.2	MONITORAMENTO DE PROCESSOS	27
3	GERENCIAMENTO DE PROCESSOS DE NEGÓCIO	29
3.1	CICLO DE VIDA BPM	30
3.2	MONITORAMENTO.....	32
3.2.1	Importância do monitoramento para os modelos BPMM	33
3.2.2	O papel do monitoramento em sistemas BPM	36
4	MONITORAMENTO DE ATIVIDADES DE NEGÓCIO (BAM)	38
4.1	SISTEMAS BAM	39
4.2	BENEFÍCIOS DO BAM AO GERENCIAMENTO DE PROCESSOS DE	
	NEGÓCIO BPM	42
4.3	MEDIDAS DE DESEMPENHO PARA MONITORAMENTO DE	
	ATIVIDADES.....	44
4.4	BI: APOIO NO SUPORTE À DECISÃO BAM.....	45
5	CONCEPÇÃO DA SOLUÇÃO.....	47
5.1	ESPECIFICAÇÃO BAM	47
5.1.1	Serviços	47
5.1.2	Módulos de monitoramento	48
5.1.2.1	Indicadores de desempenho	49
5.2	ESPECIFICAÇÃO DO SISTEMA.....	51
5.2.1	Requisitos de Sistema	51
5.2.2	Modelagem do sistema.....	53
6	SOFTWARE BPMS BONITA: RECURSOS PARA DESENVOLVIMENTO ..	56

6.1	ESTRUTURA DE DADOS.....	57
6.2	PÁGINAS E APLICATIVOS	57
6.3	EXTENSÃO DE API REST	59
7	DESENVOLVIMENTO	61
7.1	REQUISIÇÃO DE DADOS DO METABASE	62
7.1.1	Extensão de API REST: Requisição ao Metabase	64
7.1.1.1	Arquivo para definição da lógica	65
7.1.1.2	Arquivo de propriedades	68
7.2	ESTRUTURAÇÃO DA PÁGINA BAM	70
7.2.1	Layout da interface.....	70
7.2.2	Elementos de interação	72
7.2.2.1	Gráficos	73
7.2.2.2	Botão para minimizar e maximizar painel	75
7.2.2.3	Elemento gráfico de carregamento.....	76
7.2.3	Funcionalidades operacionais	77
7.2.3.1	Solicitação HTTP	77
7.2.3.2	Funções JavaScript para estruturação dos dados	78
7.2.3.3	Controle de acessibilidade	81
7.3	INCORPORAÇÃO DA PÁGINA AO PORTAL DE PROCESSOS	82
8	RESULTADOS.....	84
9	CONCLUSÃO	88
9.1	PERSPECTIVAS FUTURAS	88
	REFERÊNCIAS.....	90

1 INTRODUÇÃO

Com o aumento competitivo e a necessidade de gerar cada vez mais valor para o negócio, as empresas precisam de respostas muito mais rápidas para tomada de decisões imediatas. A exigência de melhorias na execução dos processos e tarefas associadas é imprescindível para se alcançar a excelência gerencial e operacional, agregando maior valor aos seus produtos e serviços.

A necessidade de melhoria na gestão do negócio desencadeado pelo emaranhado de processos ineficientes torna inevitável a busca por novas práticas de gerenciamento. Neste contexto, surgiu o conceito de gerenciamento por processo, chamado de BPM (*Business Process Management*), definido por um conjunto integrado de capacidades corporativas relacionadas ao alinhamento entre governança, métodos, tecnologia, pessoas e cultura (BROKCKE & ROSEMAN, 2014).

Mesmo sendo um grande desafio implementar mudanças organizacionais englobando todas as capacidades, as práticas previstas pela disciplina BPM vêm se tornando cada vez mais utilizadas, afinal, a eficiência de uma empresa está diretamente ligada à qualidade de seus processos. Associando a estratégia do negócio e rendimentos esperados a partir de processos melhores, conseqüentemente é obtido resultados mais assertivos ao final do processo. Assim os clientes terão uma boa percepção de valor e ficarão mais satisfeitos (ALMEIDA, 2018).

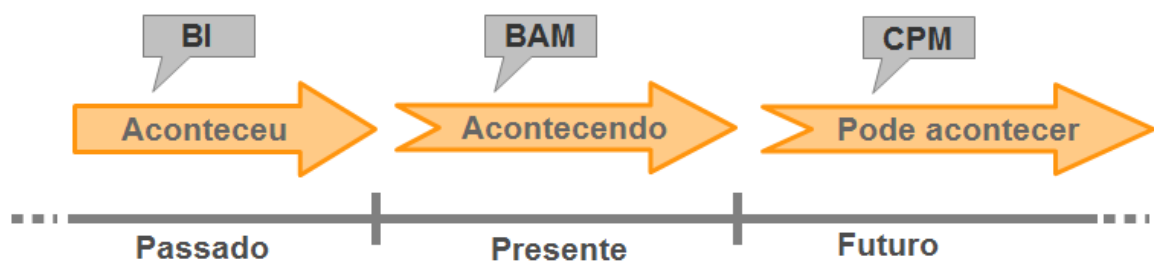
Com o objetivo de sistematizar e estruturar a gestão por processos nas organizações, a abordagem gerencial BPM é instituída de práticas e princípios. Segundo a *Association of Business Process Management Professionals* (ABPMP) através do Guia BPM CBOK (*Common Book of Knowledge*), BPM é compreendido pela prática das atividades de identificar, projetar, executar, documentar, medir, monitorar, controlar e promover melhorias nos processos de uma organização (ABPMP, 2013).

Para oferecer os serviços mencionados no ambiente acadêmico da UFSC, foi estruturado o Escritório de Automação de Processos de Negócio (EAPn). Atualmente seus esforços estão direcionados na modelagem e automação de processos, sendo estes, disponíveis aos usuários da universidade através de um Portal de Processos. Visando atuar no controle e refinamento dos processos, o EAPn começou a partir de 2020 a adotar dentre suas atividades, o monitoramento.

O ambiente operacionalizado e disponível atualmente é composto por um sistema de *Business Intelligence* (BI), possibilitando a análise de indicadores de desempenho através de *dashboards*. Essa abordagem permite o armazenamento e organização de um grande volume de dados, gerando informações demonstradas através de indicadores de desempenho.

Além de sistemas BI, demais ferramentas oferecem suporte para monitoramento e gestão do negócio. Em seus diferentes contextos, podem-se definir 3 conjuntos de ferramentas mediante as características das informações utilizadas para análise e o foco de seus resultados (ver Figura 1).

Figura 1 - Ferramentas de monitoramento em relação ao foco de análise.



Fonte - MORTARI (2014).

Conforme destacado pela imagem, a relação temporal envolvida nos sistemas e suas agregações possibilitam análises multidimensionais, gerando uma visão completa para gerenciamento das operações do negócio. Complementado, cada um dos sistemas mencionados sequencialmente na Figura 1 possibilitam respectivamente: análise histórica de processos finalizados, análise atual de atividades de processo em execução e análise futura para previsão e planejamento visando objetivos a serem traçados.

Para integralizar o ambiente de monitoramento atual e monitorar as atividades em execução dos processos automatizados, originou-se a proposição deste trabalho. O foco de interesse é respectivo ao desenvolvimento de uma solução BAM, possibilitando a tomada de ações para refinamento dos processos, visando ofertar serviços mais qualificados aos usuários e eficientes para a universidade.

1.1 DESCRIÇÃO GERAL DO PROBLEMA

Atualmente não se tem disponíveis recursos específicos para realizar o monitoramento das atividades de casos abertos dos processos automatizados. A ferramenta de automatização Bonita e o Portal de Processos disponibilizam em sua plataforma dados para acompanhamento dos casos de processo¹, porém com as limitações:

- Disponibilidade geral dos dados somente para usuários administradores, não acessível em totalidade aos usuários donos de processo².
- Informações descentralizadas, dificultando o acesso e usabilidade do sistema, agilidade, comunicação organizacional e gerenciamento.
- Não fornece os dados relevantes e necessários para o acompanhamento das atividades de modo a possibilitar tomadas de decisões.
- Formato de visualização em tabelas em uma interface não intuitiva, sem nenhum modo de agregação e exibição gráfica para resumos detalhados entre casos de processos e suas respectivas tarefas.

No formato atual, apenas informações básicas de controle estão disponíveis, com operações de filtragens (ver Figura 2) e não disponíveis aos donos de processo. Os privilégios associados compreendem somente a visualização dos casos de processo em que o usuário da sessão teve participação. Dessa forma, não garante nenhum privilégio operacional ao dono de processo de acordo com a arquitetura padrão convencionada.

¹ Um caso de processo corresponde a uma nova instância do processo, com identificador único.

² Dono de processo é a pessoa que possui a responsabilidade pelo gerenciamento do processo, garantindo seus requisitos de desempenho.

Figura 2 - Página do Portal de Processos da UFSC para acompanhamento dos casos de processos abertos.

Portal de Processos :: Iniciar Processos Consultar Processos Tarefas Painel Gerencial

UFSC /CTC /EAPn - Escritório de Automação de Processos de Negócio Bem vindo **Jhonata Vinicius Tridapalli** Sair

Casos abertos

Filtro

Nome do processo Todos **Data inicial** Insira uma data **Instância do processo** Id do Caso

Iniciado por Pesquisar... **Data final** Insira uma data

Lista de casos abertos

Id	Nome do Processo	Versão	Iniciado por	Data de Início
23056	P03 - Estacionamento	1.3	Jhonata Vinicius Tridapalli	25/04/2021 16:48

1 - 1 de 1

Fonte - Autoria própria.

Do ponto de vista das atividades, definidas pelas tarefas, as informações disponibilizadas são específicas somente a um caso de processo, conforme mostrado na Figura 3. Dessa forma, não é possível estabelecer relações entre tarefas de diferentes casos, inviabilizando o acompanhamento, supervisão e operações associadas para gerar indicadores e análises.

Figura 3 - Página do Portal de Processos da UFSC para acompanhamento das tarefas de casos de processos abertos.

The screenshot displays the 'Portal de Processos' interface. At the top, there is a navigation bar with options: 'Portal de Processos', 'Iniciar Processos', 'Consultar Processos', 'Tarefas', and 'Painel Gerencial'. Below this, a blue header bar contains the text 'UFSC / CTC / EAPn - Escritório de Automação de Processos de Negócio' on the left, 'Bem vindo Jhonata Vinicius Tridapalli' in the center, and 'Sair' on the right. The main content area includes a 'Voltar' button on the left and a 'Visão geral' button on the right. The central part of the page shows process details: 'Número da instância do processo: 23056 - Processo: P03 - Estacionamento', 'Versão do processo: 1.3', 'Iniciado em: 25/04/2021 16:48', 'Iniciado por: Jhonata Vinicius Tridapalli', 'Última atualização: 25/04/2021 16:48', and 'Estado: started'. Below these details are two panels: 'Tarefas para fazer' and 'Tarefas concluídas'. The 'Tarefas para fazer' panel contains a table with one row: 'T10 - Avaliar Requisição de Estacionamento' with the description 'Nenhuma descrição'. The 'Tarefas concluídas' panel shows the message 'Não há dados.'

Fonte - Autoria própria.

Essas limitações não permitem aos gestores, de forma prática, acompanhar o andamento das tarefas dos casos em execução. Assim, não é possível identificar eventuais problemas no fluxo de execução, gargalos, desempenho de colaboradores e, principalmente, fundamentar tomada de decisões para melhorias e refinamentos.

1.2 MOTIVAÇÃO E JUSTIFICATIVA

Dentre as práticas BPM está previsto a utilização de ferramentas que possibilitem a organização monitorar os processos, permitindo o acompanhamento de indicadores e métricas que quantifiquem resultados que possam refletir seu desempenho. Dessa forma, é possível conduzir avaliações e análises que proporcionam aos gestores a tomada de decisões gerando melhorias que auxiliem a instituição a atingir os seus objetivos.

Em projetos de implementação BPM, geralmente o foco principal é a automatização de processos, devido aos benefícios entregues a curto prazo. Porém, para validar e demonstrar que as ações de automação tomadas representam um cenário ótimo de desempenho para a organização, é inevitável a utilização de mecanismos de monitoramento. Estes mecanismos lidam com o ajuste de recursos

para garantir os objetivos do processo através de medições e avaliação de desempenho (MORAIS *et al.*, 2014).

Através do monitoramento de atividades de negócio é possível acompanhar em tempo real o desempenho dos processos, visando a alta performance de execução. O BAM possibilita decisões rápidas e efetivas, tornando os problemas visíveis rapidamente, gerando relatórios, alertas e eventos de acordo com o resumo situacional do momento.

Associado aos resultados do BI, pode-se estabelecer os requisitos mínimos que devem ser atendidos e verificados no monitoramento BAM. Dessa forma, através dos indicadores de desempenho históricos sobre determinada atividade é possível estabelecer quais as metas a serem seguidas, apontando se a situação atual do negócio está de acordo com o planejamento estratégico definido.

Entregando informações operacionais através de indicadores de desempenho, é estabelecido aos gerentes maior conscientização do negócio. Com o acompanhamento ativo do fluxo de processo e operações relacionadas, pode-se identificar focos de ineficiência e oportunidades, sendo sinalizadas e visíveis ao próprio dono do processo, podendo tomar as ações cabíveis.

Com o aumento do número de execução dos casos de processos automatizados, e principalmente com o elevado número dos casos abertos simultaneamente, o controle sobre as atividades torna-se cada vez mais essencial. À medida que as demandas sobre os colaboradores se intensificam, as atividades se tornam mais suscetíveis a atrasos e falhas, visando manter a qualidade dos serviços da universidade, ferramentas para monitoramento e controle como o BAM são indispensáveis.

1.3 OBJETIVOS

Pretende-se desenvolver uma solução BAM para integração ao Portal de Processos da UFSC, operacionalizando um ambiente de acompanhamento das atividades dos processos de negócio automatizados pelo EAPn. A solução deve atender os requisitos BAM para construção de uma interface gráfica, incluindo a definição de indicadores de desempenho e métricas específicas e aplicáveis a todos os processos automatizados. Assim, o dono de processo deve ser capaz de facilmente visualizar e analisar o resumo das atividades em execução.

Além do caráter de desenvolvimento, se tem por objetivo a realização de um estudo sobre requisitos BAM em relação a disciplina gerencial BPM. Assim, é apresentada uma contextualização sobre ferramentas e sistemas, demonstrando os conceitos utilizados e situando a solução desenvolvida ao escopo de conhecimento da área.

1.4 DELIMITAÇÃO

A definição da solução BAM adotada será específica aos recursos necessários para monitoramento visual das atividades de negócio em execução. Dessa forma, recursos de geração de relatórios documentais, tratamento de eventos e alertas previstos pela categoria de software BAM não serão considerados para implementação, não sendo uma necessidade latente.

Serão definidos apenas requisitos funcionais sobre *dashboards*, não cobrindo requisitos mais avançados que devem ser considerados na construção dos painéis e das informações a serem disponibilizadas aos usuários. Assim, não foi realizado nenhum estudo pormenorizado seguindo metodologias previstas na literatura para definição de indicadores de desempenho e o respectivo formato de exibição.

1.5 ESTRUTURA DO DOCUMENTO

O trabalho está organizado em 9 capítulos, as informações apresentadas são estruturadas da seguinte forma:

No capítulo 2 são apresentadas as informações do escritório na qual este trabalho está incluso, relatando os serviços prestados à universidade e a infraestrutura na qual está inserido.

No capítulo 3 é definido os conceitos da disciplina gerencial BPM e como estes se associam às atividades de monitoramento, descrevendo a sua importância perante os serviços do EAPn.

No capítulo 4 é descrita a fundamentação teórica acerca do monitoramento de atividades de negócio BAM, elucidando o contexto de aplicabilidade destes sistemas através de arquiteturas, recursos e serviços englobados, definindo seu relacionamento com a disciplina BPM.

No capítulo 5 é abordado as considerações acerca da solução a ser desenvolvida, descrevendo as características e requisitos a serem atendidos.

No capítulo 6 é apresentado os recursos do software Bonita utilizados para desenvolvimento, descrevendo suas especificidades para o desenvolvimento da solução.

No capítulo 7 é abordado as etapas de desenvolvimento, descrevendo como foram implementadas as funcionalidades definidas, programação, configurações dos elementos utilizados e demais características intrínsecas ao desenvolvimento.

No capítulo 8 são apresentados os resultados obtidos, exibindo a página BAM para integração ao Portal de Processos da UFSC.

No capítulo 9 são apresentadas as conclusões decorrentes do processo de concepção e desenvolvimento da solução BAM, descrevendo futuras implementações para expandir o atual trabalho dentro do projeto do EAPn.

2 ESCRITÓRIO DE AUTOMAÇÃO DE PROCESSOS DE NEGÓCIO (EAPn)

O EAPn é resultado do Projeto de Pesquisa “Estudo de Viabilidade e Proposição de um Modelo de Escritório de Automação de Processos de Negócios” iniciado em julho de 2017 (GALIMBERTI *et al.*, 2019) idealizado pelo professor doutor Maurício Floriano Galimberti junto ao Centro Tecnológico – CTC/UFSC. O intuito do projeto é pesquisar a viabilidade e os benefícios da prática por gestão de processos em um ambiente acadêmico, gerando um modelo para estruturação e implantação de um núcleo de gerenciamento por processo centrado em automação aplicável a Instituições de Ensino Superior.

Alinhado aos fundamentos da disciplina gerencial BPM, atualmente o EAPn oferece serviços de mapeamento, modelagem, documentação e automação de processos seguindo um método desenvolvido e validado pelo escritório. O método é resultado de desenvolvimento do Prof. Maurício desde 2010, o qual foi adaptado para o Trabalho de Conclusão de Curso (TCC), feito pelo então graduando em Sistemas da Informação Hugo Piovesan Gonçalves, intitulado “Guia para Modelagem e Automação de Processos de Negócios Acadêmicos: estudos de caso com processos da UFSC” de 2016 (GONÇALVES, 2016). Foi desenvolvido um guia para servir como base para orientar a modelagem e a automação dos processos de negócios acadêmicos da universidade.

O projeto de modelo do EAPn foi iniciado a partir desse guia, oferecendo serviços centrados na automação com o intuito de melhorar o gerenciamento de processos de negócios da UFSC. Com um núcleo específico, que tenha o conhecimento adequado sobre a disciplina BPM, é possível propiciar melhor gestão sobre os processos e disseminar de forma mais objetiva as práticas de gerenciamento propostas.

2.1 AUTOMAÇÃO DE PROCESSOS

As tecnologias de automatização devem auxiliar no desenvolvimento de soluções caracterizadas pela substituição de atividades manuais por aplicações. Além disso, devem possibilitar a criação de interfaces sistêmicas, integrações entre diferentes bases de dados, fluxos de trabalho mais ágeis e utilização assertiva de informações. Com isso, os benefícios são diversos, aperfeiçoando os processos de negócios e contribuindo para a transformação digital das empresas.

A automação apresenta-se como um meio para a melhoria dos processos e extremamente vantajosa se for alinhada às práticas BPM. Idealmente, os projetos de implementação de BPM compreendem as etapas de análise (AS-IS), redesenho (TO-BE) e implementação (TO-DO) (GALIMBERTI *et al.*, 2019), sendo a implementação geralmente caracterizada pela utilização de tecnologias que possibilitem a automação.

Para isso, são utilizadas ferramentas de gerenciamento de processos denominadas BPMS (*Business Process Management System*) para a automação do fluxo de informação e ações de um processo, por meio de:

- Ferramentas de modelagem e análise de processos;
- Mecanismos de regras de negócios e de fluxo de trabalho;
- Ferramentas de simulação e teste.

Através do BPMS é possível tornar processos de negócios mais eficientes e adaptáveis, porém seu conjunto de ferramentas condiciona o desenvolvimento da automação, exigindo que os BPMS se adequem constantemente às novas tecnologias do mercado.

2.1.1 Bonita BPM

No EAPn utiliza-se o software Bonita, desenvolvido pela empresa Bonitasoft, o qual oferece uma plataforma de gerenciamento de processos de negócio centrada na automação de processos. O software apresenta versões gratuitas e de código aberto mantendo seus recursos principais de modelagem e automação, justificando sua utilização.

Através de uma interface simples e intuitiva, auxilia na modelagem de processos utilizando-se da notação BPMN (*Business Process Model and Notation*), representando o processo através de diagramas e fluxos. Este tipo de representação auxilia na visualização e promove uma melhor compreensão do processo. Desta forma, mudanças no percurso do processo podem ser feitas de maneira mais simplificada.

Em relação a automação, o Bonita é uma plataforma DPA (*Digital Process Automation*) que permite criar aplicativos com base em processos, disponibilizando os recursos tradicionais de sistemas BPM. É caracterizado por alinhar práticas de modelagem, operação, automação e melhorias de processos de negócios com

tecnologias de inteligência operacional para otimizar a orquestração³ e padronizar a experiência do usuário.

O conceito DPA é uma especificação das tecnologias de um BPMS. No caso do Bonita, é associado principalmente ao uso de aplicativos com foco em melhorar a eficiência e experiência do cliente (CLAIR, 2019). Independentemente da nomenclatura utilizada e do contexto associado às ferramentas destinadas a aspectos da automação, estas visam promover principalmente: fluxos de trabalhos automáticos, interfaces interativas e inteligentes, colaboração entre usuários, adaptação à mudança, rastreabilidade e transparência nas interações feitas entre usuário e sistema.

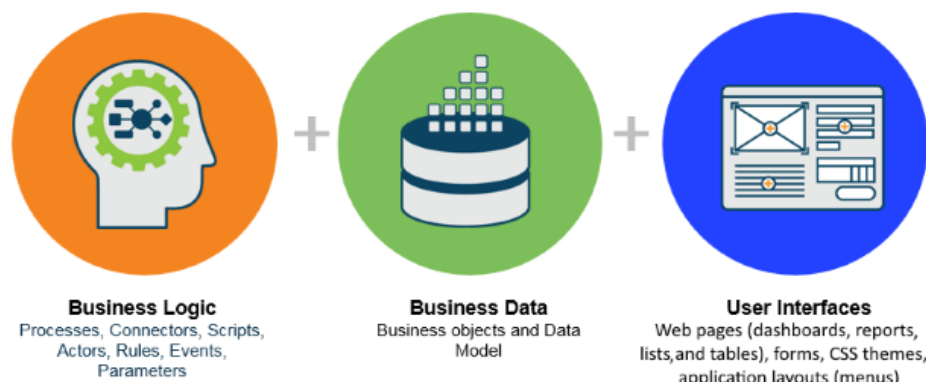
O Bonita oferece um ambiente completo de desenvolvimento, apresentado em 3 módulos:

- **Bonita Studio:** Ambiente de desenvolvimento dos diagramas de fluxo, definição de bases de dados, regras de negócio, pessoas envolvidas, conexão com sistemas externos.
- **Bonita UI Designer:** Ambiente composto de recursos para construção de formulários digitais para preenchimento dos dados definidos no Studio. Além disso, disponibiliza ao usuário ferramentas gráficas, de estruturação e codificação para desenvolver interfaces personalizadas.
- **Bonita Portal:** É um Portal que permite ao usuário simular os processos e executar as tarefas envolvidas. Possibilita gerenciar os processos, disponibilizando um ambiente para administrador.

Cada módulo é definido por uma IDE específica, possibilitando o desenvolvimento e utilização da aplicação. Segundo o colaborador e desenvolvedor técnico da Bonitasoft Pierrick Voulet, o desenvolvimento de aplicações no Bonita se resume ao padrão: modelo, controle e visualização, sendo associado respectivamente a estrutura de dados, lógica de negócios e interface de usuário responsiva (ver Figura 4) (VOULET, 2018).

³ Orquestração refere-se à relação entre atividades e eventos do processo, promovendo a compreensão da sequência de trabalho.

Figura 4 - Estrutura de componentes do software Bonita.



Fonte – VOULET (2018).

Conforme verificado na definição de interface de usuário da Figura 4, uma ferramenta BPMS completa é constituída de ferramentas que possibilitam o monitoramento dos processos através de painéis de indicadores. Neste caso, estas são disponíveis apenas nas versões *enterprise* (não gratuita) do Bonita. Porém, através de recursos avançados disponibilizados na versão *community* (gratuita) é possível efetuar a integração entre sistemas e assim, desenvolver um ambiente personalizado que possibilite atingir os fins propostos pelo trabalho.

2.1.2 Portal de Processos UFSC

O Portal de Processos é a plataforma de acesso aos processos automatizados no Bonita BPM. Através dele os usuários da universidade conseguem iniciar processos, executar tarefas e acompanhar o andamento das atividades. O ambiente foi desenvolvido utilizando o módulo UI Designer do Bonita através do recurso *Application Page* para criação de páginas personalizadas, possibilitando a criação de um ambiente que substitua o Bonita Portal e que atenda às necessidades específicas de cada negócio, constituindo um aplicativo.

As páginas são integradas aos dados de negócio dos processos, possibilitando a recuperação direta através de API (*Application Programming Interface*)⁴. No

⁴ API é um conjunto de rotinas para acesso a um software, aplicação ou plataforma web para utilização de seus serviços e recursos.

aplicativo são definidos fatores de interação, lógica e *layout*, totalmente customizáveis e rapidamente adaptáveis às necessidades do negócio.

Com o objetivo de tornar esse ambiente ainda mais completo, será desenvolvida uma página de aplicação que possibilite aos gestores dos processos o acompanhamento de informações através de *dashboards*, resolvendo os problemas das limitações de monitoramento do Bonita. Aspectos técnicos e conceituais usados para concepção e desenvolvimento da solução, requisitos de serviços BAM e requisitos do sistema serão abordados no capítulo 5.

2.1.3 Monitoramento: Motivação sob a perspectiva da automação

Com a automação de processos é possível se obter muitos benefícios, principalmente quando aplicado a organizações que não possuem em sua cultura práticas gerenciais como o BPM. A automação alinhada ao gerenciamento orientado por processos reduz possíveis focos de ineficiência, que a longo prazo podem desencadear problemas maiores. Além disso, visa reduzir os custos de execução, recursos financeiros e humanos possibilitando que estes sejam distribuídos para fins mais críticos (GALIMBERTI *et al.*, 2019).

Motivado por todos os possíveis benefícios mencionados, espera-se que após a automação, a empresa presencie um cenário de melhoria no desempenho de seus processos. Porém, como verificar e analisar de modo objetivo e quantitativo o desempenho associado à execução do processo? Neste contexto, é apresentada a proposta do trabalho, utilizando o monitoramento das atividades dos processos automatizados como uma ferramenta para medição de desempenho.

Automatizar processos não é uma demanda única que se encerra assim que os processos atuais são automatizados. É preciso acompanhá-los de forma contínua para que melhorias sejam implementadas sempre que novas necessidades sejam identificadas. Para isso, utilizam-se ferramentas de monitoramento que possibilitem análises sistemáticas.

2.2 MONITORAMENTO DE PROCESSOS

A partir de março de 2020 o EAPn iniciou projetos para desenvolvimento de soluções para monitoramento. O projeto inicial denominado “Integração de *dashboards* ao Portal de Processos da UFSC para monitoramento do desempenho

dos processos de negócio automatizados pelo EAPn” foi o precursor, sendo desenvolvido pelo próprio autor deste trabalho. O presente documento constitui o segundo projeto na área, visando complementar o ambiente de monitoramento já existente.

Atualmente o Portal de Processo conta com uma página específica para monitoramento através de painéis BI desenvolvidos utilizando o software Metabase. Através deste, os donos de processo podem acompanhar indicadores de desempenho e avaliar constantemente os processos de negócio. As implementações tiveram como referência um mapeamento e revisão sistemática da literatura executada pelo EAPn, com foco em indicadores chaves de desempenho genéricos para avaliações de processos de negócio que adotam práticas BPM.

Característica e recursos do Metabase não serão abordados, visto que o foco do trabalho não compreende a utilização do ambiente para criação de painéis. No capítulo 7 é especificado apenas as características de interesse sobre a plataforma para obtenção de informações dos painéis existentes.

3 GERENCIAMENTO DE PROCESSOS DE NEGÓCIO

Um processo de negócio é uma sequência de atividades ou tarefas (objeto de interesse do BAM) estruturadas com o objetivo de entregar algum resultado de valor para o cliente. Com a evolução e modernização, as organizações se tornaram compostas por inúmeros processos de negócio, motivando transformações em suas práticas gerenciais. Dessa forma, a abordagem de gestão por processos de negócios BPM torna-se uma solução, compreendendo práticas que possibilitem o aumento da produtividade e redução de custos.

O gerenciamento de processos de negócios visa o aperfeiçoamento contínuo dos processos, através de uma visão sistêmica da organização, fazendo uma gestão voltada inteiramente a processos e não a silos funcionais⁵ (departamentos). Essa nova visão permite um acompanhamento total do processo independentemente dos departamentos associados à produção do bem ou serviço, possibilitando enxergar os processos ponta-a-ponta.

Ao fazer um pedido de um produto em uma plataforma de venda online até que o mesmo seja entregue na sua residência, o mesmo percorre inúmeros setores da organização que o produz, cada um com suas respectivas tarefas, seja comercial, produção ou transporte. No momento que as tarefas do setor são realizadas, automaticamente a responsabilidade do produto se torna do próximo setor, sendo que geralmente, não há compartilhamento de conhecimento entre ambos. Desmistificando essa cultura de gestão, acompanhar um processo ponta-a-ponta, transversal a diferentes silos funcionais, envolve o todo de uma organização, acompanhando o andamento do pedido até o mesmo ser entregue e assim, gerar valor tangível ao cliente.

Com a prática da disciplina gerencial BPM é possível conhecer os processos executados, medir e gerenciar, com a finalidade de sistematizar, facilitar e melhorar as operações envolvidas e, conseqüentemente, gerar benefícios para a organização com o aumento da produtividade e o melhor aproveitamento dos recursos.

⁵ Os silos funcionais são relacionados aos setores individuais de uma empresa, responsáveis por um conjunto de atividades específicas.

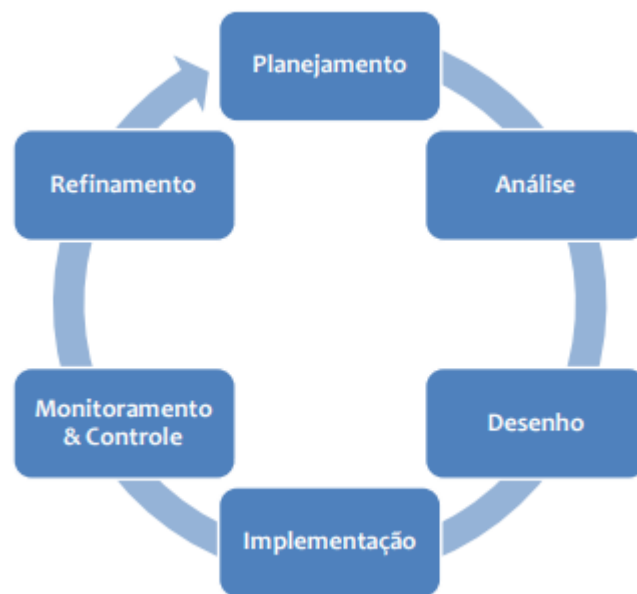
3.1 CICLO DE VIDA BPM

A gestão por processo BPM é fundamentada no princípio de melhoria contínua, implicando em um comprometimento permanente da organização. Dessa forma, deve ser garantido o alinhamento dos processos à estratégia organizacional, possibilitando alcançar os objetivos traçados.

O ciclo de vida BPM consiste em um conjunto de fases com uma sequência lógica que possibilitam o acompanhamento de todas as etapas de implementação do gerenciamento por processos, sendo semelhante às ações do ciclo PDCA: planejar (*plan*), fazer (*do*), checar (*check*) e agir (*act*) (ABPMP, 2013). Com objetivos semelhantes, os ciclos têm o intuito de promover uma visão focada nas causas e não somente nas consequências, resultando em uma ferramenta de análise e mudança de processos de constante planejamento. Assim, busca-se identificar oportunidades de melhorias promovendo as ações necessárias para atingir os fins desejados.

Alinhado ao Guia CBOK (Guia para Gerenciamento de Processos – Corpo Comum de Conhecimento de ABPMP), a disciplina gerencial BPM propõe um ciclo de vida completo de 6 fases, orientado à necessidade de cada organização e variável de acordo com a maturidade do cenário de utilização de BPM. A Figura 5 apresenta as fases que, segundo o BPM CBOK (2013), exemplificam o modelo típico de ciclo BPM.

Figura 5 - Fases do ciclo de vida BPM.



Fonte – ABPMP (2013).

Resumidamente, cada fase pode ser descrita por:

- 1. Planejamento:** Analisar informações dos processos disponíveis identificando de que maneira estão alinhados aos serviços prestados. Nesta etapa é realizado o diagrama do processo atual, conhecido também por modelo AS-IS (Como é).
- 2. Análise:** É realizada uma análise do atual processo de negócio, compreendendo um estudo sobre todo o ambiente do negócio.
- 3. Desenho:** É realizado o diagrama do processo, criando um modelo com o objetivo de representar a situação futura do processo, conhecido também por modelo TO-BE (Como será). Esse modelo é uma projeção concebida a partir da análise do processo AS-IS.
- 4. Implementação:** Consiste na execução dos processos que foram identificados, mapeados e redesenhados nas fases anteriores, atribuídos de serviços de automação e possíveis mudanças organizacionais.

5. **Monitoramento e controle:** Obter informações resultantes da execução dos processos que possibilitem estabelecer medidas de desempenho e possíveis focos de melhorias.
6. **Refinamento:** Mediante os resultados da fase anterior associado às metas estabelecidas e melhorias propostas, são executadas as alterações nos processos buscando otimizações e benefícios à organização.

Considerando o processo de adequação da universidade às práticas de gerenciamento propostas, e devido ao pouco tempo transcorrido desde a implantação e disponibilização dos serviços do EAPn, o gerenciamento dos processos atualmente contempla, primordialmente, as fases 1, 2, 3 e 4. Para essas fases, foram desenvolvidos modelos com um conjunto de diretrizes sistematizando a metodologia aplicada, que foram validados e que serviram como guia para a adequação de 6 processos modelados e automatizados conforme as práticas BPM, estando em operação e disponíveis para execução.

Visando estabelecer o ciclo contínuo com todas as fases propostas pelo gerenciamento BPM, o BAM é integrado para complementar o ambiente de monitoramento existente. Assim, com cada vez mais recursos disponíveis, os trabalhos realizados nas fases anteriores não perderão utilidade com o transcorrer do tempo, visto a contínua adequação dos processos com a realidade volátil das organizações. Por isso, a gestão BPM engloba ações de monitoramento, caracterizando o conceito de melhoria contínua.

A fim de compreender melhor o conceito de monitoramento e controle conforme a disciplina gerencial BPM, foco de interesse do trabalho, é apresentando o tópico a seguir, 3.2. Neste são definidos os conceitos e características intrínsecas ao tema, abordando formatos de implementação e seu impacto no refinamento para melhoria do desempenho dos processos e, por consequência, ao negócio como um todo.

3.2 MONITORAMENTO

Segundo o Guia CBOK, equiparando com a fase verificar (*Check*) do ciclo de vida PDCA, é atribuído ao monitoramento o objetivo de medir o desempenho real do

processo em comparação ao desempenho esperado (ABPMP, 2013). Para isso, monitorar se caracteriza por uma maneira de analisar, coletar e oferecer informações, as quais são relativas à estruturação dos dados das atividades de um processo.

3.2.1 Importância do monitoramento para os modelos BPMM

Modelos de Maturidade de processo de negócio, designados de BPMM (*Business Process Maturity Model*), possibilitam à organização avaliar a maturidade das práticas relacionadas à gestão por processos. Para isso, são definidos parâmetros que possibilitam identificar o quão avançadas as organizações estão na implementação de BPM.

Atualmente existe uma imensa quantidade de modelos BPMM. Segundo a ABPMP (*Association of Business Process Management Professional*) já foram identificados 150 modelos propostos que abordam diferentes pontos de vista (ABPMP, 2013). Uma deficiência desses modelos tem sido o foco simplificador de apenas uma dimensão para medir a maturidade do BPM e a falta de aplicações reais (ROSEMANN *et al.*, 2004). Dessa forma, avaliar a maturidade e orientar as organizações para um cenário de processos mais estruturados e organizados se torna um desafio, porém, extremamente necessário.

Vários fatores são cruciais para o sucesso do BPM que podem complicar ou impedir sua implementação. Fatores críticos de sucesso comumente mencionados são: organização e mudança cultural, alinhamento da abordagem BPM às metas e estratégias corporativas, foco no cliente e seus requisitos, medição e melhoria de processos, a necessidade de uma abordagem estruturada ao BPM, compromisso de gerenciamento (ROSEMANN & BRUIN, 2005).

A complexidade de aplicação das práticas BPM, contento todos os fatores mencionados, resulta na dificuldade de implementação e progressão para estágios mais altos de maturidade. Por isso, vem se discutindo diferentes formatos e definições para modelos que encontrem o equilíbrio entre a complexidade e simplificação (ROSEMANN & BRUIN, 2005). Nesse contexto, surge a importância do monitoramento, definido por alguns autores como fator crucial de maturidade.

O modelo BPMM-Lee (LEE *et al.*, 2007) aplicável ao escopo de BPM e processos, define como nível máximo de maturidade as organizações na qual os processos são monitorados pré-ativamente e controlados, utilizando os dados de

desempenho dos processos sistematicamente para melhorá-los. Complementando, o modelo de Escada de Maturidade do Processo (PML) (HARMON, 2004, 2007) define como nível máximo de maturidade quando os processos são medidos e gerenciados, contando com uma equipe de melhoria de processos.

O artigo *“Developing a business process management maturity model: A study of 300 Iranian superior companies”* (SHAFIEI & HAJIHEYDARI, 2014) apresenta uma revisão da literatura em busca de indicadores de maturidades BPM. Para isso, é considerado os principais fatores para o sucesso de sua implementação, resultando em um guia para as organizações.

Como resultado obteve-se uma lista de 10 indicadores identificados como essenciais, dentre eles, o monitoramento (ver Figura 6). Este foi categorizado como uma ferramenta associada à medição de índices e do processo de melhorias. O estudo prosseguiu através de um questionário distribuído entre 300 empresas superiores iranianas para identificar se as características definidas cobrem os conceitos de maturidade utilizados.

Na visão das organizações, em termos de importância, a arquitetura de processos é considerada o indicador mais importante, enquanto que o monitoramento está dentre os menos importantes, na oitava colocação, segundo SHAFIEI & HAJIHEYDARI (2014).

Figura 6 - Peso e classificação de indicadores de maturidade.

Category	process		operational		Tool			organizational		
Category weight	0.323		0.189		0.294			0.194		
Dimensions of the category	Scope of implementation	Process architecture	Stakeholder communication and management	BPM governance	methods	BPM technology	monitoring	culture	people	Strategic alignment
Dimension weight	0.111	0.0889	0.0394	0.606	0.43	0.38	0.19	0.346	0.47	0.184
Dimension weight in its category	0.036	0.287	0.074	0.114	0.126	0.112	0.056	0.067	0.091	0.035
Dimension rank	9	1	6	3	2	4	8	7	5	10

Fonte - SHAFIEI & HAJIHEYDARI (2014).

Essa pesquisa revela que as organizações não identificam dentre as suas práticas de gerenciamento por processos, ações de monitoramento, e não a utilizam como parâmetro essencial de maturidade. É enaltecido o contraste entre os conceitos teóricos apresentados pelos autores na revisão da literatura, dos quais, 19 entre os 27 artigos utilizados citam o monitoramento como um dos principais domínios e indicadores de maturidade.

Dentre as categorias da Figura 6, processo e ferramentas (incluindo monitoramento, tecnologias e métodos) são os mais importantes entre todas as categorias. Isso sugere que as organizações devem dar mais atenção a essas duas categorias tentando melhorá-las (SHAFIEI & HAJIHEYDARI, 2014). De acordo com os resultados desta pesquisa, foi identificado que as políticas organizacionais são o maior obstáculo à movimentação em direção a níveis mais altos de maturidade. Comportamento humano, motivações e estrutura organizacional foram estabelecidas como fatores mais importantes do que obstáculos à tecnologia.

Para ROSSEMAN & BRUIN (2005), o fator que coloca as organizações em níveis baixos de maturidade em práticas BPM, são fatores culturais e de desempenho. Dessa forma, uma das proposições de valor que um modelo deve oferecer, é servir como uma ferramenta de monitoramento contínuo no processo passando do estado em que está para o futuro.

3.2.2 O papel do monitoramento em sistemas BPM

O conjunto de funções associadas a um sistema BPM geralmente são limitadas por: modelagem, execução, controle e monitoramento. A aplicabilidade destes em áreas tão diversas ocasionou na existência de diferentes abordagens na definição de sistemas BPM e principalmente na falta de consenso sobre as suas principais características e funções (NEELY *et al.*, 2007).

A diversidade de definições fornece diferentes perspectivas provenientes de características próprias a cada sistema. Dentre os autores, as principais características são definidas através de sistemas que permitem à empresa planejar, medir e controlar seu desempenho. Dessa forma, deve garantir que as iniciativas de vendas e marketing, práticas operacionais, recursos de tecnologia da informação, decisão de negócios e atividades das pessoas estejam alinhadas com os negócios estratégicos (MAISEL, 2001, p. 12).

O artigo "*Towards a definition of a business performance measurement system*" (NEELY *et al.*, 2007) apresenta uma revisão da literatura buscando os papéis e funções que definem as principais características de um sistema BPM e como estas se relacionam com a medição de desempenho. Através das definições foram identificados os papéis que o sistema BPM desempenha em uma organização. O resultado é demonstrado na Figura 7.

Figura 7 - Principais funções de sistemas BPM.

	Atkinson (1998)	Atkinson <i>et al.</i> (1997)	Bititci <i>et al.</i> (1997)	Bourne <i>et al.</i> (2002)	Forza and Salvador (2000)	Gates (1999)	Ittner <i>et al.</i> (2003)	Kaplan and Norton (1996)	Kerssens-Van Drongelen and Fisscher (2003)	Lebas (1995)	Lynch and Cross (1991)	Maisel (2001)	McGee (1992)	Neely (1998)	Neely <i>et al.</i> (1995)	Otley (1999)	Rogers (1990)	Total percentage
Strategy implementation/execution	X					X	X	X			X	X	X	X		X	X	59
Focus attention/provide alignment	X						X					X	X	X		X	X	41
Internal communication (communicating performance, and priorities/objectives)			X		X				X		X			X	X	X		41
Measure performance/performance evaluation								X	X	X			X	X	X			41
Monitor progress	X	X									X	X	X	X	X			35
Planning				X							X	X	X				X	29
External communication									X				X	X				24
Rewards	X												X			X		18
Performance improvement														X	X			18
Managing relationships	X	X																12
Feedback					X											X		12
Double-loop learning	X									X								12
Strategy formulation														X				6
Benchmarking														X				6
Compliance with regulations														X				6
Control												X						6
Influence behaviour																X		6

Note: "X" is included if the definition provided refers to the role stated in the first column

Fonte - Adaptado de NEELY *et al.*, 2007.

O resultado apontado pelo autor revela o pouco consenso quanto às características, no entanto, denota que 53% dos autores mencionam em suas definições a função de “*performance measures*” (medidas de desempenho). De encontro com a temática proposta, 35% dos autores mencionam “*monitor progress*” (monitoramento de progresso) como uma das funções essenciais.

A função de medição de desempenho é levada em consideração pois o autor elege 5 categorias que englobam as funções mencionadas, sugerindo como ponto de partida para a seleção dos papéis dos sistemas BPM. A categoria relativa à medição de desempenho, considerada a principal pela recorrência de citação, é definida pela abrangência dos papéis de monitoramento de progresso, medição de desempenho e avaliação de desempenho (NEELY *et al.*, 2007). Dessa forma, é atribuído ao monitoramento como meio de efetuar a medição e avaliação de desempenho.

Os resultados da revisão da literatura apresentada ressaltam os objetivos propostos, demonstrando que o monitoramento é uma característica imprescindível para qualquer BPMS. Através de ações de aquisição, agrupamento, classificação, análise, interpretação e disseminação de dados, as organizações podem eficientemente medir seu desempenho, monitorar e compelir progresso, e estabelecer prioridades (NEELY, 1998, p.5-6)

4 MONITORAMENTO DE ATIVIDADES DE NEGÓCIO (BAM)

O conceito de monitoramento de atividades de negócios surgiu em 2005 pela empresa Gartner, Inc. (JOSHI, 2015), descrevendo os processos e tecnologias que melhoram a consciência da situação e permitem a análise de indicadores críticos de desempenho de negócios com base em dados em tempo real (GARTNER, 2005). O BAM centraliza o foco em análise para as atividades em execução, fornecendo um resumo situacional do negócio aos gerentes da organização, garantindo a rápida tomada de decisão.

A utilização do BAM resulta no aumento da rapidez e eficiência das operações de negócio, mantendo o controle do que está acontecendo e tornando os problemas visíveis (OPSERVICES, 2016). Com sua aplicação, o ambiente de monitoramento é enriquecido, proporcionando uma visão mais ampla das atividades. Mediante as variações, instabilidade e ocorrências inesperadas, torna-se necessário a orientação do planejamento de acordo com a situação apresentada. Portanto, o BAM é o elemento crítico para atingir os objetivos alinhados com as estratégias de negócio (KANG & HAN, 2008).

Tendo em vista as inúmeras operações executadas simultaneamente em uma organização, incluindo processos, sistemas e pessoas, é essencial mecanismos de inteligência para monitorar e controlar. Assim, pode-se gerenciar constantemente os requisitos de conformidade, custos e capacidade, visando a excelência operacional, reduzindo ou eliminando atrasos, gargalos e uso ineficiente de mão de obra e materiais (KANG & HAN, 2008). Nestes cenários, o BAM traz visibilidade ao trabalho em andamento e controle sobre as transações operacionais do dia a dia (JUFURU, 2007).

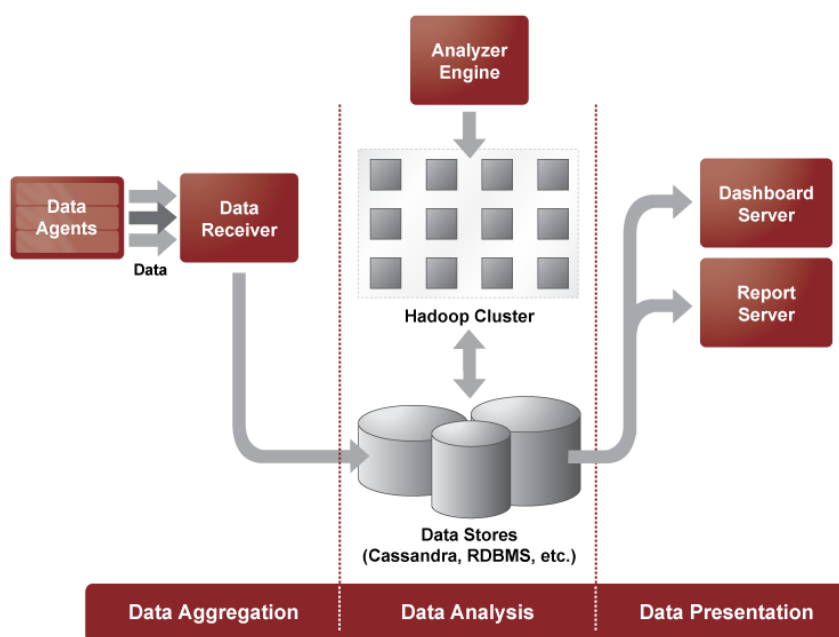
Associado ao requisito de tempo real, as ações oriundas da identificação de oportunidades são imediatas, oferecendo o suporte necessário à tomada de decisão para lidar com a ocorrência de eventos. Através do mapeamento das necessidades e requisitos de suporte aos atores envolvidos, é possibilitado a estes perceber, compreender e responder aos eventos que têm um impacto significativo e seus processos de negócios (DAHANAYAKE *et al.*, 2011).

4.1 SISTEMAS BAM

Os recursos previstos em sistemas BAM devem permitir essencialmente a coleta de dados, processamento e exibição dos resultados para visualização de indicadores de desempenho em tempo real. Empregando mecanismos visando a tomada de decisão, notificações e alertas, os sistemas BAM proporcionam acompanhamento de métricas para identificação da causa raiz de problemas.

Segundo a documentação da empresa WS02, atuando no desenvolvimento de produtos e soluções na área de sistemas e tecnologia, a arquitetura de sistema BAM é composta pelos módulos: agente de dados, receptor de dados, estrutura de análise e por último, apresentação dos resultados (WS02, s.d.).

Figura 8 - Módulos de arquitetura BAM.



Fonte - Site WS02 (s.d).

Após a canalização de dados e identificação de eventos, o acesso aos resultados é disponibilizado em painéis e relatórios, usualmente em formato gráfico. Para os mesmos fins propostos, diferentes arquiteturas são encontradas. Em resumo, a arquitetura apresenta engloba de maneira concisa os módulos principais necessários. Porém, as tecnologias associadas e serviços previstos para cada um dos módulos apresenta uma enorme diversidade.

Os recursos encontrados apresentam diferentes objetivos, dificultando o entendimento sobre estes sistemas. Devido ao fato de não existir padrões formais que enumeram características específicas que os sistemas BAM devem incluir, estes têm sido mal utilizados (DAHANAYAKE *et al.*, 2011). Mediante as vantagens e necessidade para identificar *insights* operacionais necessários para fazer decisões eficazes do negócio, as empresas lutam com a identificação e uso adequado dos sistemas BAM devido a disponibilidade de uma grande variedade de recursos (DAHANAYAKE *et al.*, 2011).

Pela sua dificuldade inerente, os recursos naturalmente incluídos preveem sua integração com sistemas tradicionais de TI, sendo ERPs, CRMs, SCMs e demais softwares, incluindo ferramentas de gerenciamento de processos BPMS. O recurso representativo do BAM é que ele monitora muitos sistemas corporativos simultaneamente e exibe a situação excepcional no painel se os sintomas do problema forem identificados por regras predefinidas (KANG & HAN, 2008).

Relacionado a uma ampla gama de aplicabilidade, as ferramentas associadas são geralmente estruturadas com características específicas ao ambiente empregado. Mediante a diversidade de atividades, seja organizacional, operacional ou comercial, muitos dos sistemas BAM desenvolvidos oferecem soluções em nichos específicos. Segundo a empresa brasileira OpServices (2016), distribuidora do software BAM FiveMetrics, associar processos que permitem integrar diferentes aplicações de diferentes setores é uma tarefa que exige um esforço dos fornecedores BAM. Na verdade, selecionar um sistema BAM adequado é um desafio (DAHANAYAKE *et al.*, 2011).

Para o conjunto de arquiteturas e recursos possíveis para estes sistemas, atualmente é oferecido uma enorme diversidade de softwares, sendo os próprios fornecedores BAM se diferenciando no que oferecem (WEBMETHODS, 2016). Dentre os níveis de detalhamento na disponibilização dos resultados e recursos associados, o BAM pode ir de simples medições, até sistemas sofisticados para aprendizado e previsão de comportamento.

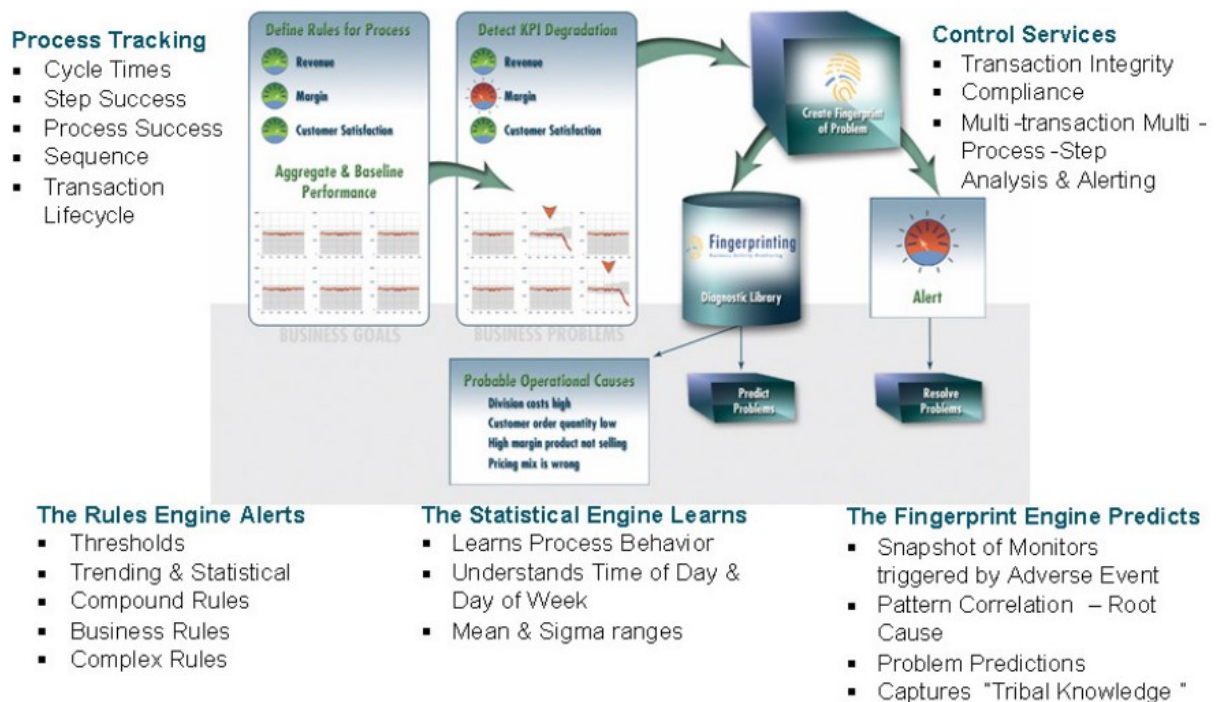
Segundo JUFURU (2007), existem dois tipos de soluções BAM:

- “*Monitoramento de negócios em tempo real com recursos analíticos e painéis.*”

- *BAM mais avançado que é capaz de realizar correlação automática de eventos complexos e análise preditiva por meio de correspondência de padrões de ameaças e oportunidades predefinidas. ”*

O diagrama da Figura 9 resume os tipos de serviços e mecanismos oferecidos por uma solução BAM razoavelmente completa (WEBMETHODS, 2006).

Figura 9 - Serviços e mecanismo de uma solução BAM.



Fonte - WEBMETHODS (2006).

Apresentado uma visão mais sintetizada, BHOWMIK & JOHANEES (2018) descreve em um estudo de caso o monitoramento da atividade de negócios de pagamentos bancários sob a perspectiva de BAM, buscando associar inteligência analítica e estatística para rastreamento e exibição de eventos e informações. Analisando o BAM a partir de uma visão corporativa, foram definidos os seguintes módulos típicos para o sistema:

- **Dados e eventos:** Adaptação com diferentes bancos de dados, sistemas e tecnologias.

- **Correlação e análise de eventos:** Correlação com diferentes eventos, detecção de contextos, indicadores chave de desempenho KPI e análises de situação.
- **Apresentação e notificações:** Alertas de e-mail, gerenciador de alarme, portais de terceiros e painel web.

De forma geral, esses requisitos e serviços mencionados podem ser generalizados a qualquer sistema de monitoramento de atividades. Segundo BHOWMIK & JOHANEES (2018), para oferecer os resultados previstos, um sistema BAM depende de uma abordagem de análise, modelo e adaptação. Para aplicação, o formato de arquitetura e serviços devem ser modelados de acordo com o negócio, criando um ambiente eficaz com as tecnologias mais adequadas para monitoramento dos processos da organização. Como resultado elementar, o sistema BAM precisa monitorar métricas de desempenho em termos de processos de negócios (HAN *et al.*, 2010).

4.2 BENEFÍCIOS DO BAM AO GERENCIAMENTO DE PROCESSOS DE NEGÓCIO BPM

Através da aplicação dos conceitos da disciplina BPM, as atividades de negócio se tornaram mais claras à organização, possibilitando compreender suas relações. Com os modelos de processos apresentados de forma explícita através de sistemas para gerenciamento, informações anteriormente não disponíveis começaram a ser reveladas, impactando diretamente nas análises sobre o negócio.

Com a definição da estrutura de atividades que compõem o processo, o conjunto de tarefas para execução de sistemas, manuais, envio e recebimento, passaram a ter o registro entre os eventos e transações. Neste contexto, ações de verificação de conformidade são possibilitadas, visando a detecção de inconsistência entre um modelo de processo e seu correspondente *log* de execução (ROZINAT & AALST, 2008).

Associado ao contexto de monitoramento previsto pela disciplina BPM, o BAM se tornou um componente indispensável. Ao ter seu foco direcionado a execução das atividades, a intervenção é direta na estrutura operacional do processo de negócio. Além disso, uma vez que o BAM rastreia as execuções do processo e sabe quando

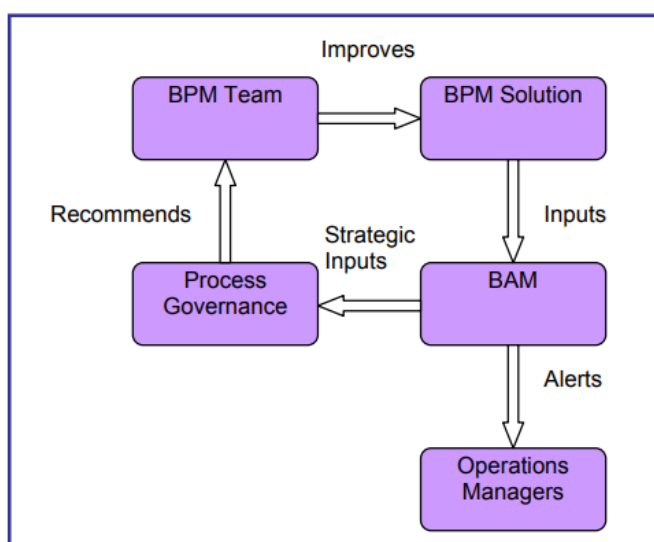
elas têm sucesso ou falham, ele acumula registros valiosos de comportamento que podem levar à melhoria geral do processo (WEBMETHODS, 2016).

Mediante o contexto possibilitado pelo BPM e com o suporte tecnológico do BPMS, se consolidou o cenário ideal para aplicação de BAM. Com a modelagem, automação e gerenciamento de todo o fluxo de informação e ações do processo de negócio, gerou-se os insumos necessários para o monitoramento. Dessa forma, a ferramenta BAM foi vinculada a módulos de BPMS em uma arquitetura mais robusta e abrangente, atuando sobre as informações dos processos automatizados.

“Uma corporação deve definir um processo em torno da solução BAM para obter dados estratégicos do BAM e alimentá-los para uma equipe de governança de processo. A equipe de governança de processos, por sua vez, analisa essas entradas e fornece uma orientação para a equipe de BPM para refinar os processos de negócios e sistemas de TI existentes”. (JUFURU, 2007)

Seguindo a definição de JUFURO (2007), é demonstrado na Figura 10 uma arquitetura de associação entre BAM e BPM.

Figura 10 - Modelo de melhoria contínua segundo a disciplina BPM através de BAM.



Fonte – JUFURU (2007).

A combinação de BPM e BAM se apresenta como uma enorme vantagem competitiva para o sucesso dos negócios. Assim, com o suporte BPM para fornecer

uma visão orientada dos ativos de TI, o BAM fornece o máximo impacto comercial e benefícios (WEBMETHODS, 2016). Ao ampliar a visão relativamente estática para um cenário muito mais dinâmico, as interações e relações são tratadas com maior precisão.

4.3 MEDIDAS DE DESEMPENHO PARA MONITORAMENTO DE ATIVIDADES

Para proporcionar aos gerentes um ambiente eficaz de monitoramento das atividades para análise de desempenho, a definição das métricas e indicadores deve apresentar as medidas que possibilitam o correto entendimento situacional sobre as atividades. A criação de um ambiente BAM não é apenas ter a tecnologia e os processos certos, mas também as empresas devem definir o conjunto certo de métricas (HAN *et al.*, 2010).

Geralmente é referenciado a utilização de painéis BAM para medição através indicadores de desempenho ou KPI (*Key Performance Indicator*), porém seu uso também está fundamentalmente relacionado a métricas. Por se tratar de medidas de atividades em execução, a análise de situação através de uma visão de evolução histórica não se aplica diretamente, pois o foco é nas atividades no presente e não no contexto passado. Porém, as correlações entre estas podem ser expressas através de indicadores, resultando em medidas mais complexas.

Para a atuação correta no refinamento e melhorias das atividades e do processo como um todo, a definição dos indicadores deve proporcionar uma interpretação intuitiva e direta, não sendo propensa a ambiguidades. Relacionado a eventos, os indicadores devem ser correlacionados para identificação de situações fora do normal, condições específicas não atendidas e demais ocasiões indicativas de problemas. Para isso, os recursos gráficos devem ser definidos de modo a facilitar a identificação das situações e o entendimento do resultado da medição.

Para obter o aumento da eficiência e entregar um serviço com o máximo valor agregado, correlacionar os indicadores aos objetivos do negócio e ao gerenciamento dos processos é essencial. As atividades de BPM devem estar intimamente relacionadas às métricas de desempenho, porque as métricas são os impulsionadores e avaliadores de processos de negócios (HAN *et al.*, 2010).

4.4 BI: APOIO NO SUPORTE À DECISÃO BAM

Integrar ao ambiente de monitoramento ferramentas para análises de diferentes contextos, fornece o maior enriquecimento de informações para a tomada de decisão. Ambas as ferramentas têm o mesmo objetivo: melhorar o desempenho do processo. Porém, enquanto o BI fornece uma visão estratégica e tática para estruturação de um planejamento com tendências históricas, o BAM se restringe a visão operacional para ações imediatas.

A confusão entre os sistemas é derivada da semelhança em alto nível dos mecanismos de funcionamento, partindo da necessidade de aquisição de dados para análise e apresentação dos resultados em diferentes formas. Além da semelhança dos benefícios promovidos, principalmente por se restringirem a métodos visando o aprimoramento do processo decisório.

Da mesma forma que a integração de BAM como um *suite* BPMS se viabilizou como uma solução às necessidades do mercado, a utilização do BAM em conjunto com o BI não seria diferente. Ambos os sistemas podem funcionar separadamente, inclusive este é o formato disponibilizado comercialmente, porém a adequação para utilização de ambos os sistemas ao negócio possibilita potencializar os resultados nas análises individuais geradas, tais como:

- O uso dos resultados do BI como parâmetro de referência para a definição de eventos no BAM, estabelecendo as medidas das atividades correspondendo ao valor histórico esperado.
- Reconhecimento das causas das mudanças das tendências históricas do BI, através da identificação de ocorrência indesejadas na execução das atividades BAM. Ao categorizar e identificar os pontos em destaque de áreas de preocupação específicas, pode-se definir possíveis necessidades estratégicas do negócio.

No nível corporativo, o BAM não é visto como um sistema isolado, funcionando com processos através de BPMS ou softwares BI, BRM (*Billing and Revenue Management*), CPM (*Corporate Performance Management*) e demais no contexto EIS (*Executive Information System*), ampliando a abordagem de busca e análise. O

Business Activity Monitoring não se trata de uma evolução do *Business Intelligence* ou de um substituto e sim de uma complementação (DRESDNER, 2002).

5 CONCEPÇÃO DA SOLUÇÃO

Para definição da solução a ser implementada, seguindo os requisitos BAM e visando resolver os problemas levantados, são elaboradas as especificações a serem atendidas ao longo do desenvolvimento. Associado ao contexto do EAPn, a solução deve ser compatível para integração aos sistemas e tecnologias vigentes e adequada ao modelo de dados dos processos automatizados.

Visto que a solução se caracteriza por um sistema, neste capítulo são levantados os requisitos de usuário e sistema, descrevendo as funcionalidades e recursos estabelecidos. As motivações perante o modelo adotado são apresentadas, explanando as considerações e fundamentação acerca da proposta.

5.1 ESPECIFICAÇÃO BAM

Visto a especificidade das aplicações BAM e variedades de serviços englobados, pode-se encontrar soluções em diferentes níveis de complexidade. Conforme descrito no capítulo 3, a divergência entre conceito e sistema dificulta a identificação da solução mais adequada ao cenário do negócio e quais módulos e serviços são necessários, alinhado aos objetivos da organização.

O monitoramento dos processos automatizados é uma iniciativa recente proposta pelo EAPn, disponibilizando atualmente ferramentas de *Business Intelligence*. O estágio atual é de contextualização dos dados dos processos, fornecendo indicadores genéricos para avaliação e análise de desempenho. Seguindo o procedimento gradual de disponibilização de informações aos usuários donos de processo, a solução BAM adotada se enquadra nos mesmos objetivos do BI, propondo-se inicialmente a apresentar uma visão geral das atividades em andamento.

5.1.1 Serviços

Os serviços oferecidos pela solução se restringem ao monitoramento das atividades através de *dashboards*, sendo este o único recurso disponível para representação situacional e análises. No contexto entre BPM e BAM, os painéis gráficos são os elementos centrais, e a partir destes, são associados maiores recursos para geração de relatórios documentais e alertas, não abordados pela solução.

Pelo contexto inicial do monitoramento na universidade, ainda não existem cenários que se adequem a alguns dos serviços descritos no capítulo 3, os quais proporcionam análises com maior detalhamento. Assim, incluir recursos tecnológicos para identificação de tendências, relacionamento entre variáveis para predição, rastreamento de falhas, entre outros, só faz sentido se a organização já possui o mínimo de entendimento sobre seus processos e dados.

Tratando-se dos processos de negócio e do monitoramento efetivo dos donos de processo, tem-se por objetivos principais: possibilitar o reconhecimento de contexto, exibir indicadores de desempenho para avaliação e análises de situação. Assim, tendo à disposição os *dashboards*, pode-se facilmente alcançar os objetivos eficientemente sem necessitar de demais recursos.

5.1.2 Módulos de monitoramento

Os módulos definem a estrutura de monitoramento utilizada para composição do *dashboard*, agrupando os indicadores em seções de análises específicas de acordo com as características de seus resultados. Cada módulo segue um padrão de detalhamento proveniente da própria estrutura de dados do BPMS Bonita, sendo estes:

- **Casos de Processo:** Situação atual dos casos de processo em andamento.
- **Tarefas de Processo:** Situação atual das tarefas humanas em andamento.
- **Usuários:** Situação atual dos usuários atribuídos a tarefas humanas em andamento.

Através dessa estrutura de agrupamento, contendo seus indicadores específicos, o processo de análise é centralizado somente no módulo de interesse ou na correlação entre eles, conforme necessário. Dessa forma, cada módulo apresenta visualmente os indicadores mais importantes em seu contexto, unificando e ordenando em uma única tela para que as informações possam ser monitoradas em um piscar de olhos (NAFIE & ELTAHIR, 2016).

A concepção dos módulos teve como referência o software Bizagi Automation da empresa Bizagi. No EAPn é utilizado um de seus produtos, o software Bizagi Modeler, para modelagem e documentação dos processos automatizados. Um dos recursos do Bizagi Automation é a geração de relatórios BAM para visualização

gráfica do status dos casos de processo em andamento, utilizando uma estrutura similar de módulos, no caso do Bizagi, denominado de relatórios, sendo estes: processo, tarefa e recursos. Adequações foram realizadas de acordo com os objetivos do EAPn e tendo em consideração a estrutura da base de dados utilizada.

Utilizando o software Bizagi, NAFIE & ELTAHIR (2016) aplicaram o monitoramento BAM em um estudo de caso, podendo ser visualizado no artigo os resultados da utilização da estrutura de análise BAM. Dessa forma, é evidenciado a adequação do modelo proposto, fortalecendo a decisão de utilização dessa estrutura de monitoramento.

5.1.2.1 Indicadores de desempenho

Para cada um dos conjuntos de módulos definidos anteriormente, são associados indicadores de desempenho específicos e inerentes à sua respectiva descrição. Através destes, o dono de processo pode quantificar as variáveis do objeto em análise, possibilitando correlacionar o desempenho verificado aos objetivos da organização, a fim de garantir que estão indo na direção certa (BHATTI *et al.*, 2014).

Partindo dos muitos conceitos para a medição de desempenho, NEVES (2012) reúne as definições encontradas em seus diferentes contextos, resumindo os resultados no objetivo central de promover decisões baseadas em fatos. Associado ao BPM, com o objetivo de potencializar a melhoria contínua, os indicadores apresentam-se como um meio de excelência ao dispor da gestão para a monitorização dessa mesma melhoria (NEVES, 2012).

“A captura de eventos de processos de negócios suficientes (por exemplo, início de tarefa, término de tarefa) e seu processamento oportuno nos permite gerar métricas de processo estatisticamente válidas e responder de maneira crítica em relação ao tempo. Contanto que tenhamos um meio de capturar e correlacionar eventos de processos de negócios, podemos produzir métricas e visualizações do desempenho do processo atual.” (BAQUERO *et al.*, 2016).

Para correlação dos indicadores com os eventos específicos das instâncias de processos, são definidos 3 status indicativos das operações do negócio para

monitoramento: no prazo, em risco e atrasado(a). Através desses parâmetros são categorizados os resultados do painel, neste caso, apresentando o tempo como referência e meta.

A relação dos status é dada através da correlação dos resultados do BI, no qual é obtido o tempo médio atual para realização das tarefas e casos de processo, estabelecendo os prazos a serem seguidos para execução. O tempo médio é estabelecido como limiar para indicação do atraso da atividade, o intervalo para estabelecer a região de risco é dado por: 85% a 100% do tempo médio para os casos de processo e 75% a 100% do tempo médio para as tarefas de processo. A diferença dos limites da região é resultado da ordem de grandeza de execução, enquanto os casos são na ordem de dias e meses, as tarefas são na ordem de horas ou no máximo dias, sendo mais voláteis ao atraso. O intervalo que estabelece o cumprimento do prazo é inferior aos respectivos intervalos definidos e superior para o intervalo de atraso.

Para cada um dos módulos foram definidos os seguintes indicadores e especificações:

1. Casos de processos
 - a. Quantidade total de casos no prazo, em risco e atrasados.
 - b. Quantidade de casos no prazo, em risco e atrasados por processo.
2. Tarefas de Processo
 - a. Quantidade total de tarefas humanas no prazo, em risco e atrasadas
 - b. Quantidade de tarefas humanas no prazo, em risco e atrasadas por processo.
 - c. Quantidade de tarefas humanas no prazo, em risco e atrasadas por tarefa.
3. Usuários
 - a. Quantidade de tarefas humanas no prazo, em risco e atrasados por ator de processo.
 - b. Quantidade de usuários com tarefas humanas no prazo, em risco e atrasadas.

O projeto conceitual do painel envolve uma estrutura de representação visual dos resultados através de elementos gráficos mais comuns, sendo a solução

caracterizada por gráficos de barras, rosca (*donuts*) e tabelas (apresentado o detalhamento dos resultados). Além dos indicadores específicos apresentados acima, métricas gerais representadas por numerais brutos são definidas para contextualizar o usuário, apresentando um resumo quantitativo geral das atividades consideradas, sendo estas:

- Quantidade total de casos de processos ativos;
- Quantidade total de tarefas humanas ativas;
- Quantidade total de tarefas humanas ativas atribuídas;
- Quantidade total de usuários diferentes atribuídos a tarefas humanas ativas.

A atualização dos resultados dos indicadores é efetuada pela própria atualização da página, dessa forma não possuem características dinâmicas para atualização em tempo real ou através da identificação de entrada de um novo evento. Para diminuir o período de latência são necessárias soluções mais complexas, sendo especialmente desafiador no contexto BAM, tornando difícil lidar com o volume de dados (BAQUERO *et al.*, 2016).

Por não se tratar de processos críticos, de eventos com baixa taxa de atualização de estados e uma cadeia de fluxo de processos de negócios não distribuída, o requisito de atualização em tempo real não se apresenta vantajoso em contraste à complexidade associada ao desenvolvimento. Neste contexto, a seleção da tecnologia subjacente é crítica para abordar os requisitos de tempo real da solução BAM (BAQUERO *et al.*, 2016).

5.2 ESPECIFICAÇÃO DO SISTEMA

O sistema desenvolvido é caracterizado por uma página web para integração ao Portal de Processos da UFSC, sendo constituída através dos próprios recursos disponibilizados pelo Bonita para criação de páginas de aplicação (ver capítulo 6). Para representação do sistema é utilizado a modelagem UML contendo as especificações de operação e interações entre os componentes do sistema e usuário.

5.2.1 Requisitos de Sistema

Tratando-se de uma iniciativa do EAPn, provendo serviços para qualificar a gestão dos processos automatizados, não foi realizado um levantamento de requisitos

junto ao usuário final. A solução abordada segue requisitos intrínsecos a soluções BAM, fornecendo funcionalidades e propriedades encontradas e atendidas em sistemas do gênero. Assim, o levantamento de requisitos para entendimento da solução a ser concebida foi definido internamente pelo EAPn, visando beneficiar e facilitar o trabalho do usuário final.

Por não existir nenhuma implementação para monitoramento de atividades, o sistema web se conceitua como uma implementação preliminar. Assim, a partir deste, gerando o mínimo entendimento sobre as atividades de negócio, a solução pode ser adaptada e acrescida de demais funcionalidades de acordo com necessidades percebidas pelos donos de processo.

A especificação dos requisitos BAM que essencialmente deverão ser abordados pela solução são demonstrados na Tabela 1. Estes são apresentados através da classificação de requisitos funcionais e não funcionais, conforme as características estabelecidas pela Norma ISO/IEC 9126: funcionalidade, usabilidade, confiabilidade, eficiência, manutenibilidade e portabilidade.

Tabela 1 – Requisitos funcionais e não funcionais.

Tipo	Requisito
Funcional	O sistema deve emitir um resumo das atividades em execução dos processos automatizados
Funcional	O sistema deve possibilitar a visualização gráfica de indicadores de desempenho e métricas
Funcional	O sistema deverá ser capaz de detalhar e exibir as informações consideradas para construção dos gráficos
Funcional	O sistema deve gerar os status de casos de processos e tarefas a partir das informações do Metabase
Funcional	O sistema deve conter recursos que facilitem a visualização dos elementos gráficos
Funcional	O sistema deve emitir informações indicando condições adversas
Não-Funcional	O sistema deve ser desenvolvido com linguagens específicas e compatíveis com o UI Designer
Não-Funcional	O sistema deve permitir a adição de novas funcionalidades
Não-Funcional	A interface do sistema deve se adequar independentemente da quantidade de dados a serem exibidos

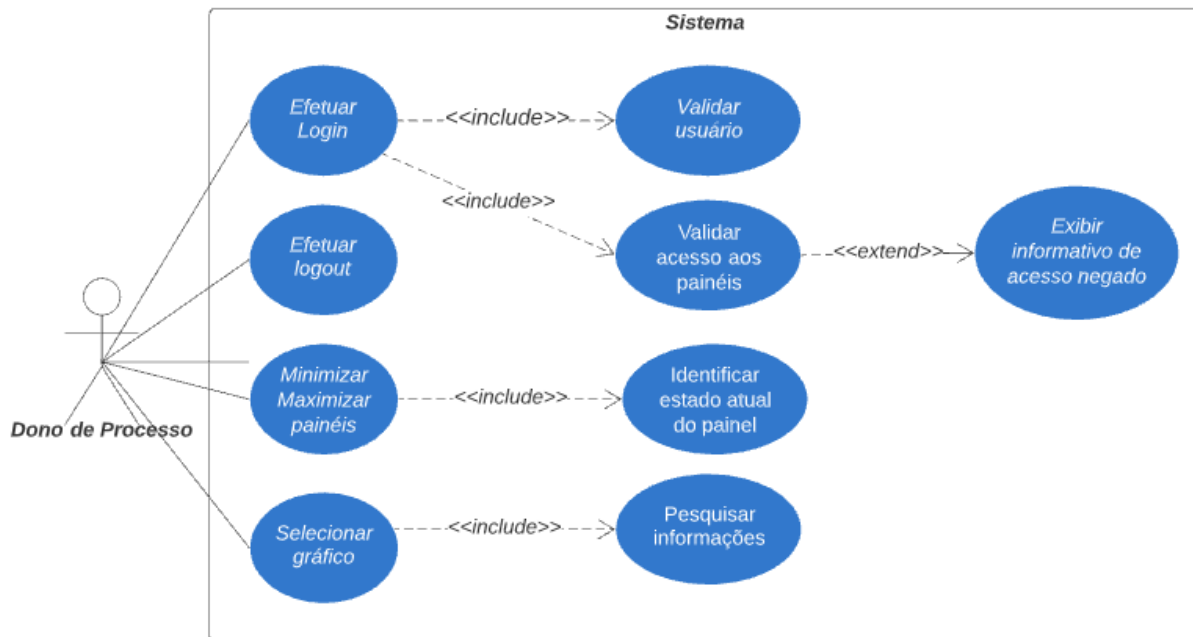
Fonte - Autoria própria.

5.2.2 Modelagem do sistema

Através de diagramas UML é especificado as relações comportamentais de interação entre usuário e página web. A UML inclui nove diagramas inter-relacionados que são usados para modelar diferentes aspectos do sistema (SONG, 2001), neste caso, será empregado o diagrama de casos de uso, descrevendo cada interação entre sistema e ator especificando seus objetivos.

O diagrama de casos de uso representa funcionalidades completas para o usuário e não, funcionalidades internas do sistema, definindo a coleção de modos pelos quais o sistema pode ser utilizado pelos atores envolvidos (TACLA, *s.d.*). O conjunto de casos de uso, atores e relacionamentos podem ser visualizados na Figura 11.

Figura 11 - Diagrama de casos de uso para o sistema proposto.



Fonte - Autoria Própria.

O diagrama apresentado resume os aspectos que constituem a solução implementada, orientando o desenvolvimento mediante as especificações estabelecidas. Para tornar mais fácil a compreensão de como o sistema deve funcionar, uma descrição para cada caso de uso é apresentada abaixo:

- **Efetuar login:** Acessar Portal de Processos com as credenciais UFSC.
 - **Validar usuário:** Confirmar credenciais de acesso.
 - **Validar acesso aos painéis:** Confirmar que o usuário autenticado é dono de processo. Caso contrário, uma mensagem informativa é exibida informando que nenhum painel pode ser exibido.
- **Efetuar logout:** Pressionar botão para sair do sistema do Portal de Processos.
- **Minimizar e maximizar painéis:** Efetuar a ação de minimizar ou maximizar a janela contendo o painel de acordo com o seu estado atual
 - **Identificar estado atual do painel:** Identificar estado atual da janela indicando a ação que pode ser efetuada sobre ela.
- **Selecionar gráfico:** Selecionar um dos elementos gráficos para detalhamento das informações, podendo ser uma coluna do gráfico de barras ou uma seção do gráfico de rosca.

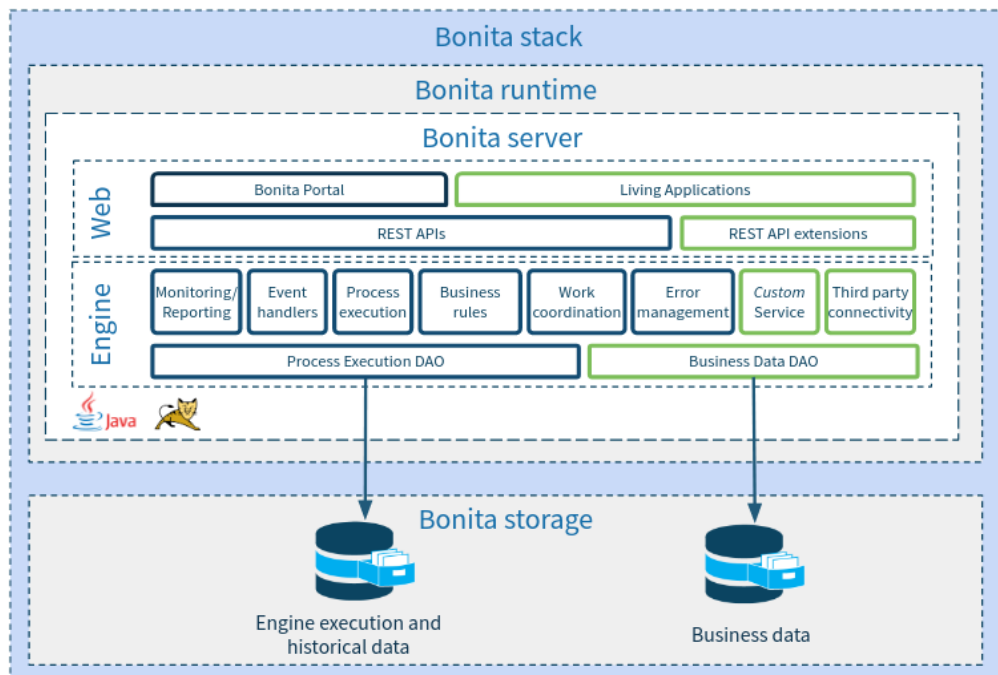
- **Pesquisar informações:** Pesquisar informações sobre o elemento gráfico selecionado para exibição no formato de tabela.

6 SOFTWARE BPMS BONITA: RECURSOS PARA DESENVOLVIMENTO

Para desenvolvimento da aplicação conforme os requisitos definidos no capítulo anterior, será empregado tecnologias disponíveis pelo próprio software de automatização, o Bonita. Caracterizado por uma plataforma DPA, o Bonita oferece suporte para o desenvolvimento de aplicativos de baixo código, promovendo suporte às necessidades do desenvolvedor profissional com recursos como um ambiente de desenvolvimento dedicado para o *design* de IU (KUMAR, 2013).

Os componentes do “Bonita stack”, conforme a Figura 12, define a pilha de componentes para implantar e entregar aplicativos aos usuários finais (BONITASOFT, *s.d.*). Em tempo de execução, é disponibilizado um servidor para gerenciar a execução dos processos (Engine) e o portal web com as interfaces e aplicações para usuários finais e administradores (Web).

Figura 12 - Visão geral da arquitetura de componentes do software Bonita.



Fonte – Bonitasoft (2021).

Os recursos utilizados se restringem principalmente aos mecanismos dos componentes web, determinado as propriedades da aplicação. As ferramentas e aspectos do Bonita são relativas à utilização da edição *community* versão 7.6.3. Uma

visão geral de cada recurso associado ao desenvolvimento da solução são apresentados nos subtópicos abaixo, especificando as particularidades de interesse.

6.1 ESTRUTURA DE DADOS

O Bonita possui uma aplicação de gerenciamento que utiliza 2 bancos de dados relacionais RDBMS (*Relational DataBase Management System*) para armazenar dados do Bonita Engine e do Modelo de Dados de Negócio. Para cada modelo de dados e suas respectivas tabelas são atribuídos os seguintes propósitos:

- O modelo de dados do Bonita Engine armazena informações sobre as definições, configurações e histórico de execução dos processos implementados, usuários e informações de configurações da plataforma Bonita;
- O modelo de dados de negócio ou BDM (*Business Data Model*) armazena os dados corporativos compartilhados por processos e aplicativos. É definido como um conjunto de objetos de negócio, tornando-se variáveis da instância de um processo.

Para conectar-se com o banco de dados, o Bonita oferece os drivers JDBC para conectividade com os sistemas de gerenciamento H2, PostgreSQL e MySQL, executadas dentro de um servidor Apache Tomcat. Para as etapas de desenvolvimento e testes é utilizado o sistema H2, vinculado automaticamente pelo Bonita em suas configurações padrões de instalação. Para as aplicações em servidores de produção é utilizado o MySQL.

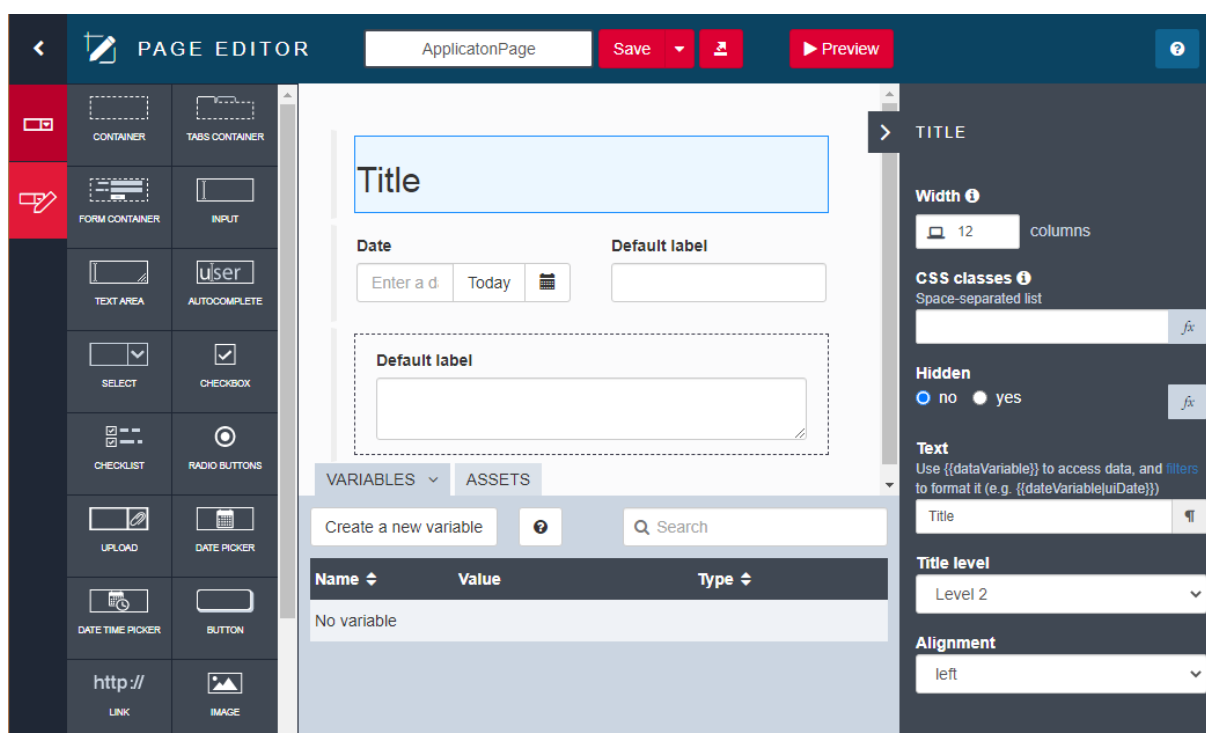
6.2 PÁGINAS E APLICATIVOS

O UI designer permite que os desenvolvedores projetem páginas com interfaces de usuário responsivas, comumente chamadas de páginas de aplicação ou páginas personalizadas (ver Figura 13). Essas denominações são associadas à sua finalidade, referente ao uso em aplicativos ou à customização do Bonita Portal. Este recurso destina-se à programação *front-end* da aplicação, possibilitando configurar a interface gráfica através de elementos visuais e estabelecer as operações resultantes da interação com o usuário.

O conteúdo da página é definido com a utilização de elementos de interação (*widgets*) disponibilizados para uso através da ação pegar e soltar, tais como: botões, textos, tabelas, links e imagens. Os *widgets* são compostos de propriedades que são vinculados a variáveis de dados que definem suas operações e características gráficas, podendo ser configurado de acordo com as necessidades do usuário. As variáveis utilizadas podem ser do tipo string, JSON, API externa, expressão Javascript ou um parâmetro URL, possibilitando a utilização de diferentes estruturas de dados e a implementação de funcionalidades complexas.

Caso seja necessário utilizar elementos diferentes dos disponibilizados, o desenvolvedor pode criar um *widget* personalizados definindo sua aparência e comportamento através das linguagens HTML, CSS e JavaScript, além de *frameworks* associados. Após a criação, os *widgets* personalizados ficam disponíveis junto aos demais para utilização em qualquer página.

Figura 13 - Ambiente de desenvolvimento do UI Designer.



Fonte - Autoria própria.

As páginas web são importadas no Bonita Portal para que sejam incorporadas a um aplicativo (em algumas fontes o recurso é denominado aplicação), permitindo o

acesso para navegação através de um endereço URL. As páginas são um recurso independente com definição própria, não havendo a obrigatoriedade de estar agregada a um aplicativo e inclusive, podendo estar associada a mais de um aplicativo caso necessário.

Para permitir e facilitar a navegabilidade entre diferentes páginas, é definido um aplicativo, conforme estabelecido na Figura 12 por *living application*. Através deste, é disponibilizado um ambiente customizado para um perfil de usuário específico, visando uma necessidade comercial e possibilitando a interação com dados e processos de negócios de maneira mais eficiente que o Bonita Portal padrão. Dessa forma, o aplicativo é responsável por definir os requisitos de navegação entre as suas páginas associadas.

O construtor de aplicativos é uma interface onde é especificado o conjunto de páginas para criar: o mapeamento entre os recursos, a estrutura de menu a ser empregada, o contexto e navegação atribuída a cada página e informações gerais de descrição. A estrutura de navegação destina-se a especificar para quais páginas as seções dos menus devem apontar e a hierarquia a ser utilizada.

6.3 EXTENSÃO DE API REST

API REST é um modelo de arquitetura de comunicação que permite acessar dados com solicitações HTTP, seguindo um conjunto de restrições a serem usadas para fornecer APIs de acesso a um serviço web.

Através das APIs é possível acessar todos os objetos do Bonita, como processos, tarefas e usuários, por meio de métodos de requisição que efetuem as ações de criar, recuperar, atualizar ou excluir. Utilizando as requisições e invocando os métodos específicos a cada API, pode-se obter informações para definir fluxos de trabalho, iniciar um processo, listar informações para uso em páginas e formulários ou até mesmo para recuperar resultados em aplicativos externos.

O Bonita disponibiliza uma biblioteca de APIs com categorias específicas a cada recurso, com chamadas e respostas estruturadas definidas em seu mecanismo, especificadas por um conjunto de URL de solicitações. O conteúdo de retorno é configurado através de parâmetros e filtros de consulta e para cada objeto contido na resposta são associados todos os seus atributos.

Assim, os dados podem ser recuperados via requisição de sistemas externos mediante autenticação e para utilização nas próprias páginas de aplicação. A requisição de dados do Bonita Engine por meio das páginas requer a integração através da API padrão, EJB (*enterprise JavaBeans*) ou baseada em REST. Geralmente associado a mecanismos em um servidor separado, a integração baseada em REST se torna o caminho a seguir (KOPLOWITZ, 2019).

Utilizar esses recursos com respostas previamente estruturadas torna-se um problema em aplicações mais complexas, quando se faz necessário o uso de informações correlacionadas que necessitam de diferentes solicitações de APIs e quando não se utilizam todas as informações contidas no conteúdo da resposta. Dessa forma, para se obter as informações desejadas pode ser necessário desenvolvimentos em *front-end* (formulários, páginas e interfaces visíveis aos usuários).

Para possibilitar ao desenvolvedor a utilização de APIs com um conteúdo de resposta mais adequada à aplicação, é utilizada a ferramenta de criação de Extensão de API REST. Esse recurso é usado para consultar os dados de mecanismo ou de negócios do Bonita criando uma nova estrutura de requisição e resposta com todas as operações em *back-end*. Os arquivos são programados através da linguagem Groovy e são inseridos para utilização no Bonita Portal.

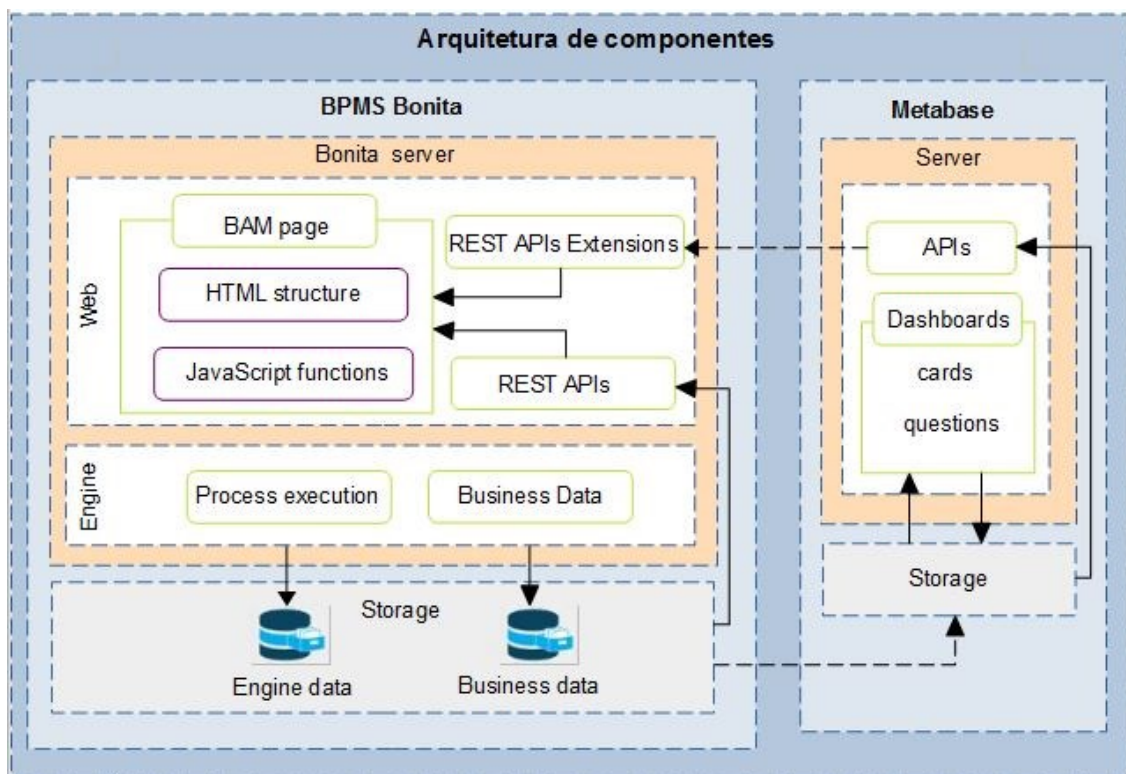
Suas funcionalidades não se restringem à utilização dos dados provenientes do Bonita, mas também de dados de sistemas de terceiros (banco de dados externos, serviços da web, etc.). O processo de integração e utilização de informações do Metabase é proveniente de solicitações por APIs realizadas exclusivamente através dessa ferramenta, visto a necessidade de segurança no processo de autenticação para acesso ao Metabase.

7 DESENVOLVIMENTO

Neste capítulo são apresentadas as particularidades para implementação utilizadas na elaboração do projeto. Serão abordados os procedimentos práticos empregados nas etapas de desenvolvimento, explicitando os métodos aplicados e especificações de estruturação do sistema, de acordo com os objetivos e requisitos definidos.

Utilizando os recursos para desenvolvimento descritos no capítulo 6, conforme a estrutura de componentes do Bonita da Figura 12, é estabelecida a arquitetura de comunicação e o fluxo de informação entre elementos empregados no desenvolvimento da página (ver Figura 14). Assim, é concebida uma visão geral do modelo de componentes, contemplando sua organização e associações.

Figura 14 – Arquitetura de comunicação entre os componentes do Bonita e Metabase.



Fonte - Autoria própria.

Para o desenvolvimento foi aplicado uma metodologia incremental ao longo de todo o ciclo do projeto, conforme previsto nas metodologias de engenharia de software. Através da definição do conjunto de atividades necessárias foi estabelecida as etapas a serem executadas, sendo efetuadas de modo linear e gerando subconjuntos de resultados de forma completa e funcional.

Para cada ciclo em que novas funcionalidades são incluídas, o desenvolvimento é revisado de acordo com as especificações iniciais. Dessa forma, com a utilização da metodologia é possível sistematizar as práticas empregadas, garantindo o suporte às exigências de qualidade do sistema e que os objetivos definidos sejam atendidos.

O modelo incremental segue características do modelo em cascata e iterativo, onde para cada etapa é previsto o planejamento, modelagem, construção e implantação e verificação do incremento a ser desenvolvido. Similar ao modelo iterativo, ao final de cada etapa são realizados os refinamentos, porém sem englobar novos requisitos de sistemas e apenas atuando na melhoria e correções até sua validação.

Voltando à aplicação no desenvolvimento de sistemas, aplicar o procedimento metodológico é fundamental ao atribuir maior visibilidade aos processos, auxiliando na tomada de decisão em situações imprevistas, conforme o planejamento e estimativas definidas. Seguindo o método proposto, é facilitado o acompanhamento da evolução do projeto, justamente por ser composto de funcionalidades que podem ser vistas de modo isolado e serem agregadas conforme a sua evolução. Dessa forma, neste capítulo é apresentada as etapas seguindo a ordem de desenvolvimento estabelecida.

7.1 REQUISIÇÃO DE DADOS DO METABASE

O Metabase disponibiliza um conjunto de APIs para acessar informações dos recursos disponíveis criados através de uma documentação com uma visão geral das requisições. Utilizando os *endpoints* definidos para cada rotina de API é possível acessar o serviço de interesse através da aplicação Bonita.

Para proteger o acesso e iniciar a comunicação com o Metabase, é preciso um *token* de sessão, garantindo a autenticação necessária para efetuar as solicitações. O *token* gerado é incluído nos cabeçalhos de qualquer solicitação subsequente, tendo

como características: sessão com duração máxima de 14 dias, credenciais são armazenados em *cache* devido a limitação dos *logins* por taxa de segurança e a autenticação deve ser feita através das credenciais de um usuário com status de administrador.

Garantindo o acesso através da autenticação é possível realizar as requisições a aplicação do Metabase. Para os objetivos do trabalho, as informações fornecidas resultarão nos parâmetros para definição das condições: no prazo, em risco, atrasado(a). O tempo médio das tarefas e casos de processo obtidos pelas consultas SQL e exibidos nos *dashboards* BI são solicitados e os valores de retorno são utilizados na criação dos indicadores BAM.

Uma consulta SQL e seu respectivo resultado é definido no Metabase como uma pergunta, esta pode ser independente e não estar associada diretamente a nenhum painel. Porém, conforme a estruturação empregada na criação dos *dashboards* BI, cada pergunta está vinculada a um único painel, e este utilizado para exibir indicadores de desempenho de um único processo de negócio.

A associação entre estes elementos é definida através de uma coleção, sendo que para cada coleção terá um painel e as respectivas perguntas associadas. Assim, para identificar de qual processo de negócio o resultado da pergunta se refere, basta verificar o painel que compõe a coleção que está inserida a pergunta. Essa estrutura de relacionamento é necessária, pois o vínculo com o processo de negócio do Bonita é dado somente através do painel.

Mediante a organização de armazenamento do Metabase, para recuperar as informações desejadas conforme descrito, é necessário utilizar os *endpoints* definidos na Tabela 2.

Tabela 2 – *Endpoints* padrão de acesso aos recursos do Metabase.

Domínio	Descrição	Endpoint
Sessão	Autenticação de usuário para geração de <i>token</i> da sessão	/api/session
Pergunta	Visualização do resultado de consultas(perguntas)	/api/card
Coleção	Conjunto de painéis e perguntas	/api/collection

Para implementar as solicitações que vise o melhor desempenho da aplicação, os *endpoints* definidos na tabela acima podem receber complementos em sua URL padrão, facilitando o desenvolvimento do usuário. Assim, pode ser especificado uma extensão do caminho ou parâmetros de consulta. Na Tabela 3 é definido os demais *endpoints* utilizados para obter os resultados das perguntas.

Tabela 3 – *Endpoints* específicos para acesso aos recursos desenvolvidos.

Endpoint	Descrição do retorno
<code>/api/card/public</code>	Lista de perguntas públicos
<code>/api/collection/:id/items?model=card</code>	Perguntas de uma colocação específica
<code>/api/public/card/:uuid/query</code>	Resultado da consulta de uma pergunta pública

Fonte - Autoria própria.

Através do *endpoint* `/api/card` é possível recuperar todas as informações da pergunta, sendo que a maioria é insignificante, visto que nosso foco é exclusivo na informação do resultado da pergunta. Assim, é empregado a solicitação `/api/public/card/:uuid/query`, determinando uma consulta específica a uma pergunta e retornando o resultado associado com reduzidas informações de definição.

Para utilizar a consulta *query* é necessário habilitar o compartilhamento público do elemento, sendo acessível somente através do identificador público (*uuid*) associado. A relação entre os resultados das solicitações possibilita identificar as perguntas de interesse. A estrutura das requisições, métodos utilizados e tratamento das respostas são realizados como parte da configuração da Extensão de API REST, estruturando o conteúdo de retorno para utilização no Bonita.

7.1.1 Extensão de API REST: Requisição ao Metabase

Para obter as informações do resultado das perguntas e dos respectivos painéis no Bonita, é necessário criar uma extensão de API REST. Através de métodos de requisição HTTP é realizada a ação sobre um dado recurso, neste caso, é

executado a autenticação de usuário no Metabase e a requisição para retorno de dados.

As extensões são estruturadas de acordo com as especificações do Bonita, contendo dois tipos de arquivos:

- Um arquivo desenvolvido em programação Groovy. Este define a lógica a ser implementada, contendo as requisições e operações sobre os dados.
- Um arquivo de propriedades. Este define as informações de chamada da API desenvolvida pelo Bonita, contendo principalmente nome e URL da extensão e o mapeamento entre a URL e o script Groovy desenvolvido.

7.1.1.1 Arquivo para definição da lógica

O código da extensão deve seguir o padrão exigido para que seja interpretado corretamente para implantação no Portal Bonita. Para isso, são utilizadas bibliotecas disponíveis em sua documentação, permitindo configurar um conjunto de funções que possibilitam operações de controle e chamada de dados internos, proporcionando as implementações pelo usuário.

Como a extensão desenvolvida é utilizada para obtenção de dados de uma plataforma externa, torna-se somente necessário declarar bibliotecas com classes operacionais para utilização, além de bibliotecas Java para possibilitar as requisições HTTP e Groovy para formatação e interpretação de dados no formato JSON.

A extensão de API é estruturada para recebimento obrigatório de um único parâmetro em sua URL de chamada, o nome do processo de interesse. Assim, para cada requisição, a extensão resultará em uma resposta relativa a um único processo. A primeira operação (ver Figura 15) necessária é a identificação do parâmetro, para que após seja executada somente solicitações ao Metabase das informações correspondentes ao processo requisitado.

Figura 15 - Função para identificação de parâmetro de solicitação da extensão de API REST.

```
def parseRequestParameters(HttpServletRequest request) {  
  
    def requestParameters = [:]  
  
    def processDisplayName = request.getParameter "processDisplayName"  
    if (processDisplayName == null) {  
        throw new IllegalArgumentException('The parameter processDisplayName is missing')  
    }  
    try {  
        requestParameters.put("processDisplayName", processDisplayName.toString())  
    } catch (NumberFormatException nbe) {  
        throw new IllegalArgumentException('The parameter processDisplayName must be a String value')  
    }  
    return requestParameters  
}
```

Fonte - Autoria própria.

As requisições de API do Metabase são executadas seguindo os métodos Java, efetuando uma instância de conexão para cada solicitação. As propriedades da solicitação HTTP são definidas pelos métodos utilizados e pelo cabeçalho instanciado, que contém os parâmetros necessários para a chamada e a obtenção da resposta desejada. Os cabeçalhos podem incluir credenciais, parâmetros para seleção do tipo de conteúdo e *cookies* de sessão, compondo o corpo da solicitação.

A solicitação inicial é referente a autenticação do usuário, verificando se este possui acesso aos dados internos do Metabase para obtenção das informações. Através do método POST são enviados os dados das credenciais de autenticação, conforme os métodos definidos na Figura 16.

Figura 16 - Requisição HTTP POST para autenticação de usuário.

```
def post = new URL("https://dashboard.processos.ufsc.br/api/session").openConnection();  
def message = '{"username":"username", "password":"password"}'  
post.setRequestMethod("POST")  
post.setDoOutput(true)  
post.setRequestProperty("Content-Type", "application/json")  
post.getOutputStream().write(message.getBytes("UTF-8"));  
  
def postRC = post.getResponseCode();
```

Fonte - Autoria própria.

Ao iniciar a requisição é obtido como retorno uma instância, que representa a conexão com o objeto referido pela URL destino, criada a partir da invocação do método *openConnection*. Para definir o envio de dados ao servidor é necessário informar que a conexão de URL deverá ser utilizada para saída, conforme definido em *setDoOutout*, com a variável boolean *true*. Dessa forma, torna-se necessário, por meio do fluxo de saída da conexão, o envio de um corpo para solicitação de autenticação, definido através do método *getOutputStream*.

Como retorno se obtém o código de *status* da solicitação HTTP, dado por uma resposta do tipo inteiro (int). Ele retornará 200 para autorização confirmada e 401 para autorização negada. Caso ocorram problemas na requisição e nenhum código puder ser discernido, o valor de retorno da resposta é -1, configurando uma resposta HTTP inválida.

Após a validação da autorização de usuário, torna-se possível fazer as requisições das informações ao Metabase através do método GET, que executa essencialmente os mesmos procedimentos definidos na solicitação POST. Conforme o trecho de código da Figura 17, é possível verificar os procedimentos e, principalmente, a definição da propriedade de solicitação através da chave obtida no método POST, indicado pelo método *setRequestProperty*.

Figura 17 - Requisição HTTP GET para retorno do resultado das perguntas SQL.

```
def postRC = post.getResponseCode();
if(postRC.equals(200)) {

    def jsonSlurper = new JsonSlurper()
    def authenticationKeys = jsonSlurper.parseText(post.getInputStream().getText())

    def getCards = new URL("https://dashboard.processos.ufsc.br/api/card/public").openConnection();
    getCards.setRequestMethod("GET");
    getCards.setRequestProperty("Content-Type", "application/json")
    getCards.setRequestProperty("X-Metabase-Session", authenticationKeys.id)
    def getResponseCard = getCards.getResponseCode();

    //to be continued ...
}
```

Fonte: Autoria própria.

Realizando operações sobre o conteúdo de retorno das solicitações é estruturada a resposta da extensão de API, definida por um elemento JSON. A Figura 18 exibe um exemplo do objeto de retorno, e apresenta a estrutura de informações

para uma pergunta relacionada ao tempo médio de execução de um caso de processo.

Figura 18 - Exemplo de conteúdo de resposta da requisição da extensão de API.

```
[
  {
    "data": {
      "cols": [
        {
          "base_type": "type/Decimal",
          "display_name": "tempo médio",
          "field_ref": [
            "field-literal",
            "tempo médio",
            "type/Decimal"
          ],
          "name": "tempo médio",
          "source": "native"
        }
      ],
      "insights": null,
      "results_timezone": "America/Sao_Paulo",
      "rows": [
        [
          10.08867913
        ]
      ]
    },
    "json_query": {},
    "status": "completed"
  }
]
```

Fonte: Autoria própria.

7.1.1.2 Arquivo de propriedades

As informações relacionadas à extensão da API REST asseguram a identificação da extensão nos diferentes módulos do Bonita, constituindo informações para definição, operação e permissões, sendo elas:

- **contentType**: definição do tipo de recurso a ser implementado;
- **name**: nome da extensão da API REST. Obrigatoriamente deve começar com “*custompage_*”;
- **displayName**: nome para exibição no Bonita Portal;

- **description:** incluir opcionalmente uma descrição;
- **apiExtensions:** definir a lista de APIs do Bonita utilizadas na extensão desenvolvida, separadas por vírgula. Opcional, caso nenhuma tenha sido utilizada;
- **method:** definição do método de operação para requisição da extensão implementada;
- **pathTemplate:** definir o caminho de URL para chamada da extensão de API;
- **classFileName:** declarar o arquivo groovy associado;
- **permissions:** declarar as permissões para autorização de acesso aos recursos para usuários.

O arquivo resultante com as devidas propriedades declaradas é demonstrado na Figura 19.

Figura 19 - Arquivo de propriedades da Extensão de API.

```
#The name must start with 'custompage_'  
#Wed May 12 13:54:17 BRT 2021  
name=custompage_cardInformationRestAPIExtension  
cardInformationRestAPIExtension.classFileName=Index.groovy  
cardInformationRestAPIExtension.pathTemplate=cardInformation  
cardInformationRestAPIExtension.method=GET  
displayName=Card information REST API Extension  
contentType=apiExtension  
description=Query Metabase Cards to retrieve user information  
cardInformationRestAPIExtension.permissions=organization_visualization  
apiExtensions=cardInformationRestAPIExtension
```

Fonte - Autoria própria.

Os dois arquivos desenvolvidos devem ser implantados no Bonita Portal, adicionando um novo recurso do tipo extensão de API. Com isso é possível a sua utilização no Bonita UI Designer, executando as requisições conforme as definições estabelecidas no arquivo de propriedades para incorporação dos dados de retorno em páginas e formulários.

7.2 ESTRUTURAÇÃO DA PÁGINA BAM

Através do ambiente de desenvolvimento UI Designer é estabelecido a estrutura da página web com as devidas funcionalidades para exibição dos indicadores para monitoramento BAM. Assim, foram definidas as operações e elementos de interação, conforme os requisitos estabelecidos na concepção da solução.

7.2.1 *Layout da interface*

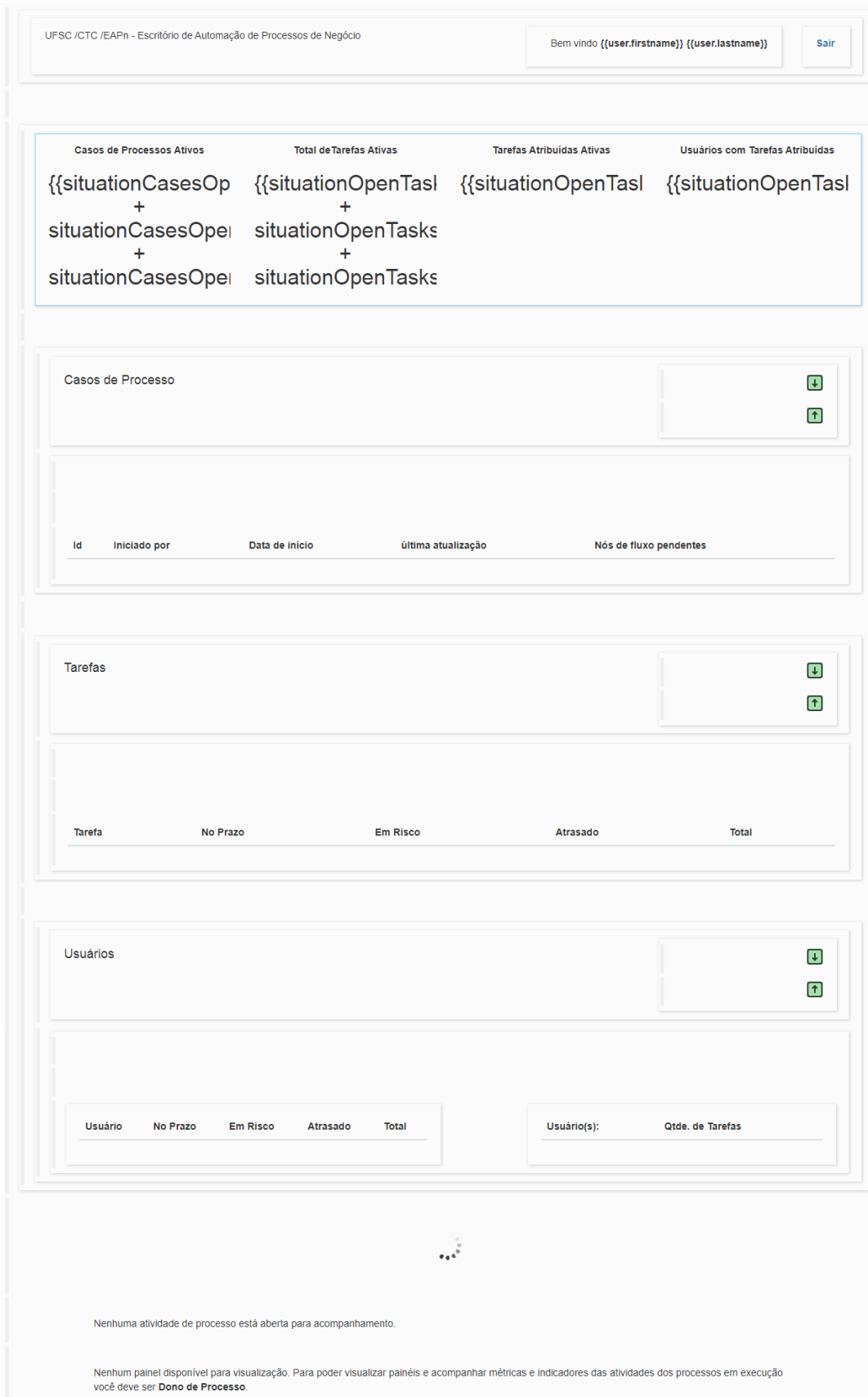
O *layout* da página segue os mesmos requisitos das demais páginas do Portal de Processos, caracterizada por uma interface simples e intuitiva, permitindo a adaptação aos novos conceitos e facilitando a procura de informações por parte dos utilizadores. A estrutura da página é definida por 4 segmentos, agrupando um conjunto específico de elementos através de *containers*, sendo estes:

- I. Cabeçalho contendo a identificação do usuário e operação de *logout*;
- II. Métricas gerais para resumo das informações exibidas graficamente;
- III. Os 3 Módulos de monitoramento contendo os gráficos e tabelas para detalhamento;
- IV. Mensagens de aviso e o elemento de carregamento.

Em aspectos gerais, o posicionamento e disposição dos elementos buscam oferecer uma arquitetura de informação que estabeleça a organização em uma sequência lógica para análise dos indicadores. Assim, conforme a estrutura de execução de um processo, a sequência é definida pela característica de composição: casos, tarefas e usuários. Através da instanciação de um processo é originado um caso que é composto por uma sequência de tarefas, sendo cada uma das tarefas atribuídas a um usuário.

O resultado pode ser visto na Figura 20, o qual exibi os elementos de acordo com a interface do ambiente de desenvolvimento do UI Designer.

Figura 20 - Interface da página definida no UI Designer Bonita.



Fonte: Autoria própria.

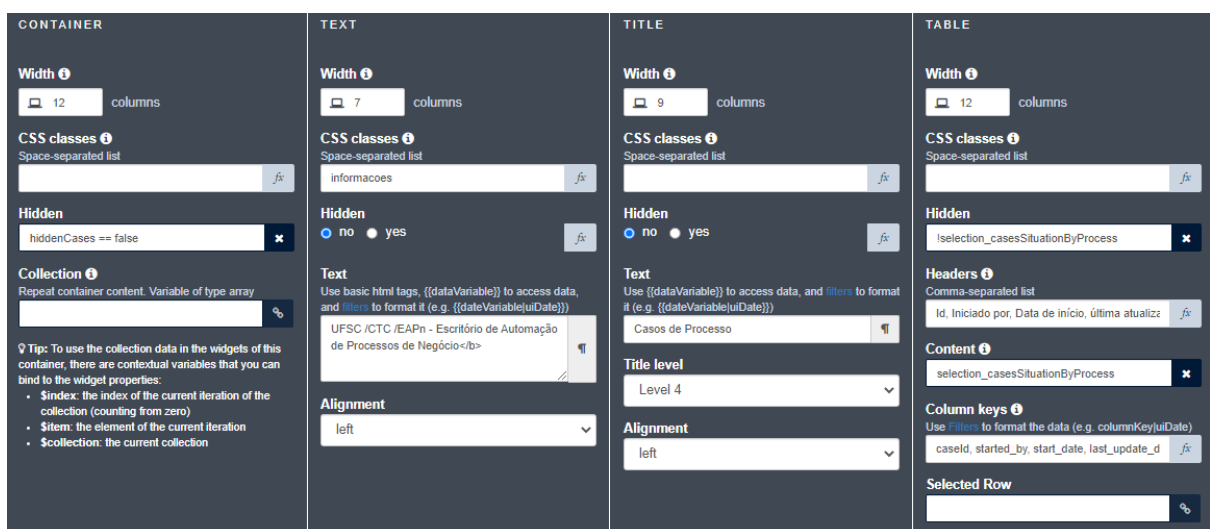
7.2.2 Elementos de interação

A interface é caracterizada por elementos de interação disponíveis no UI Designer (*widgets* padrão) e elementos customizáveis criados pelo desenvolvedor para executar operações específicas não atendidas pelos elementos padrão. Cada *widget* é responsável por um item de informação a ser inserido ou exibido, definindo através de suas propriedades o comportamento da página (BONITASOFT, s.d.).

Tratando inicialmente dos elementos disponibilizados, torna-se necessário apenas configurar propriedades de funcionamento, definindo as variáveis de entrada, saída de dados e aparência. Dentro desta categoria, para obter o resultado da Figura 20 foi empregado os *widgets*: container, texto, título e tabela.

As configurações dos elementos utilizados seguem propriedades gerais que todos os *widgets* possuem e algumas que são específicas para o tipo de *widget*. As propriedades são relacionadas a própria estrutura html do elemento e apenas definidas na interface de desenvolvimento. Cada um dos elementos descritos é aplicado mais de uma vez na página e possuem definição própria. Na Figura 21 é possível verificar um exemplo de configuração para cada tipo de elemento mencionado. As propriedades definidas refletem a utilização em um caso de uso específico.

Figura 21 - Exemplo de definição das propriedades dos elementos container, texto, título e tabela.



Fonte - Autoria própria.

A utilização desses elementos para outros casos de uso segue a mesma lógica apresentada, sendo as variáveis de entrada para definição das propriedades relativas à aplicação. Os demais *widgets* utilizados são elementos customizáveis criados pelo desenvolvedor (autor). A ferramenta disponibiliza o ambiente para criação no UI Designer, permitindo ao usuário configurar:

- **Description:** Descrição para identificar o elemento.
- **Template:** Modelo que descreve a estrutura de marcação HTML do elemento.
- **Controller:** Definir a funcionalidade/comportamento em JavaScript.
- **Assets:** Incluir ativos externos, podendo ser bibliotecas ou frameworks JavaScript, definições de CSS ou imagens.
- **Angular modules:** Incluir módulos AngularJS caso necessite usar serviços ou diretivas que não estão associadas por padrão.
- **Properties:** Definir propriedades dinâmicas para utilizar variáveis que dependem de páginas, formulários ou da definição do usuário, personalizando sua aparência ou comportamento.

Após realizar as configurações, o elemento se torna disponível para utilização em qualquer página. Os subtópicos subsequentes descrevem o desenvolvimento dos *custom widgets* necessários à aplicação.

7.2.2.1 Gráficos

Os *widgets* gráficos compreendem dois elementos customizáveis: gráfico de barras e gráfico de rosca. Para visualização dos dados é empregada a biblioteca externa JavaScript Chart.js em sua versão mais recente, 3.2.0. A biblioteca é definida no *assets* através do link:

- <https://cdnjs.cloudflare.com/ajax/libs/Chart.js/3.2.0/chart.min.js>

Para exibição dos gráficos na página web é necessário empregar o elemento HTML *canvas* (disponível na versão HTML5), sendo a única marcação necessária para estruturação do *widget*. Esse elemento é utilizado exclusivamente para desenho e renderização dinâmica de gráficos via JavaScript, assim toda a lógica de implementação, dados e formato de visualização são definidos pelo *controller*. A programação do elemento é exibida na Figura 22 para o gráfico de barras (o gráfico de rosca apresenta codificação similar).

Figura 22 - Estrutura do elemento customizável do tipo gráfico de barras.

Template

```
<div>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/3.2.0/chart.min.js"></script>
  <canvas id="chartBarCasesByProcesses"></canvas>
</div>
```

Controller

```
new Chart(document.getElementById("chartBarCasesByProcesses"), {
  type: 'bar',
  data: {
    labels: labelsProcessName,
    datasets: [
      {
        label: "No Prazo",
        backgroundColor: 'rgb(0, 246, 0)',
        data: dataNoPrazo
      }, {
        label: "Em Risco",
        backgroundColor: 'rgb(255, 255, 0)',
        data: dataEmRisco
      }, {
        label: "Atrasados",
        backgroundColor: 'rgb(255, 0, 0)',
        data: dataAtrasado
      }
    ]
  },
  options: {
    onClick: function(e) {
      var element = this.getElementAtEvent(e);
      if(element.length > 0) {
        if(element[0]._datasetIndex === 0) {
          $scope.properties.selectionResult = $scope.properties.value[element[0]._index].casosPrazo;
        }
        else if(element[0]._datasetIndex === 1) {
          $scope.properties.selectionResult = $scope.properties.value[element[0]. index].casosRisco;
        }
        else {
          $scope.properties.selectionResult = $scope.properties.value[element[0]._index].casosAtrasados;
        }
      }
      $scope.$apply();
    }
  },
  scales: {
    xAxes: [{
      ticks: {
        display: false
      }
    }],
    yAxes: [{
      ticks: {
        min: 0
      }
    }
  ]
},
  title: {
    display: true,
    text: `Situação dos Casos em andamento por Processo`,
    position: 'top'
  }
}
});
```

Fonte - Autoria própria.

O código JS do *controller* não é exibido por completo, apenas é demonstrado a etapa de criação e configuração do gráfico. O código não exibido é destinado ao tratamento dos dados JSON, recebidos da página para estruturação do conjunto de

arrays para definição dos *datasets* do gráfico, conforme estabelecido pela a biblioteca *chart.js*. Configurações opcionais são definidas para possibilitar uma melhor usabilidade, sendo empregado o ajuste de escala, legenda e título.

Para a estruturação de controle do gráfico da Figura 21 é estabelecido o evento *onclick*, identificando quando o usuário clicar em algum elemento do gráfico. Neste caso, o objetivo dessa ação é identificar qual barra do gráfico foi selecionada e retornar à informação para página, que exibirá uma tabela com informações adicionais.

7.2.2.2 Botão para minimizar e maximizar painel

O UI Designer disponibiliza dentre seus elementos padrões *widgets* do tipo botão, porém as configurações de propriedades disponíveis não permitem inserir uma variável para realização de operações. A variável de interesse indica a situação atual da janela do módulo, identificando qual operação poderá ser executada: minimizar ou maximizar.

Mediante as limitações foi desenvolvido o *custom widget* contendo uma estrutura clássica de botão, porém designando uma propriedade de entrada para recebimento dinâmico de variável. Além da definição do botão, a estrutura de marcação do *template* é constituída pela definição de um ícone construído em *css* e acessível através de uma biblioteca de código aberto:

- <https://css.gg/arrow-down-r.css>

O ícone é caracterizado por uma seta e utilizado como o conteúdo de texto do botão, indicando as respectivas ações de minimização ou maximização da janela passíveis de serem executadas pelo usuário. A programação do elemento é exibida na Figura 23 para o botão de maximizar (o botão de minimizar apresenta código similar).

Figura 23 - Estrutura de elemento botão para operação de minimizar e maximizar janela.

Template

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<!-- Add icon library -->
<link href='https://css.gg/arrow-down-r.css' rel='stylesheet'>
<style>
.btn3{
  background-color: #A5DFB2;
  border: none;
  color: #283723;
  padding: 0px 0px;
  cursor: pointer;
}
</style>
</head>
<body>
<div class="text-{{ properties.alignment }}">
  <button id="myFram" class="btn3"><i class="gg-arrow-down-r"></i></button>
</div>
</body>
</html>
```

Controller

```
function ($scope) {
  document.getElementById('myFram').addEventListener("click", function() {
    $scope.properties.variable = true;
    $scope.$apply();
  });
  $scope.$apply();
}
```

Fonte - Autoria própria.

A única operação de controle necessária é a identificação do clique sobre o botão e atualização de uma variável indicando a ação.

7.2.2.3 Elemento gráfico de carregamento

Definido puramente em HTML e CSS, o elemento gráfico de carregamento animado é utilizado durante a inicialização da página até que todas as operações e resultados correspondentes tenham sido carregados e exibidos. O código pode ser encontrado em bibliotecas de código aberto específicas em *Spinners*, o elemento utilizado é denominado *fading circle*.

7.2.3 Funcionalidades operacionais

Para definir as funcionalidades da página, variáveis e scripts de comando são utilizados para manipular dados corporativos ou para conter informações de operações que determinam como os elementos da página se comportam. Através da conexão entre as variáveis de operação e elementos HTML dinâmicos são definidas as interfaces de entrada e saída de dados e a dependência entre operações.

O conjunto de operações JavaScript desenvolvidas podem ser classificadas perante os seguintes propósitos centrais:

- Identificação de usuário para controle de acesso a página;
- Requisição de informações utilizando APIs do Bonita;
- Requisição de informações da extensão de API para requisição de informações do Metabase;
- Estruturação dos dados em formato JSON contendo as informações de casos, tarefas e usuários para definição dos *datasets* dos gráficos;
- Atualização de variáveis para definição das propriedades dos *widgets*.

Seguindo a estrutura do UI Designer, são definidas funções JavaScript independentes entre si, executadas quando chamadas ou na inicialização da página. Assim, para obter os resultados desejados é necessário o relacionamento entre diferentes operações. Para isso, os procedimentos são executados de modo sequencial, garantindo que ao iniciar qualquer operação está tenha à disposição os dados necessários para processamento.

7.2.3.1 Solicitação HTTP

Através de API REST é realizado solicitações HTTP para acessar dados de objetos do Bonita. Para desenvolvimento da aplicação somente foi necessário estipular métodos para leitura (GET) de recursos, utilizando os resultados em operações ou para exibição das informações na interface da página.

Tabela 4 – *Endpoints* utilizados na página.

Endpoint	Descrição
../bpm/process	Lista de definições de processos
../bpm/processParameter	Lista de parâmetro dos processos
../bpm/case	Lista de casos abertos de processos
../identity/user/	Lista de detalhes do usuário
../system/session/unusedId	Informação da sessão atual
../extension/cardInformation	Recuperar as informações definidas na Extensão de API REST

Fonte - Autoria própria.

Para utilização de chamadas API REST usando solicitações HTTP GET, a resposta é armazenada no modelo de dados da página. Para qualquer sistema que dependa de requisições HTTP e utilize especificações do protocolo 1.1, que não há paralelização de requisições, exige-se boas práticas de programação, demandando principalmente:

- Diminuir ao máximo o número de requisições;
- Diminuir o volume de dados de retorno das requisições, requerendo somente os de interesse para as demais operações.

Essas práticas visam a performance da aplicação, garantindo que futuramente quando o fluxo de dados seja maior que o atual não ocorra perdas demasiadas de desempenho.

7.2.3.2 Funções JavaScript para estruturação dos dados

Utilizando o conteúdo de retorno das solicitações HTTP é realizado o processamento dos dados, estabelecendo uma estrutura de agrupamento utilizada para criação dos gráficos. A série de atividades executadas sequencialmente irá produzir o resultado em formato JSON, visto a facilidade proveniente de ser uma sintaxe baseada em JavaScript.

Para as implementações JavaScript são aplicados elementos construtivos e blocos de código, tais como declarações condicionais, laços, funções e métodos específicos. A página é composta por 3 funções principais, cada uma delas associadas a um dos módulos de monitoramento, realizando o arranjo das informações de acordo com os casos de processo, tarefas e usuários.

Os casos de processo são agrupados de acordo com o seu respectivo processo e a situação do caso, inserindo as principais informações da sua definição (ver Figura 23).

Figura 24 - Estrutura JSON resultante do agrupamento dos casos de processo.

```
[
  {
    "processName": "Nome Processo",
    "casesOnTime": [
      {
        "caseId": 1000,
        "started_by": "Usuário 1",
        "start_date": "2021-05-07 13:37:53.638",
        "last_update_date": "2021-05-07 13:39:52.618",
        "activeFlowNodes": 1
      }
    ],
    "casesAtRisk": [
      {
        "caseId": 1001,
        "started_by": "Usuário 2",
        "start_date": "2021-05-07 14:27:13.528",
        "last_update_date": "2021-07-07 14:27:53.611",
        "activeFlowNodes": 1
      }
    ],
    "casesOverdue": [
      {
        "caseId": 1002,
        "started_by": "Usuário 3",
        "start_date": "2021-03-07 11:31:12.101",
        "last_update_date": "2021-03-07 11:31:19.848",
        "activeFlowNodes": 1
      }
    ]
  }
]
```

Fonte - Autoria própria.

As tarefas são agrupadas de acordo com seu processo de origem, contabilizando para cada tarefa as respectivas situações e inserindo as informações sobre os executores. Assim, nessa mesma estrutura é detalhado e definido as informações dos usuários atribuídos e suas informações de identificação (ver Figura 25).

Figura 25 - Estrutura JSON de resposta do agrupamento das tarefas de processo.

```
[
  {
    "processName": "Nome Processo",
    "taskInfo": [
      {
        "taskName": "Nome Tarefa",
        "actorName": "Nome ator",
        "tasksOnTime": 1,
        "tasksAtRisk": 0,
        "tasksOverdue": 1,
        "tasksAssigned": 1,
        "usersAssigned": [
          {
            "id": 121,
            "name": "Nome Usuário",
            "tasksOnTime": 0,
            "tasksAtRisk": 0,
            "tasksOverdue": 1
          }
        ]
      }
    ]
  }
]
```

Fonte - Autoria própria.

Complementado a estrutura JSON anterior, é desenvolvido um agrupamento específico para usuário, de acordo com a situação de suas respectivas tarefas atribuídas. Um usuário, com o mesmo papel de ator associado ou não, pode estar atribuído como executor de diferentes tarefas em diferentes processos. Assim, é realizado um agrupamento para contabilização da situação de suas tarefas atribuídas, sem considerar a identificação da tarefa ou processo, apenas a situação do usuário dentre todas suas atribuições.

O agrupamento resultante da Figura 26 é uma contagem a partir da estrutura da Figura 25, visando facilitar a geração dos gráficos associados.

Figura 26 - Estrutura JSON de resposta do agrupamento dos usuários para suas tarefas atribuídas.

```
{
  "listUserTaskOnTime": {
    "user": "Nome Usuário",
    "qtd": 1
  },
  "listUserTaskAtRisk": {
    "user": "Nome Usuário",
    "qtd": 4
  },
  "listUserTaskOverdue": {
    "user": "Nome Usuário",
    "qtd": 2
  }
}
```

Fonte - Autoria própria.

As operações mais complexas desenvolvidas são para a geração das estruturas JSON apresentadas. Além destes, apenas códigos mais simples para atualização de variáveis e operações mais básicas para controle e definição das propriedades dos elementos de interação da página foram desenvolvidos.

7.2.3.3 Controle de acessibilidade

A página desenvolvida é de utilidade apenas de usuários donos de processo, mesmo que os dados apresentados não sejam sigilosos, decidiu-se implementar um filtro de usuário para efetuar o controle de acessibilidade à página. Dessa forma, ao acessar a página e ser verificado através de operações definidas que o usuário não é dono de processo, o conteúdo é desabilitado e é exibido um texto informativo indicando o ocorrido.

Essa restrição pode ser removida ou alterada facilmente caso futuramente seja necessário habilitar o acesso a diferentes tipos de usuários.

7.3 INCORPORAÇÃO DA PÁGINA AO PORTAL DE PROCESSOS

A página desenvolvida é inserida no contexto da aplicação do Portal de Processos, sendo integrada como um recurso independente podendo estar associada a um ou mais aplicativos, demandando um mapeamento específico a cada aplicação.

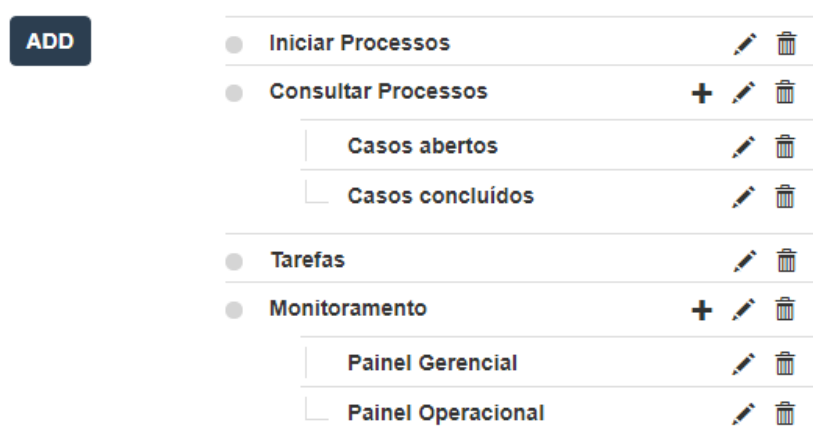
Para criar o vínculo é necessário definir o caminho para acesso ao recurso da página, estabelecendo o contexto para invocá-la através do domínio definido para a aplicação. O acesso criado ao Portal de Processos é referenciado por:

- <https://processos.ufsc.br/bonita/apps/eapnPortal/PainelOperacional/>

A URL é especificada na estrutura de navegação do aplicativo (ver Figura 27), compondo o menu de nível superior do Portal de Processos. Porém, por ser um recurso independente, o acesso à página pode ocorrer de maneira individual, não necessitando estar obrigatoriamente associado ao menu do aplicativo gerado.

Figura 27 - Estrutura de navegação do Portal de Processos da UFSC.

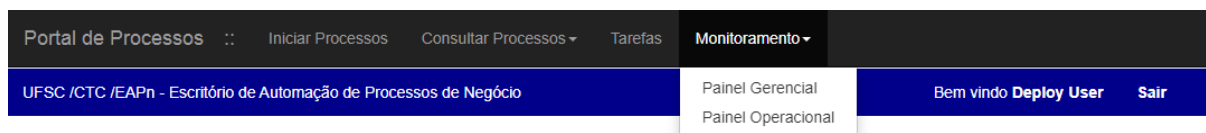
Navigation



Fonte - Autoria própria.

Para a navegação definida é gerada a estrutura do Portal de Processos da Figura 28.

Figura 28 - Menu do Portal de Processos da UFSC.



Fonte: Autoria própria.

8 RESULTADOS

Como resultado foi obtido a página para monitoramento das atividades dos processos automatizados em estágio operacional, pronta para iniciar testes e validação com os usuários. A estrutura apresentada demonstra sua aplicação em um ambiente de produção, integrado ao servidor online do Portal de Processos da UFSC (ver figura 29).

Figura 29 - Página de monitoramento BAM no Portal de Processos da UFSC.

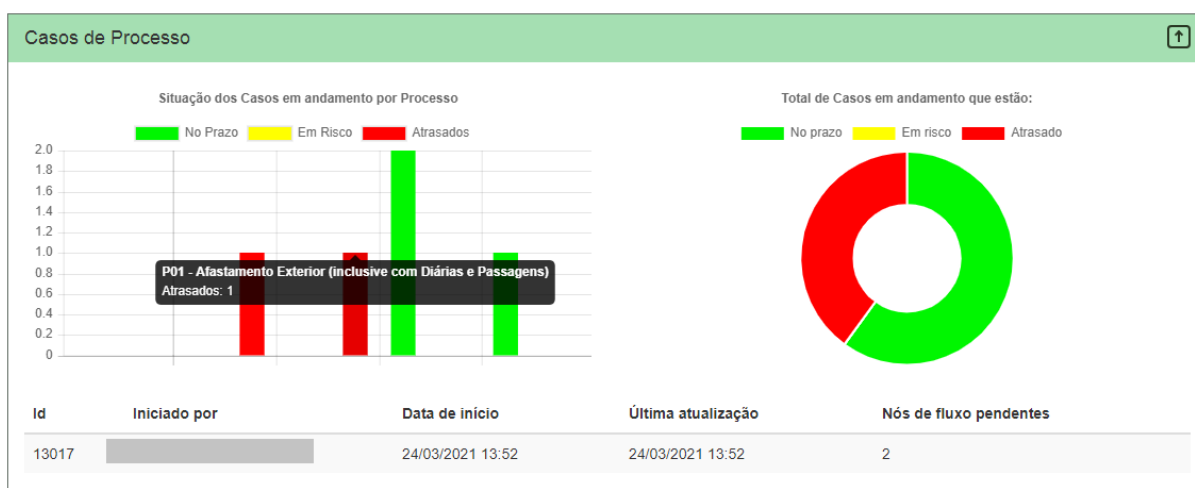


Fonte - Autoria própria.

As informações resultantes são oriundas dos processos automatizados pelo EAPn, apresentadas mediante acesso via usuário de teste (*deploy*), que contém as mesmas permissões de dono de processo. Independente dos processos e da quantidade de tarefas, a página seguirá o mesmo padrão da Figura 28.

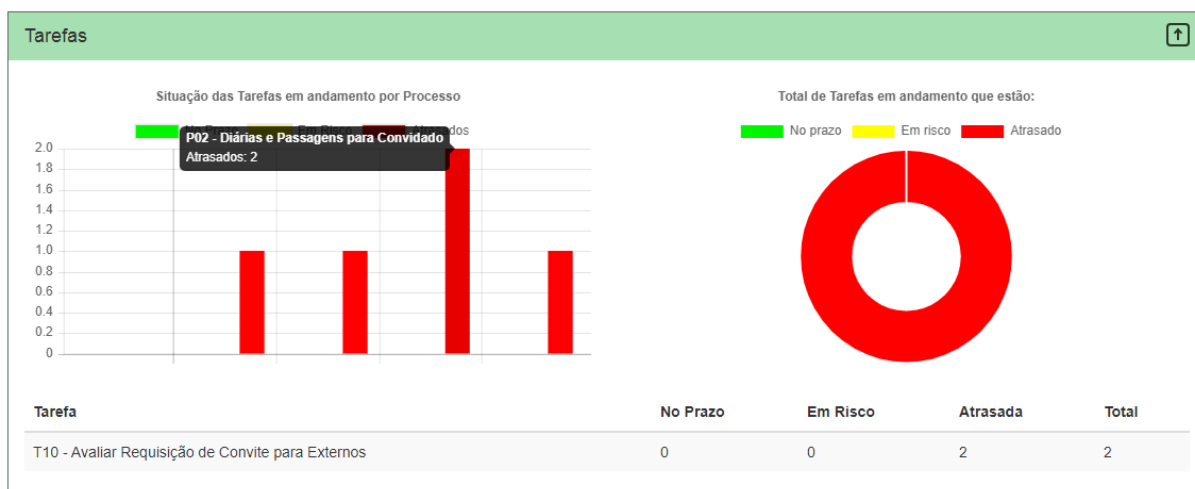
Para a situação apresentada, sendo utilizados dados reais, ocorreu a coincidência de cada tarefa estar associada a um único ator específico, fazendo com que os gráficos de tarefas e usuários ficassem iguais. Na prática, com o aumento do número de casos ativos simultaneamente, a relação entre gráficos irá gerar maior contraste. Para cada módulo é possível obter os detalhes em tabelas conforme ilustrado nas Figuras 30, 31 e 32.

Figura 30 - Resultado do módulo de monitoramento de casos de processos.



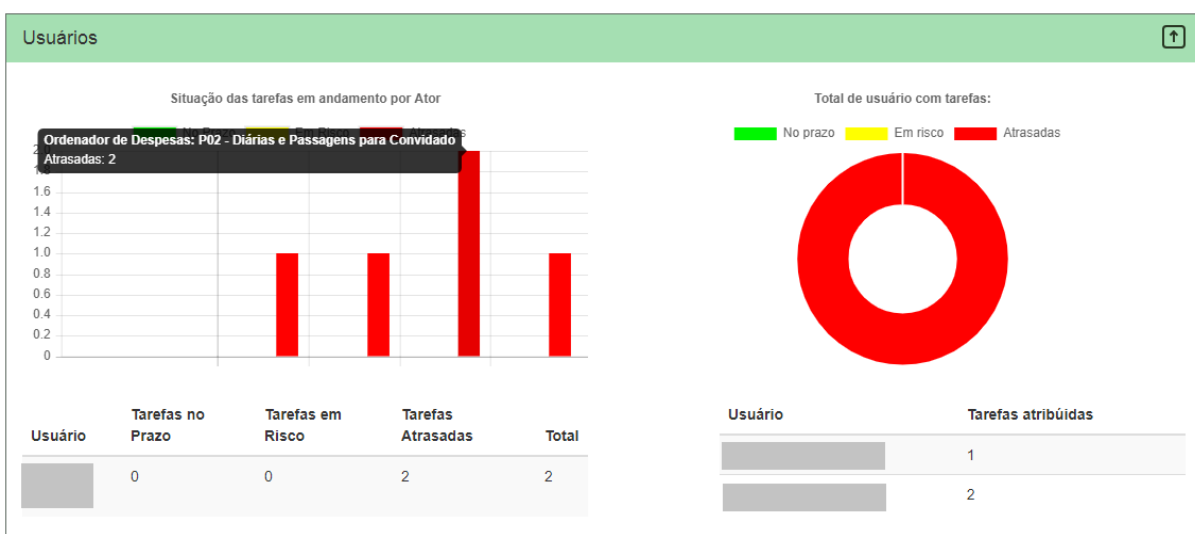
Fonte - Autoria própria.

Figura 31 - Resultado do módulo de monitoramento de tarefas.



Fonte - Autoria própria.

Figura 32 - Resultado do módulo de monitoramento de usuário.



Fonte - Autoria própria.

Os nomes dos usuários reais foram cobertos com a tarja cinza, evitando qualquer tipo de divulgação não autorizada.

Além do desenvolvimento da página, o trabalho resultou na obtenção de experiência na utilização de recursos para criação de elementos customizáveis utilizando recursos externos, até então, não utilizados em nenhum desenvolvimento do EAPn. Além da aplicação de recursos de Extensão de API REST que poderá vir a

serem aplicados em inúmeras situações, visando a melhoria de performance na utilização das APIs do Bonita, aquisição de dados externos e estruturação de futuras páginas de aplicação.

Assim, as implementações do presente trabalho podem ser utilizadas como ponto de partida para fundamentação de futuros projetos, caso sejam necessárias implementações associadas a este escopo.

9 CONCLUSÃO

A integração da página de monitoramento de atividades ao Portal de Processos da UFSC, estendendo os recursos disponíveis para monitoramento e análises, promoverá aos donos de processos e desenvolvedores do EAPn uma poderosa ferramenta para análise e posteriormente melhoria dos processos automatizados. Através de um sistema padronizado e analítico, será possível com maior propriedade e formalidade elaborar e definir estratégias.

Tomar decisões e prover ações em tempo de execução do processo é uma importante iniciativa para alavancar o desempenho da organização, auxiliando no cumprimento e acompanhamento das metas e prazos estipulados. Do ponto de vista informacional e operacional, é ampliado o escopo de conhecimento sobre a execução dos processos, tornando visível aos gerentes informações analíticas anteriormente desconhecidas.

O gerenciamento de desempenho proporcionará o direcionamento de todos os esforços em prol da efetividade das atividades do processo. Dessa forma, identificar se os objetivos traçados pela organização estão de acordo com a execução, garantindo a máxima eficiência. A utilização dos *dashboards* se apresenta bastante vantajosa devido ao aumento do número de casos de processos executados, possibilitando da forma mais eficiente possível a visualização de indicadores de desempenho.

Para o corpo comum de conhecimento BAM, esse trabalho é apresentado como um demonstrativo do potencial obtido com a agregação ao BPM. Assim, serve como referencial teórico e fundamentação para demais aplicações, devido o caráter genérico dos painéis desenvolvidos e a adequação aos conceitos previstos pelo gerenciamento de processos de negócios.

As pesquisas para embasamento do trabalho e referencial teórico apresentado podem ser continuadas visando agregar análises mais detalhadas acerca do assunto. Alicerçado em uma aplicação, resultados e testes de hipóteses podem ser elaborados para acrescentar conhecimento no contexto de BAM e BPM.

9.1 PERSPECTIVAS FUTURAS

A continuidade do projeto poderá ser dada pelo desenvolvimento de indicadores específicos à cada processo, incluindo informações do negócio. Assim,

pode-se obter maiores detalhamentos nas análises e gerar indicadores específicos capazes de identificar mais profundamente particularidades associadas à execução dos processos.

Além disso, o projeto pode ser estendido futuramente agregando tecnologias previstas dentro do contexto BAM, complementando o ambiente desenvolvido, tais como:

- Geração de relatórios situacionais em formato de documento;
- Definição de alertas e eventos automáticos;
- Identificação de variações de desempenho automáticas;
- Extensão dos recursos gráficos possibilitando maiores interações com o usuário.

Através dessas iniciativas é dado início à operacionalização de um ambiente visando a transformação digital do negócio, seguindo os conceitos iBPMS (*intelligent Business Process Management Suite*). Assim, complementando os serviços previstos pelo BPMS, é adicionando recursos objetivando a colaboração entre tecnologias para operações inteligentes de negócio.

REFERÊNCIAS

ASSOCIATION OF BUSINESS PROCESS MANAGEMENT PROFESSIONAL. **BPM CBOK - Guia para o Gerenciamento de Processos de Negócio**. [S./], ed. 1, v. 3, 2013.

ALMEIDA, V. N. **O que é Gerenciamento de Processos de Negócio: entenda como os processos podem afetar a sua organização**. Blog Euax Consulting. [S./], 2018.

BAQUERO, A. V.; PALACIOS, R. C.; MOLLOY, O. **Real-time business activity monitoring and analysis of process performance on big-data domains**. [S./], 2016.

BHATTI, M. I.; AWAN, H.; RAZAE, Z. **The key performance indicators (KPIs) and their impact on overall organizational performance**. [S./], 2014.

BHOWMIK, A.; JOHANNES, A. S. **Business Activity Monitoring | BAM - How Business Activity Monitoring Can Improve Agility And Effectiveness Of Next-Generation Payments Bussines**. [S./], 2018.

BONITASOFT. **What is Bonita?**. [S./], 2021. Disponível em: <https://documentation.bonitasoft.com/bonita/2021.1/what-is-bonita>. Acesso em: 04 mai. 2021.

BONITASOFT. **UI Designer overview**. [S./: s.d.]. Disponível em: <https://documentation.bonitasoft.com/bonita/2021.1/ui-designer-overview>. Acesso em: 06 mai. 2021.

BROCKE, J. V.; ROSEMAN, M. **Business Process Management**. Encyclopedia of Management, Vol. 7, [S. /], 2014.

CLAIR, LE C. **RPA, DPA, BPM, And DCM Platforms: The Differences You Need To Know. Apply Forrester's Automation Framework To Process Automation**. [S./], 2019.

DAHANAYAKE, A.; WELKE, R. J.; CAVALHEIRO, G. **Improving the understanding of BAM technology for real-time decision support**. International Journal of Business Information Systems. [S.l.], 2011.

DRESDNER, H. **Business Activity Monitoring: 'New Age' BI ?**. Gartner Inc. [S.l.], 2002.

GONÇALVES, H. P. **Guia para Modelagem e Automação de Processos de Negócios Acadêmicos: estudos de casos com processos da UFSC**, 2016.

GALIMBERTI, M. F.; MARIANI, A. C.; CORDEIRO, V. C.; TRIDAPALLI, J. V.; DE PIERE, E. R.; PERTERS, S. **Modelagem e Automação do Processo de Negócio "Afastamento Exterior (Inclusive com Diárias e Passagens)"**. Relatório Técnico INE 002. Universidade Federal de Santa Catarina, 2019.

GALIMBERTI, M. F.; MARIANI, A. C.; GONÇALVES, H. P.; CORDEIRO, G. V.; TRIDAPALLI, J. V. **Método de Modelagem e Automação de Processos de Negócios Acadêmicos com BPMS: estudo de caso com BPMS Bizagi e IES UFSC**. Relatório Técnico INE 001. Universidade Federal de Santa Catarina, 2019.

GARTNER, INC. **Selection Requirements for Business Activity Monitoring Tools**. [S.l.], 2005.

GOMES, P. C. T. **BAM - Monitoring of Business Activities**. Blog OpServices. [S.l.], 2016. Disponível em: <https://www.opservices.com/bam-monitoramento-das-atividades-de-negocio/>. Acesso em: 29 abr. 2021.

HAN, K. H.; HAN S. W.; CHOI, S.; KANG, J. **Business activity monitoring system design framework integrated with process-based performance measurement model**. WSEAS Transactions on Information Science and Application. Vol. 7, [S.l.], 2010.

HARMON, P. **Business Process Change**. Burlington, MA, USA. 2007.

JOSHI, M. **Five reasons why BAM and CEP have failed you**. Appdynamics, INC. [S./], 2015. Disponível em: <https://www.appdynamics.com/blog/product/five-reasons-why-bam-and-cep-have-failed-you/>. Acesso em: 29 abr. 2021.

JUFURU, V. **Business Activity Monitoring - Economic Impact on Industry Verticals**. BPTrends. [S./], 2007.

KANG, J. G.; HAN, K. H. **A Business Activity Monitoring System Supporting Real-Time Business Performance Management**. Third International Conference on Convergence and Hybrid Information Technology. Busan, South Korea, 2008.

KOLOWITZ, R. **The Forrester Wave: Software For Digital Process Automation For Deep Deployments. The 10 Providers That Matter Most And How They Stack Up**. [S./], 2019. Disponível em: https://iranbizagi.ir/wp-content/uploads/2020/06/The-Forrester-Wave%E2%84%A2_-Software-For-Digital-Process-Automation-For-Deep-Deployments-Q2-2019.pdf. Acesso em: 07 mai. 2021.

KUMAR, A. **Bonita REST API & Application Architecture**. [S./], 2013. Disponível em: <https://vitalflux.com/bonita-rest-api-application-architecture/>. Acesso em: 05 mai. 2021.

LEE, J.; LEE, D.; SUNGWON, K. **An Overview of the Business Process Maturity Model (BPMM)**. Seoul, Korea, 2007.

MAISEL, L. S. **Performance Measurement Practices: A Long Way from Strategy Management**. The Balanced Scorecard Report. p 12, 2001.

MORAIS, R. M. DE; DE PÁDUA, S. I. D.; COSTA, A. L. **An analysis of BPM lifecycles: From a literature review to a framework proposal**. Business Process Management Journal, [S./], v. 20, n. 3, p. 412–432, 2014.

MORTARI, C. E. **BAM - Uma visão geral**. Blog iProcess. [S./], 2014. Disponível em: <https://blog.iprocess.com.br/2014/10/bam-uma-visao-geral/>. Acesso em: 22 abr. 2021.

NAFIE, F.; ELTAHIR, M. A. **Real-Time Monitoring and Analyzing Business Process Performance**. International Journal of Engineering And Science. Vol. 6. [S.l.], 2016.

NEELY, A. D. **Measuring Business Performance: Why, What, and How**, The Economist and Profile Books Ltd. London, UK. p 5-6. 1998.

NEELY, A.; FRANCO-SANTOS, M.; KENNERLEY, M.; MICHELI, P.; MARTINEZ, V.; MASON, S.; MARR, B.; GRAY, D. **Towards a definition of a business performance measurement system**. Cranfield University, Cranfield, UK. 2007.

NEVES, A. **O Uso de Indicadores Chave de Desempenho para Avaliar a Eficiência dos Sistemas de Gestão**. Instituto Superior de Educação e Ciências. Lisboa, Portugal, 2012.

ROSEMANN, M.; BRUIN, T. de. **Application of a Holistic Model for Determining BPM Maturity**. [S. l.], 2005.

ROSEMANN, M.; DE BRUIN, T.; HUEFFNER, T. **A Model for Business Process Management Maturity**. ACIS 2004 Proceedings. 6, Queensland University of Technology, 2004.

ROZINAT. A.; AALST, V. D. W. **Conformance checking of process based on monitoring real behavior**. Information Systems Journal. [S.l.], 2008.

SHAFIEI, A.; HAJIHEYDARI, N. **Developing a business process management maturity model: A study of 300 Iranian superior companies**. [S.l.], 2014.

SONG, Y. **Developing Sequence Diagrams in UML**. Proceeding of the 20th International Conference on Conceptual Modeling: Conceptual Modeling. [S.l.], 2001.

TACLA, C. A. **Análise e Projeto OO & UML 2.0**. Universidade Tecnológica Federal do Paraná, [S.d.].

VOULET, P. **Make Your Existing Solution Tastier with Serverless Salt: Feasibility.** Blog BonitaSoft Community. [S.l.], 2018. Disponível em: <https://community.bonitasoft.com/blog/make-your-existing-solution-tastier-serverless-salt-feasibility>. Acesso em: 23 abr. 2021.

WEBMETHODS, INC. **Business Activity Monitoring (BAM) - The New Face of BPM.** [S.l.], 2006.

WSO2, INC. **About BAM.** Documentation WSO2. [S.l.: s.d.]. Disponível em: <https://docs.wso2.com/display/BAM200/About+BAM>. Acesso em: 29 abr. 2021.