

Universidade Regional do Noroeste do Estado do Rio Grande do Sul

Curso de Ciência da Computação

Departamento de Ciências Exatas e Engenharias

Shirlei Quevedo dos Santos

Simulação de uma Solução de Integração utilizando o ARENA®

Ijuí – RS.

2015.

Shirlei Quevedo dos Santos.

Simulação de uma Solução de Integração utilizando o ARENA®

Trabalho de Conclusão do Curso apresentado ao Curso de Ciência da Computação da Universidade do Noroeste do Estado do Rio Grande do Sul, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Orientadora: Professora Dra. Fabrícia Carneiro Roos Frantz.

Ijuí – RS.

2015.

Shirlei Quevedo dos Santos.

Simulação de uma Solução de Integração utilizando o ARENA®

Trabalho de Conclusão do Curso apresentado ao Curso de Ciência da Computação da Universidade do Noroeste do Estado do Rio Grande do Sul, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Banca Examinadora:

Professor Dr. Sandro Sawicki,
Coordenador da Disciplina

Professora Dra. Fabrícia Roos Frantz,
Presidente da Banca – Orientadora

Professor Dr. Rafael Zancan Frantz,
Membro

Ijuí – RS.

2015.

***Este trabalho é dedicado ao Eduardo, a
minha família, meus amigos e, em
especial aos meus avós: Geraldo e Noeli
(in memoriam).***

AGRADECIMENTOS

Em primeiro lugar, e não poderia ser diferente, quero agradecer a Deus pelo amparo dado nos momentos difíceis e pelas bênçãos concedidas a cada dia.

Aos meus amados avós Geraldo e Noeli, sem os quais eu nada seria, nada teria aprendido e nada conquistaria.

Aos meus demais familiares, em especial meus tios e tias que não me deixaram sozinha depois que meus avós partiram.

Ao meu querido Eduardo, amor da minha vida, imprescindível e indispensável para mim, ao qual agradeço pela paciência, carinho, cumplicidade, compreensão e amor.

A todos os professores com os quais tive a honra de conviver durante andamento de meu curso na Unijuí, em especial, a minha orientadora Professora Dra. Fabricia Roos Frantz e ao Professor Dr. Rafael Frantz os quais foram fundamentais para a realização deste trabalho.

Por fim, mas não menos importantes, minhas amigas e amigos queridos, que sabem o quanto são importantes para mim, pois eu lhes digo isso sempre que posso.

*“Lembramos continuamente, diante de
nosso Deus e Pai, o que vocês têm
demonstrado: o trabalho que resulta da fé, o
esforço motivado pelo amor e a
perseverança proveniente da esperança em
nosso Senhor Jesus Cristo. 1
Tessalonicenses 1:3”*

RESUMO

O mercado empresarial moderno demonstra a necessidade de se garantir processos de negócio ágeis, confiáveis e lucrativos. As organizações empresariais apostam em sistemas de informação robustos para atingir estes objetivos. Porém, na maioria dos casos, o ambiente de software das organizações empresariais é formado por diversos tipos de softwares, cada um com uma funcionalidade, uma característica da evolução do processo de negócio e das ferramentas computacionais. Neste cenário, surgiu a necessidade de se garantir a gestão da informação disponível no processo de forma unificada e confiável, havendo a necessidade então de fazer com que as diversas aplicações do ecossistema de software trabalhassem de forma integrada. Este é o campo de atuação da chamada EAI – Integração de Aplicações Empresariais. Este trabalho se propõe a descrever as características da EAI, bem como, criar uma proposta de solução de integração para um problema abstrato denominado Café. Para isso, será utilizado o recurso da simulação computacional, a qual possibilita que um processo do mundo real tenha seu funcionamento analisado através de uma ferramenta de software, sem a necessidade de sua efetiva implementação, garantindo assim a eficiência do processo e a economia de recursos.

Palavras Chave: Integração de Aplicações Empresariais, Simulação, Arena, Guaraná DSL.

ABSTRACT

The modern business market demonstrates the need to ensure agile business processes, reliable and profitable. Business organizations are betting on robust information systems to achieve these goals. However, in most cases, the software environment of business organizations is formed by various kinds of software, each with a feature, a feature of the evolution of the business process and computational tools. In this scenario, the need to ensure the management of the available information in a unified and reliable process, with the need then to make the various applications of the software ecosystem to work in an integrated manner. This is called the playing field EAI - Enterprise Application Integration. This paper aims to describe the characteristics of EAI as well as creating a proposal for integration solution to an abstract problem called Café. For this, the use of computer simulation will be used, which enables a real-world process has its operating analyzed by a software tool, without the need for its effective implementation, thus ensuring the efficiency of the process and saving resources.

Keywords: Enterprise Application Integration, Simulation, Arena, Guaraná DSL.

LISTA DE FIGURAS

Figura 1: Modelagem do Sistema a ser Simulado.....	19
Figura 2: Mundo x Modelo x Simulação.	24
Figura 3: Metodologia para Projeto de Simulação.....	27
Figura 4: Etapas do Projeto de Simulação.	28
Figura 5: Empresas que utilizam o ARENA®.	31
Figura 6: Interface principal do software ARENA®.....	34
Figura 7: Exemplo de fluxo de processo para o ARENA®.	34
Figura 8: Fluxo básico de um processo no ARENA®.....	35
Figura 9: Integração de Aplicações Empresariais.	36
Figura 10: Concepção do Guaraná.	43
Figura 11: Modelo de uma solução de integração.....	44
Figura 12: Arquitetura do Guaraná SDK.	45
Figura 13: Solução de integração com o Guaraná.	47
Figura 14: Modelo abstrato: Estudo de caso Café.	49
Figura 15: Modelo conceitual: Estudo de caso Café - Fluxograma.	49
Figura 16: Modelo computacional: Estudo de caso Café - ARENA®.	50
Figura 17: Cenários de Envio de Mensagens por segundo.....	55
Figura 18: Tempo médio de espera da entidade na fila.	56
Figura 19: Tempo médio de espera total no fluxo.	56

LISTA DE TABELAS

Tabela 1: Evolução das Ferramentas de Simulação.....	21
--	----

LISTA DE ABREVIATURAS E SIGLAS

EAI – EmpresariaisEnterprise Application Integration

DSL – Domain Specific Language

TI – Tecnologia da Informação

ACD – Activity Cycle Diagrams

GUI - Graphical User Interface

SGBD – Sistema Gerenciado de Banco de Dados

FTP – Protocolo de Transferência de Arquivos

RPC – Remote Procedure Call

RMI – Remote Method Invocation

ESB - Enterprise Service Bus

CORBA - Common Object Request Broker Architecture

SOAP - Simple Object Access Protocol

UML - Unified Modeling Language

EPC - Event-driven Process Chain

BPMN- Business Process Model and Notation

SUMÁRIO

Capítulo 1: Introdução	14
1.1. Motivação	15
1.2. Objetivos	16
1.3. Metodologia	16
1.4. Estrutura do Trabalho	17
Capítulo 2: Referencial Teórico	18
2.1. Simulação Computacional	18
2.2. Modelagem de Sistemas	19
2.3. Histórico da Simulação Computacional	20
2.4. Aplicação da Simulação	22
2.5. Modelos de Simulação	23
2.6. O Processo de Simulação	26
2.6.1. Os Softwares de Simulação	29
2.6.2. O ARENA®	30
2.7. Integração de Aplicações Empresariais	35
2.7.1. Estilos de Integração	37
2.7.2. Topologias de Integração	38
2.7.3. Padrões de Integração	38
2.7.4. Soluções de Integração	40
2.7.4.1. Linguagens de Propósito Geral	41
2.7.4.2. Linguagens de Propósito Específico	41
2.8. A Tecnologia Guaraná	43

Capítulo 3: Estudo de Caso	48
3.1 <i>Descrição do problema</i>	48
3.2 <i>A proposta de solução</i>	51
Capítulo 4: Resultados	54
4.1 <i>Apresentação dos Resultados</i>	54
Capítulo 5: Considerações Finais	57
5.1 <i>Conclusão</i>	57
5.2 <i>Contribuições</i>	57
5.3 <i>Trabalhos Futuros</i>	58
Referências Bibliográficas	59

Capítulo 1: Introdução

O crescimento exponencial da utilização de sistemas informatizados atinge grande parte dos setores da sociedade atual. As Tecnologias de Informação e Comunicação (TIC's) trouxeram ao cotidiano das pessoas uma série de funcionalidades e facilidades que garantem agilidade e confiabilidade na realização de diversas atividades.

Nas organizações empresariais estar sempre um passo a frente do concorrente pode ser vital para a saúde do negócio, o que torna indispensável a utilização de processos ágeis e confiáveis, destacando-se aqui a utilização dos sistemas de informação.

Geralmente as empresas buscam melhorar seus processos de negócio utilizando recursos da Tecnologia da Informação (TI). Muitas vezes, em um mesmo ambiente de negócio, várias aplicações com funcionalidades diferentes, são utilizadas, formando o que se denomina de ecossistema de software da empresa.

Com o passar do tempo, normalmente, as organizações empresariais passam a ter necessidade de que as aplicações integrantes de seus ecossistemas interajam umas com as outras, a fim de melhorar os processos de negócio, ou então, criar outros novos.

Neste cenário, surge o conceito de Integração de Aplicações Empresariais (do Inglês *Enterprise Application Integration - EAI*). A EAI visa proporcionar a unidade entre diversos softwares existentes no negócio, tornando-os transparentes quando de sua utilização (Frantz, 2012). Para Hohpe e Woolf (1999, *apud* Frantz, 2012) uma solução de integração faz com que as aplicações legadas sejam reutilizadas, todas as suas funcionalidades estejam disponíveis e também, possibilita que novas funcionalidades sejam criadas.

Porém, convém destacar que a implementação de soluções de integração além de acarretar custos à organização, pode conter erros, por isso é interessante que a solução de integração primeiramente seja submetida a uma simulação.

Através da simulação computacional podemos estudar a viabilidade da solução projetada, sem a necessidade de sua implementação no ambiente real, o que traz algumas vantagens, como por exemplo, a possibilidade de detecção de falhas no processo e a economia de recursos financeiros.

Atualmente existem diversas ferramentas para simulação computacional, bem como, para a integração de aplicações. Neste trabalho optamos em utilizar o software para simulação denominado ARENA®, o qual é largamente utilizado para este fim. Já em relação à integração de aplicações, analisamos a Linguagem de Domínio Específico Guaraná DSL¹.

Desta forma, nos propusemos a realizar a transformação de um problema abstrato, em uma solução de integração, simulando seu funcionamento no ambiente computacional, a fim de observar o comportamento do software e promover uma análise de resultados, buscando referência no modelo criado através da Guaraná DSL apresentado por Frantz (2012).

1.1. Motivação

A maioria dos sistemas de informação das empresas atuais foram implantados ao longo dos anos de atividades dessas empresas, acompanhando a evolução dos processos de produção e gestão, gerando um ecossistema de software heterogêneo.

A integração das aplicações que fazem parte do ecossistema de software é uma tarefa importante para as organizações empresariais e que exige suporte tecnológico, pois não seria conveniente usar soluções *ad-hoc*.

Existem atualmente uma série de tecnologias para a integração de aplicações, as quais ainda estão em pleno desenvolvimento. Uma destas tecnologias denomina-se Guaraná, e ela possui uma DSL para modelar soluções de integração.

A modelagem conceitual de uma solução com a tecnologia Guaraná DSL representa de forma abstrata o que uma solução deve fazer, porém, para executar uma solução de integração é necessário implementá-la.

Implementar uma solução de integração é uma atividade custosa e propensa a erros. Simular uma solução de integração antes de sua implementação seria uma forma de poder medir algumas de suas propriedades antes de sua implementação,

¹ Linguagem de domínio específico (DSL), do inglês Domain Specific Language). Linguagem específica criada para modelar um determinado conjunto de problemas.

ou seja, analisar dados relevantes ainda na fase do projeto. Para simular uma solução de integração é necessário utilizar ferramentas da área de simulação.

1.2. Objetivos

O objetivo geral deste trabalho propor um modelo de simulação para uma solução de integração de aplicações e implementá-lo implementado por meio da ferramenta de software ARENA®. Os dados obtidos por meio da simulação realizada com o uso do ARENA® serão analisados e apresentados de forma gráfica.

Com a realização deste estudo, serão abordados conceitos práticos e teóricos acerca da modelagem de sistemas, integração de aplicações empresariais, e simulação computacional.

Entendemos que o estudo da integração de aplicações empresariais por meio da simulação computacional é de grande valia acadêmica frente à sua crescente aplicação no mercado atual, assim sendo, podem ser apresentados os seguintes objetivos específicos para o presente trabalho:

- i. Conhecer o campo de estudos relativo à simulação computacional;
- ii. Explorar as funcionalidades da ferramenta ARENA®;
- iii. Adquirir conhecimento acerca dos conceitos e aplicações da Integração de Aplicações Empresariais;
- iv. Efetuar a transformação de um problema abstrato em modelo de simulação;
- v. Simular o funcionamento de uma solução de integração baseado na Integração de Aplicações Empresariais;

1.3. Metodologia

A metodologia empregada para o desenvolvimento deste trabalho apresenta-se dividida em quatro etapas: primeiramente, a partir de bases bibliográficas, artigos científicos, livros especializados e pesquisas na Web, foi realizada uma revisão da literatura no que tange a Simulação Computacional e Integração de Aplicações Empresariais, tendo o foco voltado para o uso do software Arena e do estudo da Linguagem Guaraná.

Em seguida, na segunda etapa, fez-se um estudo detalhado do software ARENA® mediante o uso de manuais. A terceira etapa consistiu na transformação de um modelo abstrato em uma solução de integração de aplicações no ambiente do software ARENA®. A quarta e última etapa centrou-se na análise dos resultados obtidos com a simulação do modelo proposto.

1.4. Estrutura do Trabalho

Este trabalho está organizado da seguinte forma: no Capítulo 2 é apresentado o referencial teórico que embasou o desenvolvimento do presente trabalho, conceituando Simulação Computacional e Integração de Aplicações Empresariais, respectivamente; o Capítulo 3 descreve o estudo de caso analisado e a proposta de solução desenvolvida; no Capítulo 4 são apresentados os resultados obtidos no processo; o Capítulo 5 aborda as considerações finais seguindo a sequência: conclusão, contribuições e trabalhos futuros.

Capítulo 2: Referencial Teórico

Neste capítulo são apresentados os principais conceitos relativos e necessários ao desenvolvimento deste trabalho: Simulação Computacional e Integração de Aplicações.

2.1. *Simulação Computacional*

Simular problemas, soluções e processos vêm se mostrando uma abordagem de estudo de destaque nas mais variadas áreas do conhecimento, tais como: engenharias, ciências exatas, biológicas, humanas e ambientais. Isso vem sendo observado em razão do crescimento da complexidade dos problemas propostos, bem como, do aumento exponencial dos recursos computacionais.

Assim, este crescimento na utilização das TIs ajudou a impulsionar a área da simulação computacional, tanto no desenvolvimento de ferramentas, quanto na sua efetiva aplicação no mercado atual.

Pela definição dada por Pegden (1995, apud Lemos, 1999) a simulação é o desenvolvimento de um modelo lógico que reproduz a realidade, a fim de avaliar o comportamento e o desempenho de sistemas sob as mais variadas condições.

Segundo Prado (1999), diversas áreas vêm utilizando, cada vez mais, o suporte da simulação para análise de uma gama diversificada de sistemas: sistemas de manufatura, transportes, sistemas de informação, sistemas militares, sistemas de serviço público, etc. Portanto, a simulação caracteriza-se como uma ferramenta a ser utilizada não somente em uma área específica, mas sim, uma ferramenta que pode ser utilizada em qualquer tipo de sistema que possa ser modelado computacionalmente.

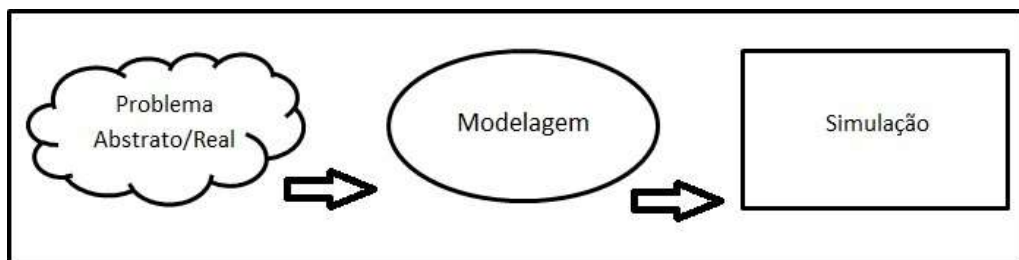
Neste contexto, a Simulação firmou-se como uma poderosa ferramenta no suporte à tomada de decisões para o desenvolvimento de sistemas mais eficientes, como dito por HARREL *et al.* (2002 apud Sabbadini e Gonçalves, 2005). Seu uso quando da execução de projetos é uma excelente técnica preditiva e preventiva, visando assim tanto dar suporte ao desenvolvimento de novos projetos, quanto melhorar aqueles já existentes.

Ante a todo rol de aplicações, cada vez mais impulsionadas pelo mercado extremamente competitivo, as organizações empresariais vêm se utilizando das técnicas de simulação computacional para garantir qualidade, eficiência e confiabilidade do desenvolvimento e aprimoramento de seus processos de negócio.

2.2. Modelagem de Sistemas

Para podermos utilizar a simulação computacional em um sistema, nos deparamos primeiramente com a necessidade de modelar o ambiente a ser simulado, ou seja, precisamos planejar, organizar, definir como o sistema opera, ou como desejamos que ele venha a operar.

Figura 1: Modelagem do Sistema a ser Simulado.



Fonte: autoria própria.

Para Prado (1999), o estudo da modelagem de sistemas engloba uma série de atividades para se adequar o sistema às necessidades desejadas, como por exemplo, alterações de layout, troca de equipamentos, automatização de processos.

Estas alterações buscam dimensionar o sistema adequadamente a fim de se reduzir ao máximo os pontos de gargalo, ou seja, aqueles pontos onde são formadas filas que podem fazer com o processo passe a ter baixo desempenho, ou até mesmo vir a falhar.

De acordo com Trivelato (2003), temos que:

“Um modelo é a representação do conhecimento e a principal ferramenta para o estudo do comportamento de sistemas complexos. Quando o modelo é uma representação válida de um sistema, informações significativas podem ser retiradas sobre sua dinâmica ou seu desempenho.”

A partir disso, fica claro que, ao termos um modelo computacional adequado ao processo de negócio, podemos submetê-lo à simulação computacional para verificar a sua eficácia, reduzindo custos e garantindo sua maior operacionalidade.

2.3. Histórico da Simulação Computacional

Como já mencionado anteriormente, no que tange a análise de projeto e operação de sistemas, atualmente a simulação computacional apresenta-se como uma das ferramentas mais poderosas do mercado de TI. A análise de um projeto de sistema de negócio por meio da simulação permite, entre outros, a verificação da viabilidade técnica, econômica e funcional do projeto. (Porto *et al.*, 1999).

Os simuladores produzem respostas prováveis a certas situações testadas, ou seja, representam o comportamento de um sistema real ou abstrato. O objetivo da simulação é analisar e entender o comportamento do sistema testado, submetendo-o a diversas ações e decisões alternativas. Pedgen (1995, apud Lemos, 1999).

A definição para o termo Simulação dada por Prado (1999) é:

“Simulação é uma técnica de solução de um problema pela análise de um modelo que descreve o comportamento do sistema usando um computador digital”.

Já em Oliveira (1994, apud Sabbadini, Gonçalves, 2005), a simulação é conceituada como um conjunto de entidades abstratas, representativas de objetos reais, entre as quais se estabelecem relacionamentos.

Segundo Harrel *et al.* (2002, apud Sabbadini e Gonçalves, 2005) um processo de experimentação de um sistema real usado para determinar seu comportamento é chamado de simulação. Para Banks e Carson (1984, apud Sabbadini, Gonçalves, 2005), a simulação é a imitação da operação de processos reais ou sistemas, em um dado período de tempo.

Podemos inferir então que a simulação, enquanto processo computacional é a experimentação de um modelo do mundo real, através de uma ferramenta de software, com o objetivo prever seu comportamento, auferindo-se sua viabilidade técnica e garantindo, principalmente, a eficiência em sua aplicação concreta e economia de recursos.

Foi na década de sessenta que surgiram às primeiras linguagens para simulação, decorrentes do advento do surgimento dos computadores. Até então a modelagem de sistemas era realizada manualmente através da Teoria das Filas, que é um método analítico que utiliza fórmulas matemáticas para a elaboração da modelagem de sistemas (Prado, 1999).

Porém, de acordo com o apresentado em Prado (1999), a década de setenta foi considerada um marco na evolução da simulação devido ao grande volume de trabalhos produzidos, fato este atribuído ao surgimento dos mainframes.

No estudo abordado em Porto e Lobão (1999), nota-se que as ferramentas e linguagens de simulação evoluíram na mesma proporção em que evoluíram hardware e software, assim, em cinquenta anos, passamos dos modelos matemáticos para simuladores inteligentes interativos munidos de interface gráfica.

Na tabela abaixo, adaptada de Porto e Lobão (1999), Gavira *et al.* (2002) e Prado (1999) pode-se observar a relação entre a evolução da tecnologia disponível e os sistemas de simulação existentes. Esta relação encontra-se de acordo com a linha de tempo apresentada por Chwif e Medina (2006).

Tabela 1: Evolução das Ferramentas de Simulação.

Localização no Tempo	Ferramenta de Simulação	Tecnologia Disponível
Década de 50	Linguagens de Programação de Propósito Geral: FORTRAM e ALGOL.	Surgimento do Computador.
Década de 60	Primeiras Linguagens de Simulação: GPSS e SIMULA.	Computadores usados comercialmente.
Década de 70	Década de ouro da Simulação: Surgimento de novas Linguagens: GASP, SIMSCRIPT e EXELSIM.	Computadores de grande porte – Mainframes.
Década de 80 e 90	Simuladores de alto nível: Arena, Taylor, Promodel, AudoMod.	Computadores pessoais com alto desempenho, recursos de animação, simuladores visuais.
Após década de 90	Pacotes Flexíveis de Programas para simulação.	Animações mais poderosas, integração com outras linguagens de programação, facilidade de uso.

Fonte: Adaptado de Porto e Lobão (1999) e Gavira *et al.* (2002).

Mediante esta evolução no campo da simulação, dentre as linguagens e softwares comumente utilizados, elencamos algumas principais, citadas por Gavira *et al.* (2002) e Frantz *et al.* (2014):

- Fortran: Linguagem de propósito geral também aplicada em simulação;
- GPSS: General Purpose Simulation System – torna mais amigável a construção de modelos computacionais para certos tipos de eventos discretos;
- System Dynamics – SD: programas baseados em sistemas dinâmicos utilizados para estudar e administrar sistemas de realimentação complexos;
- Ansys: ferramenta amplamente utilizada na área das engenharias que permite o desenvolvimento de plataformas de testes para simulação;
- MatLab/Simulink: utilizada para simulação, modelagem e análise de sistemas;
- AMESim e AnyLogic: aplicado à simulação em diversos domínios de aplicação;
- Arena: software aplicado na simulação discreta de diversos segmentos, sendo considerado uma das mais poderosas ferramentas de simulação.

2.4. Aplicação da Simulação

Conforme citado em Prado (1999), o recurso da simulação vem sendo aplicado nas mais diversas áreas operacionais: hospitais, terminais de transportes, linhas de produção, logística, comunicações, enfim, uma vasta gama de aplicações empresariais e ainda, largamente utilizado para fins acadêmicos.

Segundo Sabbadini e Gonçalvez (2005), a simulação como ferramenta para análise e resolução de problemas complexos tem as seguintes aplicações:

- a) Projeto e análise de plantas de negócio; (implementação de dimensionamento e análise de viabilidade)
- b) Análise de processos diversos;
- c) Compreensão de sistemas reais;

- d) Estudo de sistemas complexos; (análise do efeito de mudanças operacionais);

Assim, mediante o já exposto, é notável que a simulação computacional permite que sejam analisadas alterações em sistemas e plantas físicas de negócio, sem a necessidade de implanta-las no mundo real, reduzindo custos e garantindo eficiência do processo.

2.5. Modelos de Simulação

Um modelo de simulação pode ser conceituado como uma abstração lógica de um problema ou sistema real. Os modelos de simulação são aqueles desenvolvidos para serem testados. Eles diferem dos modelos de otimização, os quais são resolvidos.

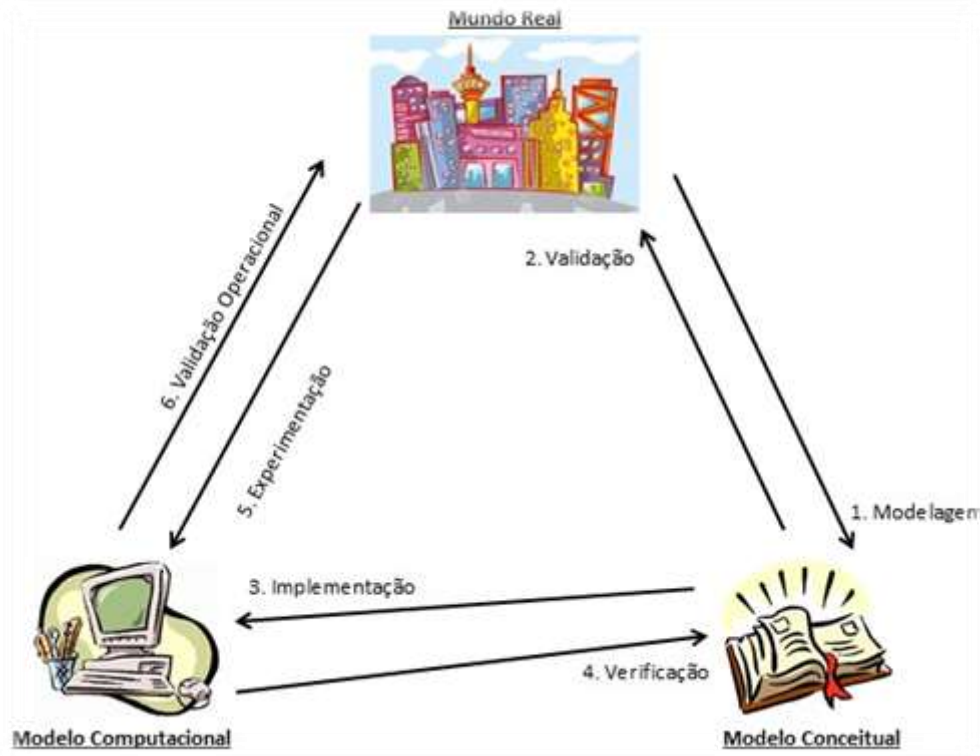
Comumente, os modelos de simulação são usados para analisar o sistema real a ser implementado ou melhorado. A modelagem pressupõe um processo de criação, uma boa descrição do objeto e um determinado grau de abstração.

De acordo com Chwif e Medina (2006), um modelo é:

“uma representação do mundo real, ou pelo menos de parte dele. Portanto, a validação de um modelo é realmente muito direta – em princípio. Tudo o que devemos fazer é checar se o modelo comporta-se como o mundo real sob as mesmas condições. Se ele se comporta, então é um modelo válido, caso contrário, não é válido.”

A figura abaixo demonstra o conceito de modelo de simulação em relação ao mundo real apresentada por Chwif e Medina (2006):

Figura 2: Mundo x Modelo x Simulação.



Fonte: adaptado de Chwif e Medina (2006).

Partindo-se do problema do mundo real, de acordo com Chwif e Medina (2006), a **representação** do processo de simulação pode se dar através das seguintes formas:

- Modelo Abstrato;
- Modelo Conceitual;
- Modelo Computacional;
- Modelo Operacional.

Os autores defendem a importância da utilização de modelo conceitual como forma de garantir a eficácia do sistema a ser simulado, ou seja, assegurar a confiabilidade do modelo computacional. Para isso, os autores citam a ferramenta ACD – *Activity Cycle Diagrams* para a elaboração de modelos conceituais, os quais poderão ser traduzidos de forma mais amigável para o modelo computacional, mas esta atividade também pode ser realizada com o uso de outras ferramentas, como por exemplo, a UML.

Estes mesmos autores, classificam os modelos de simulação em tipos, os quais podem ser:

- Modelo Simbólico: símbolos e gráficos (fluxogramas, DFD, layouts);
- Modelo Analítico: modelagem matemática (programação linear, teoria das filas);
- Modelo de Simulação: análise do comportamento do sistema de forma dinâmica e aleatória.

Os modelos de Simulação podem ser computacionais e não computacionais. Os modelos não computacionais, como por exemplo, os protótipos, não são nosso foco aqui neste trabalho, por isso tratamos acerca apenas dos Modelos Computacionais.

Para estes modelos de simulação computacional, abaixo está descrita a classificação adotada por Banks e Carson (1984, apud Sabbadini e Gonçalves, 2005):

- Estáticos ou Dinâmicos: Modelos Estáticos são aqueles modelos nos quais o tempo não é um fator relevante, ou seja, representam o sistema em um ponto específico de tempo. Modelos Dinâmicos representam as mudanças sofridas no sistema ao longo do tempo, como por exemplo, a simulação do funcionamento de um estabelecimento comercial no horário das 08:00 às 22:00 horas.
- Determinísticos ou Estocásticos: Os Modelos Determinísticos não contém variáveis aleatórias, seu conjunto de entradas resulta em um único conjunto de saídas. Os Modelos Estocásticos possuem uma ou mais variáveis aleatórias de entrada, o que resulta em saídas aleatórias.
- Contínuos ou Discretos: Quando as variáveis modificam-se continuamente ao longo do tempo, temos um Modelo Contínuo. Já quando as variáveis se modificam em um conjunto de pontos específicos do tempo, alterando o estado do sistema do mesmo modo, diz-se que o Modelo é Discreto.

Por fim, destacamos os princípios considerados essenciais, que devem ser observados quando do desenvolvimento de um modelo de simulação, de acordo com os autores Banks e Carson (1984, apud Sabbadini e Gonçalves, 2005):

- Deve ser de fácil compreensão;
- A característica evolucionária deve estar presente, ou seja, do menor grau de complexidade para o maior;
- Os objetivos e propósitos devem estar definidos e direcionados;
- Todas, ou a maior parte, das questões importantes devem estar contempladas;
- Possuir possibilidade de atualização e adaptação
- Os usuários devem ser ouvidos quando da elaboração do modelo.

2.6. O Processo de Simulação

Uma vez apresentados os principais conceitos de simulação, cabe agora discutir acerca do processo a ser adotado para criação uma simulação propriamente dita.

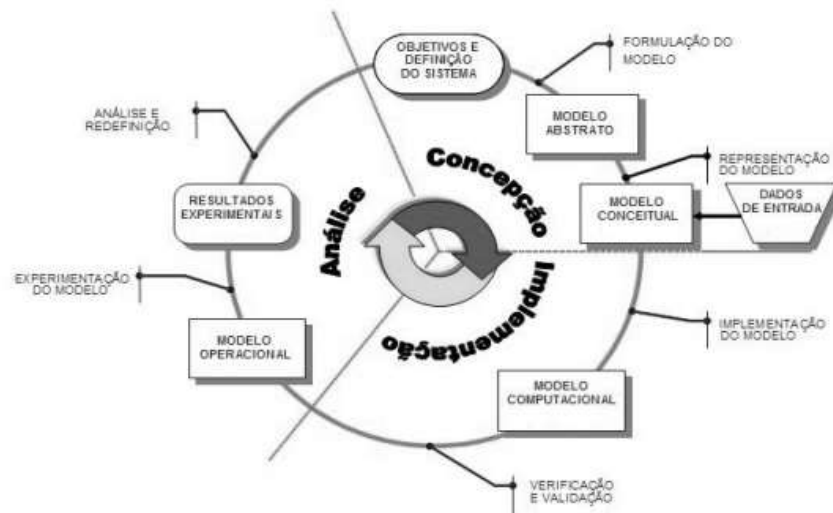
Em resumo, para a construção de um modelo de simulação devemos primeiramente elaborar um modelo conceitual, a partir dele, constrói-se o modelo computacional capaz de imitar o comportamento do sistema e depois, é feita a simulação através das ferramentas disponíveis, com o objetivo de se analisar os resultados obtidos (Sabbadini e Gonçalves, 2005).

Existem diversas metodologias para desenvolvimento de um processo de simulação, abaixo seguem algumas delas:

- a) Chwif e Medina (2006) sugerem três etapas para os projetos de simulação: concepção, implementação e análise. Na primeira etapa, o sistema é descrito de forma clara, apresentando os objetivos e os dados de entrada. A implementação corresponde a elaboração do modelo conceitual do sistema, que posteriormente é traduzido em modelo computacional para ser analisado através dos resultados gerados quando de sua execução. Na terceira e última etapa, com base nos resultados

obtidos através da simulação, pode-se auferir as conclusões, melhorias e falhas podem ser apuradas. A figura a seguir demonstra de forma visual esta metodologia.

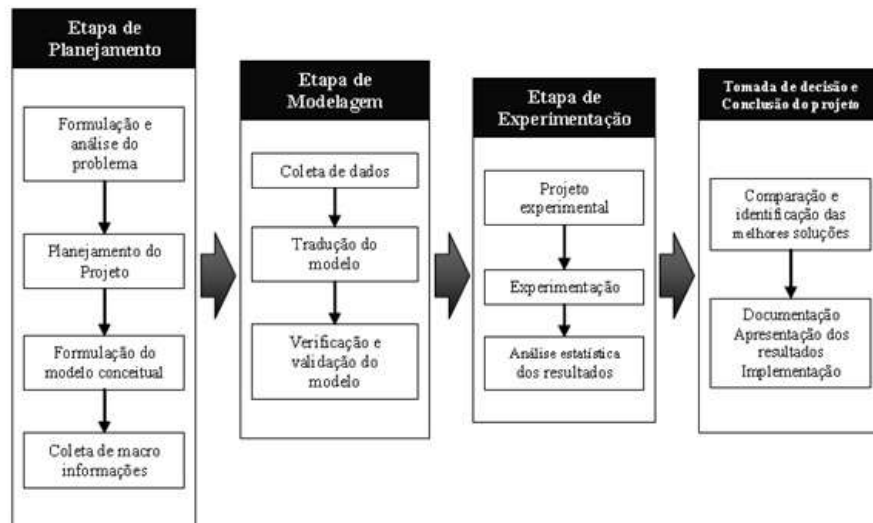
Figura 3: Metodologia para Projeto de Simulação.



Fonte: Chwif e Medina (2006).

- b) Para Prado (2004, apud Landa, 2007) o projeto de simulação compreende duas etapas: planejamento e execução. No planejamento o problema é identificado, bem como são determinadas as metas do sistema, coletados e validados os dados, desenvolvido um protótipo e por fim, criado o cronograma de execução. Na fase da execução o sistema proposto deve ser simulado e com os resultados gerados, apresentado o relatório final.
- c) Na metodologia proposta por Freitas (2001, apud Landa, 2007) o processo de criação do modelo de simulação é desenvolvido em quatro etapas, conforme exibido na figura abaixo:

Figura 4: Etapas do Projeto de Simulação.



Fonte: Freitas (2001, apud Landa, 2007).

Segundo Sabbadini e Gonçalves (2005) esta metodologia também é proposta por Shannon (1998), Banks (1999), Banks e Carson (1984) e Harrel *et al.* (2002), por isso, dada sua relevância, para melhor compreensão, cada uma das etapas representadas no modelo acima, é descrita de forma sucinta em seguida:

- I) **Formulação e Análise do Problema:** compreensão do problema, quais os resultados esperados e como será validado;
- II) **Planejamento do Projeto:** definição das necessidades a serem supridas com o projeto, bem como, seu cronograma de execução;
- III) **Formulação do Modelo Conceitual:** a partir do modelo conceitual é gerado o modelo computacional, por isso devem ser definidas as variáveis, os parâmetros, os componentes e as interações lógicas;
- IV) **Coleta de Macro Informações e Dados:** organização das informações para que possa ser realizada a análise crítica destas;
- V) **Tradução do Modelo:** transcrição do modelo conceitual para o modelo computacional, utilizando uma ferramenta de simulação;
- VI) **Verificação e Validação:** aqui o sistema é testado e os resultados obtidos analisados em relação ao modelo real;
- VII) **Projeto Experimental Final:** experimentos são projetados objetivando a possibilidade de determinar como cada processo será executado;

VIII) Experimentação: execução das simulações para a obtenção dos dados de análise;

IX) Interpretação e Análise dos Resultados: a partir dos indicadores apresentados na simulação, analisar o resultado e verificar a necessidade da realização de novas rodadas de simulação;

X) Comparação de Cenários e Identificação das Melhores Soluções: comparar os cenários criados a fim de identificar a melhor opção;

XI) Documentação, Apresentação dos Resultados e Implementação: geração da documentação do sistema e apresentação dos resultados obtidos, para que então, caso aprovado, seja implementado.

Podemos concluir aqui que o uso de metodologias para o processo de criação de modelos de simulação orienta de forma lógica o trabalho, serve de roteiro sistemático para o processo e simplifica o desenvolvimento da solução do problema.

2.6.1. Os Softwares de Simulação

Conhecendo o processo de desenvolvimento de um modelo de simulação, a sua implementação se dá por meio de uma ferramenta computacional. Atualmente, existem diversos softwares voltados para essa funcionalidade.

Conforme descrito em Gavira (2003), os simuladores atuais, com o auxílio da computação avançada, fornecem ferramentas poderosas para as atividades de modelagem e simulação. Dentre os simuladores disponíveis no mercado, convém destacar os seguintes: Arena, Extend, Stella, AutoMod, ProModel e PowerSim.

Como principais características encontradas nestes softwares de simulação, destacam-se: Gavira (2003)

- Interface gráfica amigável ao usuário e Animação de modelos;
- Coleta automática das variáveis de saída para medição de desempenho do sistema e relatórios em tempo real;
- Orientação a objetos e importação de sub-rotinas externas;
- Tratamento de dados da simulação através de ferramentas estatísticas;
- Uso de sentenças lógicas similares à programação;

2.6.2. O ARENA®

Neste trabalho, em razão da interface amigável e da disponibilidade de versão gratuita, optamos em utilizar o software ARENA®, para o qual, deste ponto em diante, passaremos a apresentar as principais características.

O software ARENA® é um ambiente gráfico integrado de simulação. Nele não é necessário escrever nenhuma linha de código, pois todo o processo de criação do modelo de simulação é gráfico, visual, e de maneira integrada. Nele estão contidos todos os recursos para modelagem de processos: desenho, animação, análise estatística e análise de resultados (Prado, 1999).

No Brasil, o software ARENA® foi lançado em 1993 pela empresa americana Systems Modeling, a qual mais tarde, no ano de 1998 foi incorporada pela empresa Rockwell Software. Ele é o sucessor de dois outros softwares criados para simulação: o SIMAN e o CINEMA. Atualmente, o ARENA® está em sua versão 14.7.

A empresa Paragon Decision Science (www.paragon.com.br) é a representante do ARENA® no Brasil e na América Latina, sendo o único Training Center da América Latina certificado para capacitar profissionais neste software. O download da versão student do software pelo ser obtido através do site da empresa.

A versão “Student” do ARENA®, segundo o site da Paragon possui as mesmas funcionalidades da versão “Professional”, apresentando apenas restrição no tamanho do fluxo do processo de simulação.

De acordo com site da representante do ARENA® no Brasil, sendo uma poderosa ferramenta para análise de cenários, ele passa a ser essencial à medida em que os sistemas aumentam de tamanho e de complexidade. ARENA® (2015)

Com o ARENA®, mediante a análise dinâmica dos processos e a interação de elementos, é possível determinar pontos de gargalos, melhores condições de operação, comportamento de filas, ocupação de recursos enfim, o comportamento geral do modelo simulado. ARENA® (2015)

No site da empresa Paragon, como principais clientes utilizadores do ARENA® estão elencadas as empresas conforme a figura mostrada logo abaixo:

Figura 5: Empresas que utilizam o ARENA®.



Fonte: site da Paragon. ARENA® (2015).

Além de ser uma linguagem de simulação, o ARENA® também é um ambiente de trabalho onde o modelo pode ser testado, apresentado seus resultados e ainda, visualizado seu funcionamento através de recursos de animação. Prado (1999).

Segundo Prado (1999), basicamente o ARENA® enxerga o sistema a ser modelado como um conjunto de estações de trabalho que prestam serviços a clientes ou entidade que se movem através do sistema. Para a construção de um modelo de simulação com o ARENA®, primeiramente deve ser construído um fluxograma do processo o qual deve ser constituído pelas estações de trabalho e as opções de fluxo entre as entidades.

O processo de construção dos modelos é realizado por meio de uma Interface Gráfica para o Usuário (do termo em Inglês *GUI - Graphical User Interface*), bastante intuitiva, onde o mouse possui papel de destaque.

Prado (1999) descreve de forma detalhada as características e elementos do software, um bom detalhamento do ARENA® também pode ser encontrado em Miranda *et al.* (2006), a seguir estão descritos alguns itens:

- *Basic Process Panel*: contém os módulos essenciais à elaboração de um projeto de simulação;

- *Advance Transfer Panel*: com os módulos específicos para a configuração do transporte das entidades no modelo;
- *Advance Process Panel*: tem os módulos específicos para a configuração dos processos no modelo;
- *Report Panel*: detém os relatórios com os resultados da simulação;
- *Navigate Panel*: possibilita a criação de teclas de atalho para uma navegação mais eficiente e rápida;
- *Create*: é responsável pela entrada dos dados no sistema, o estágio inicial do processo;
- *Dispose*: corresponde ao fim do fluxo do processo simulado;
- *Process*: módulo responsável pela ação do processo da entidade que representa;
- *Assign*: usado para alternar o valor de algum parâmetro das entidades do modelo;
- *Resource*: elementos necessários à realização de um serviço;
- *Entity*: módulo de dados que define os vários tipos de entidades do modelo;
- *Queue*: apresenta as informações relativas ao estado das filas ocasionadas no processo;
- *Flow*: são os fluxos, ou seja, os caminhos percorridos pelas entidades ao longo do sistema;
- *Variable*: módulo usado para definir a dimensões e valores iniciais de uma variável;
- *Station*: usado na definição de uma estação (ou um conjunto de estações) que corresponde a posição física ou lógica do local onde as entidades são processadas;
- *Storage*: blocos para o armazenamento de entidades produzidas quando da execução do modelo;
- *Sets*: permitem que recursos de um mesmo tipo sejam agrupados em forma de conjunto;
- *Transport*: módulo responsável pelo transporte da entidade a ser processada até a sua estação de destino;

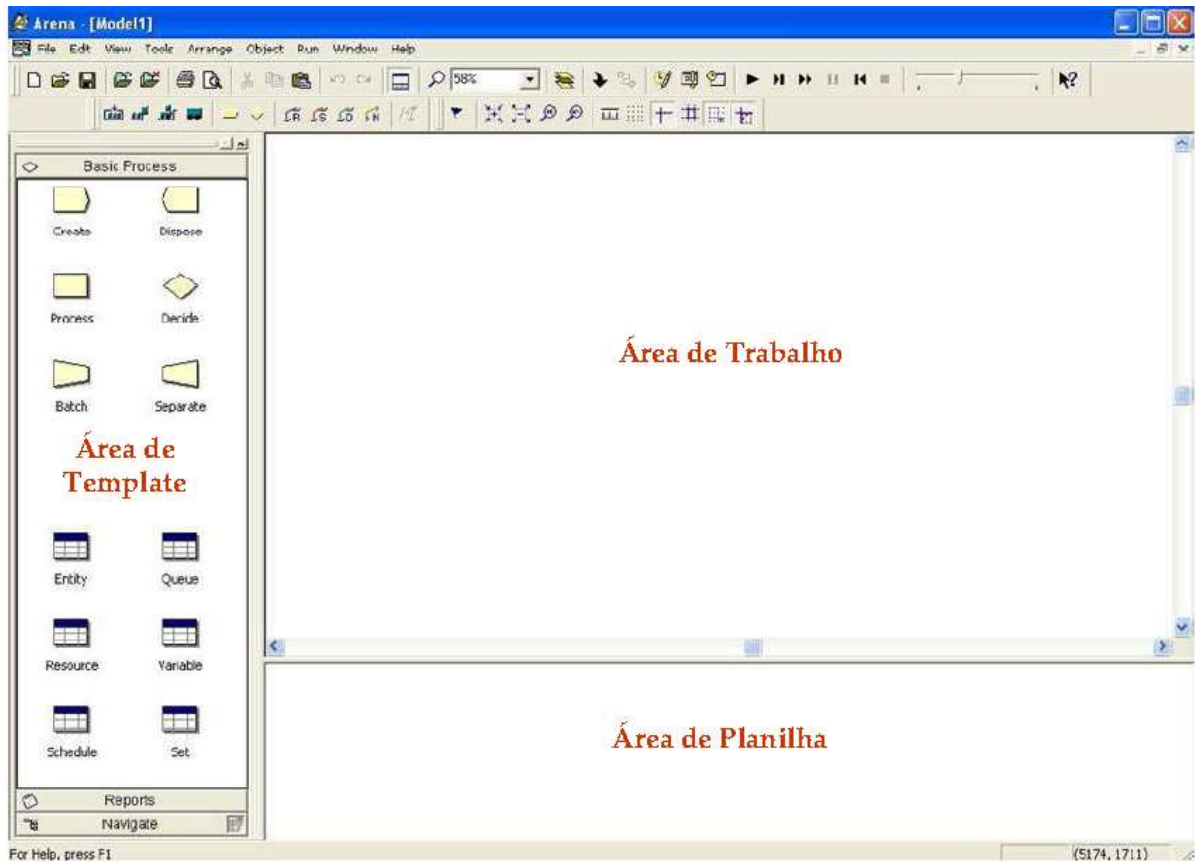
- *Request*: atribui uma unidade de transporte a uma entidade e move a unidade para a posição da entidade;
- *Network*: permite configurar a rede de trajetos dos transportadores.
- *Templates*: são os painéis de módulos, ou seja, arquivos com extensão *.tpo* que contêm um módulo do processo;
- *Animate*: recurso que permite a animação de rotas e entidades do modelo durante o processo de simulação.

Além dos elementos acima citados, o ARENA® possui algumas ferramentas muito úteis, que merecem ser também relacionadas de acordo com o apresentado em Miranda *et al.* (2006):

- Analisador de dados de entrada (*Input Analyser*): permite a análise de dados reais do funcionamento do processo, bem como a escolha da melhor distribuição estatística que se aplica a eles. Esta distribuição pode ser incorporada diretamente ao modelo;
- Analisador de resultados (*Output Analyser*): é a ferramenta através da qual pode ser realizada a análise dos dados coletados durante a simulação, sendo que esta análise pode ser gráfica;
- Visualizador da simulação (*ARENA Viewer*): permite que um modelo, previamente desenvolvido com o ARENA®, seja executado em outro computador com o ARENA Viewer instalado, sem necessidade de chave de proteção;
- *Scenario Manager*: permite que se execute um conjunto de simulação na modalidade *batch* (em lotes) para análise posterior.

Para a criação de um modelo de simulação com o ARENA® os elementos devem ser montados na Área de Trabalho, com o uso dos componentes disponibilizados na Área de Templates. A Área de Planilha exibe as propriedades de cada elemento integrante do fluxo criado.

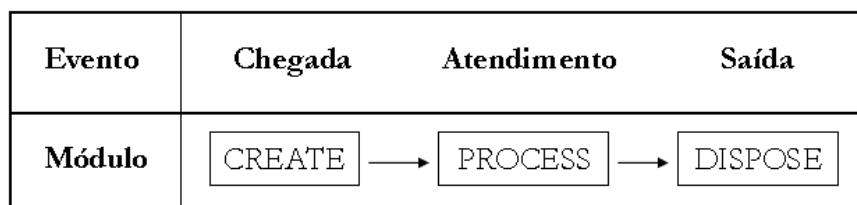
Figura 6: Interface principal do software ARENA®.



Fonte: adaptado de Miranda *et al.* (2006).

A figura abaixo exemplifica um modelo básico para um fluxo de simulação onde a partir de entrada, um processo executa determinada tarefa e, em seguida, é encerrado:

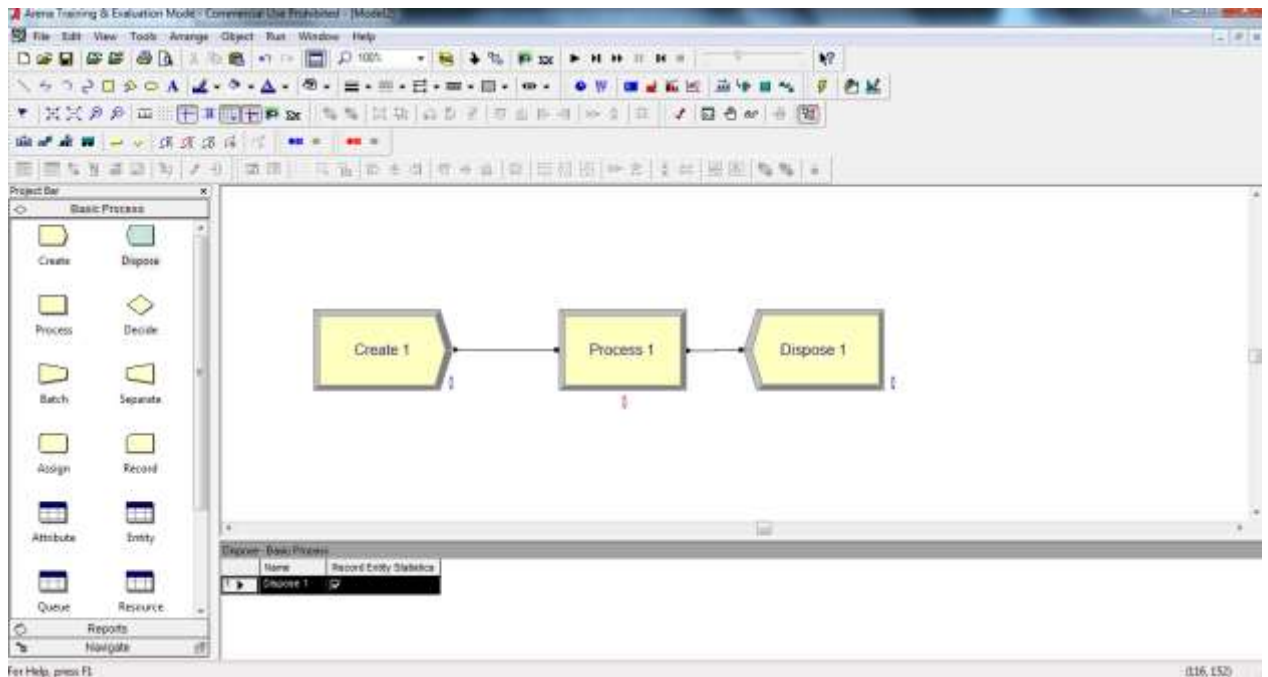
Figura 7: Exemplo de fluxo de processo para o ARENA®.



Fonte: adaptado de Miranda *et al.* (2006).

A figura a seguir exige o mesmo processo, porém traduzido para a linguagem compreendida pelo ARENA®:

Figura 8: Fluxo básico de um processo no ARENA®.



Fonte: adaptado de Miranda *et al.* (2006).

Montado o fluxo com os elementos desejados e suas respectivas ligações, ainda é necessário que sejam configuradas as propriedades da simulação, como limite de execução, quantidade de entidades liberadas a cada rodada de simulação e tempo de processamento para cada entidade. No manual do ARENA® podem ser estudados os demais elementos do software, ferramentas, propriedades e funcionalidades.

Apresentados os conceitos de simulação, bem como as principais características do software ARENA®, a próxima seção abordará as ponderações relativas à EAI.

2.7. Integração de Aplicações Empresariais

O atual cenário de mercado impõe às organizações empresariais a necessidade contínua de prestar serviços de qualidade, com agilidade, garantindo a satisfação do cliente.

Neste contexto, as empresas passaram a utilizar os recursos da TI em seus processos de negócio, na busca pela melhoria da gestão de processos e o consequente desenvolvimento de suas atividades.

De acordo com Souza e Saccol (2003), o uso dos softwares nas empresas nas décadas de 60, 70, 80, até mesmo 90, era restrito e limitava-se apenas a determinados setores.

Gradualmente, a partir de então, os sistemas informatizados foram atingindo as organizações empresariais como um todo. Em virtude disso, não raras vezes, softwares diferentes, com funcionalidades diferentes, passaram a integrar o ambiente de negócio da organização, fruto da expansão do negócio e do desenvolvimento tecnológico.

Mas, segundo Laudon e Laudon (1999, apud Klein, 2015) no contexto empresarial, é importante que a informação esteja disponibilizada de forma unificada, servindo de suporte na análise de dados e tomada de decisões.

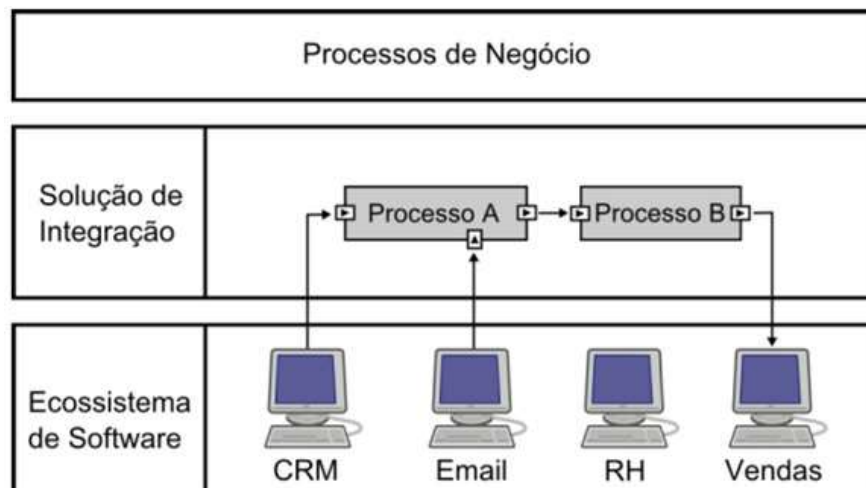
Surge então o conceito de Integração de Aplicações Empresariais - EAI, que visa fazer com que todos os sistemas de informação envolvidos no processo de negócio interajam de forma exógena, ou seja, de maneira transparente e uniforme.

Para Hohpe e Woolf (1999, apud Klein, 2015) uma solução de integração faz com que as aplicações legadas sejam reutilizadas, todas as suas funcionalidades estejam disponíveis e também, que novas funcionalidades sejam criadas.

Em Linthicum (2000, apud Klein, 2015) a integração de aplicações é definida como sendo o compartilhamento irrestrito de dados e processos de negócios.

Já no conceito descrito em Cummins (2002), a EAI refere-se ao compartilhamento irrestrito de dados e processos de negócio, reduzindo assim a complexidade da integração e melhorando a gestão da informação.

Figura 9: Integração de Aplicações Empresariais.



Segundo Frantz *et al.* (2011), caracterizam-se como desafios recorrentes das soluções de integração fazer com que as aplicações mantenham seus dados sincronizados e fazer com que suportem a agregação de novas funcionalidades.

Neste sentido, em seu trabalho Frantz (2012) cita que, uma solução de EAI deve observar alguns aspectos importantes:

- i) os recursos e interfaces das aplicações não devem ser alterados;
- ii) as aplicações devem continuar a funcionar de forma independente;
- iii) a solução deve interferir o menos possível no comportamento normal das aplicações integradas;
- iv) a integração deve ser executada sob demanda, ou seja, deve ser escalável.

Todavia, a EAI não é uma tarefa trivial, uma vez que, a maioria dos softwares integrantes do ecossistema da empresa não foi projetado levando em consideração algum tipo padrão de integração: tecnologias diferentes, interfaces não compatíveis, aplicação com finalidade específica, entre outros (Frantz, 2012).

Neste campo de estudo, uma contribuição importante foi dada pelo trabalho de Hohpe e Woolf (2003, apud Frantz, 2012), onde são apresentados padrões de integração que podem ser usados na construção de uma solução de integração.

Assim sendo, apresentado o objetivo da EAI e sua aplicação, passaremos conceituar de forma resumida a sua estrutura, a qual para melhor didática, está distribuída da seguinte forma: Estilos de Integração, Topologias de Integração, Padrões de Integração e Tecnologias de Integração.

2.7.1. Estilos de Integração

A comunicação entre as aplicações que compõem um ecossistema a ser integrado baseia-se na troca de informação, que pode ser, segundo Hohpe e Woolf (2003, apud Frantz, 2012), de duas formas:

- Compartilhamento de dados: pode se dar através de troca de arquivos (as aplicações precisam acessar o mesmo arquivo) ou mediante um banco de dados (as aplicações realizam processos de leitura e escrita na mesma base de dados). Exemplo: SGBD, FTP.

- Compartilhamento de funcionalidade: a troca de informações entre as aplicações é efetuada através de procedimento remoto ou troca de mensagens. Exemplo: RPC, RMI, Middleware, WebService.

2.7.2. Topologias de Integração

A forma como uma solução será implementada representa a sua topologia. De acordo com Frantz (2012) são consideradas topologias: ponto-a-ponto e Enterprise Service Bus (ESB).

- a) Topologia Ponto-a-Ponto: uma aplicação serve de cliente e servidor ao mesmo tempo. Aqui as aplicações compartilham serviços e dados entre si sem a necessidade de um servidor central.
- b) Topologia Enterprise Service Bus – ESB: é uma arquitetura mais elaborada onde é possível integrar diferentes aplicações através de um barramento de comunicação. As aplicações comunicam-se através deste barramento, de forma independente e sem interferência entre as aplicações.

2.7.3. Padrões de Integração

Por meio de seu trabalho, os autores Hohpe e Woolf (2003, *apud* Frantz, 2012) classificam os padrões de integração em seis categorias:

- i) *Message Constructuion* (Construção de Mensagem): define o formato e conteúdo das mensagens, uma vez que as conceitua como recipientes de dados que trafegam dentro da solução de integração, sendo formadas basicamente, por duas partes: cabeçalho e corpo. No cabeçalho ficam informações na forma de meta dados relativas à mensagem. No corpo propriamente dito, é a mensagem em si que poderá ser modificada, transformada e roteada dentro da solução de integração;
- ii) *Messaging Channels* (Canais de Mensagem): descreve as características de um sistema baseado em mensagem, no qual os canais são parte da infraestrutura da mensagem, usados para suportar o desenvolvimento da

solução de integração. Os canais são usados como recursos de/para, fazendo com que cada mensagem possa ser lida e/ou escrita de forma síncrona entre as aplicações integrantes da solução. Em resumo, um canal é um endereço lógico que deve ser configurado para a solução de integração de acordo com a infraestrutura adotada para as mensagens. Um canal pode ser usado por uma única aplicação ou então, compartilhado por duas ou mais delas. Cada tipo de infraestrutura de mensagem pode requer diferentes tipos de canais e diferentes configurações;

iii) *Message Routing* (Roteamento de Mensagem): permitem a alteração da rota da mensagem dentro da solução de integração, descrevendo os padrões para o envio e recebimento de mensagens. Mas, convém destacar uma característica importante deste tipo de padrão de integração: o conteúdo das mensagens não pode ser alterado, apenas sua rota. Este padrão ainda determina: a forma de rotear a mensagem para um ou mais destinatários, as políticas fixas e dinâmicas de roteamento, como processar individualmente cada elemento da lista de mensagens, como agrupar os resultados do processamento individual das mensagens para poderem ser depois executadas como um todo, como realizar a exclusão de mensagens não desejadas no fluxo da solução de integração.

iv) *Message Transformation* (Transformação de Mensagem): na integração de aplicações, geralmente estas não utilizam o mesmo modelo de dados. Para promover a comunicação entre estas aplicações mediante uma solução de integração então é necessário que as mensagens sejam traduzidas de um formato para o outro a fim de que sejam compreendidas pelas aplicações integrantes da solução. Este padrão define, entre outros: como integrar diferentes tipos de modelos de dados ; como enviar dados de uma aplicação para outra, como simplificar o conteúdo de uma mensagem, como processar mensagens com conteúdos equivalente, porém com formatação diferente, como minimizar as dependências ao integrar aplicações com diferentes formatos de dados, como criar formatos de dados compreensíveis indiferentemente da aplicação.

- v) *Messaging Endpoint*: como na maioria dos casos as aplicações de um ecossistema de software a ser integrado não foram concebidas levando em consideração esta possibilidade, a fim de que as aplicações possam trocar informações entre si por meio do envio e recebimento de mensagens, este padrão prevê a criação de mensagens que contenham informações capazes de realizar a comunicação entre as aplicações existentes e a solução de integração. Para isso, neste padrão estão descritas, entre outras, formas de: como criar uma interface para a solução de integração capaz de suportar a comunicação entre as aplicações.
- vi) *System Management*: descreve os padrões para o gerenciamento de uma solução de integração, documentando como detectar falhas em blocos de mensagens, como inspecionar uma mensagem sem afetar seu processamento, como rastrear uma mensagem, entre outros. Nota-se a importância deste padrão, uma vez que, uma solução de integração pode processar milhares de mensagens, processo complexo no qual podem ocorrer gargalos de desempenho tanto na solução de integração quanto nas aplicações dela integrantes.

2.7.4. Soluções de Integração

As soluções para EAI são apresentadas na forma de tecnologias ou ferramentas de integração de aplicações. Essas ferramentas vêm ao encontro da demanda das organizações empresarias por soluções de integração, em decorrência da evolução nos sistemas de informação, bem como, da busca pela melhoria contínua nos processos do negócio a fim de se garantir boa colocação no mercado.

A proposta das soluções de integração, segundo Frantz (2012), deve importar-se principalmente com dois problemas centrais encontrados nos ecossistemas de software: *garantir a sincronia das informações que fluem pela solução de integração e suportar a incorporação novas funcionalidades além das já existentes.*

Levando em consideração suas características técnicas, podemos classificar as soluções de integração em duas categorias, a saber: *Linguagens de Propósito Geral* e *Linguagens de Propósito Específico*.

2.7.4.1. Linguagens de Propósito Geral

A solução proposta pelas Linguagens de Propósito Geral pode ser aplicada para a construção de uma proposta de integração de aplicações, porém não foi criada com esse fim específico. Dentre estas citamos:

- CORBA (Common Object Request Broker Architecture), SOAP (Simple Object Access Protocol): middlewares utilizados para a integração de sistemas distribuídos;
- UML (Unified Modeling Language), EPC (Event-driven Process Chain), BPMN ((Business Process Model and Notation): possibilitam a modelagem do processo do negócio na forma de notação, porém, não possuem suporte para a análise formal de dados, priorizando a perspectiva de controle de fluxo e gerenciamento de recursos, sem a possibilidade de análise de resultados;
- REDES DE PETRI e CADEIAS DE MARKOV: através da implementação de processos matemáticos, realizam a conversão de modelos de processos de negócio em formalismos, possibilitando a análise quantitativa dos modelos.

2.7.4.2. Linguagens de Propósito Específico

As soluções de integração baseadas em Linguagens de Propósito Específico dizem respeito àquelas ferramentas e linguagens desenvolvidas com enfoque na promoção de soluções de integração de aplicações.

Conforme apresentado em Frantz (2012) as principais tecnologias ou ferramentas para a implementação de soluções de integração de aplicação com fins específicos são:

- Camel: é um framework para integração de sistemas baseado em regras implementadas através dos Padrões de Integração, desenvolvido com suporte para Java, Scala e XML para a realização da tarefa de integração. É uma ferramenta de código aberto sob a licença da Apache Software Foundation. Camel é caracterizado como uma DSL que utiliza a topologia ESB. A sua arquitetura implementa os seguintes conceitos: exchange (channels), Endpoint, Processor, Route (Routing);
- Spring: é uma ferramenta que estende o Spring Framework para suportar os padrões de integração de aplicações empresariais, tendo preocupação com a produtividade do desenvolvedor, tornando mais fácil as tarefas de construir, testar e manter soluções de integração empresarial. Ela apresenta uma interface gráfica baseada em Eclipse. Sua arquitetura é baseada em *Pipes-and-Filters*, o que significa que mensagens fluem através de várias unidades de processamento independentes (filtros), e comunicam-se através de canais (tubos). As mensagens são implementadas com um bloco de construção, os filtros são implementados com *endpoints* e os tubos são implementados com canais de mensagens (*channels*). Também se apresenta na forma de código aberto;
- Mule: é uma ferramenta de código aberto baseada em Java, cuja arquitetura é inspirada pelo conceito o de ESB. Também implementa os padrões de integração. O ambiente gráfico é baseado em Eclipse. Sua arquitetura compreende os conceitos de mensagem, *endpoint*, processador e fluxo;
- Guaraná: é uma tecnologia criada recentemente, que permite modelar soluções de integração utilizando os padrões de integração, utilizando alto nível de abstração, oferecendo aos usuários um mecanismo de tolerância a falhas baseado em regras.

Para fins acadêmicos, a tecnologia Guaraná DSL será mais bem detalhada em uma seção criada especificadamente para este fim.

2.8. A Tecnologia Guaraná

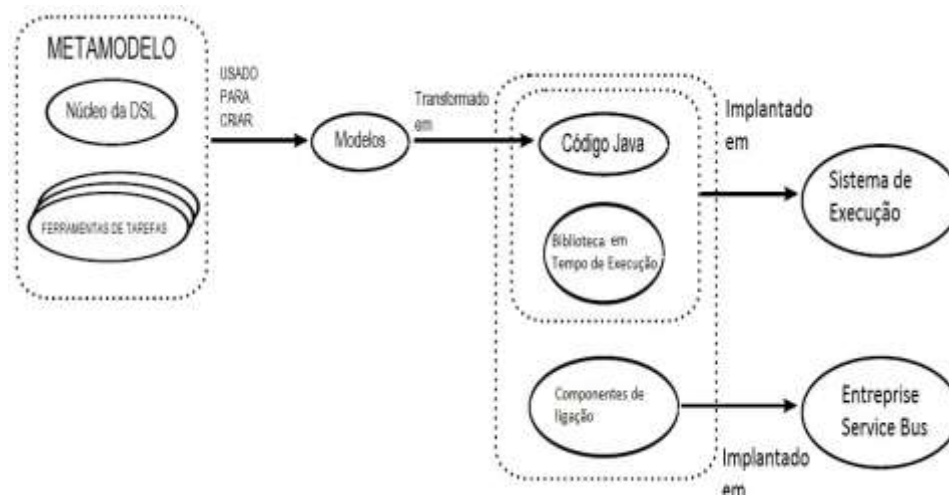
A tecnologia Guaraná DSL oferece suporte para construção de soluções de integração de uma forma mais acessível, uma vez que apresenta um alto nível de abstração.

Ela possui uma linguagem gráfica domínio específico baseada nos padrões de integração propostos Hohpe e Woolf (2003, *apud* Frantz, 2012), a qual possibilita que os engenheiros de software tenham uma visão geral do conjunto de processos da solução de integração.

A facilidade de uso, o ambiente gráfico para a construção das soluções e a tolerância a falhas baseada em regras, fazem com que a tecnologia Guaraná seja uma interessante opção no campo da Integração de Aplicações.

Abaixo, segue figura demonstrando o processo de construção da ferramenta Guaraná: o metamodelo é dividido em duas partes: uma principal (Núcleo da DSL), que suporta um subconjunto de conceitos que são assumidos como sendo úteis para um conjunto grande de soluções de integração; e uma série de caixas de ferramentas de tarefas, que suportam subconjuntos de tarefas com características específicas para um determinado domínio de integração (Frantz, 2012).

Figura 10: Concepção do Guaraná.



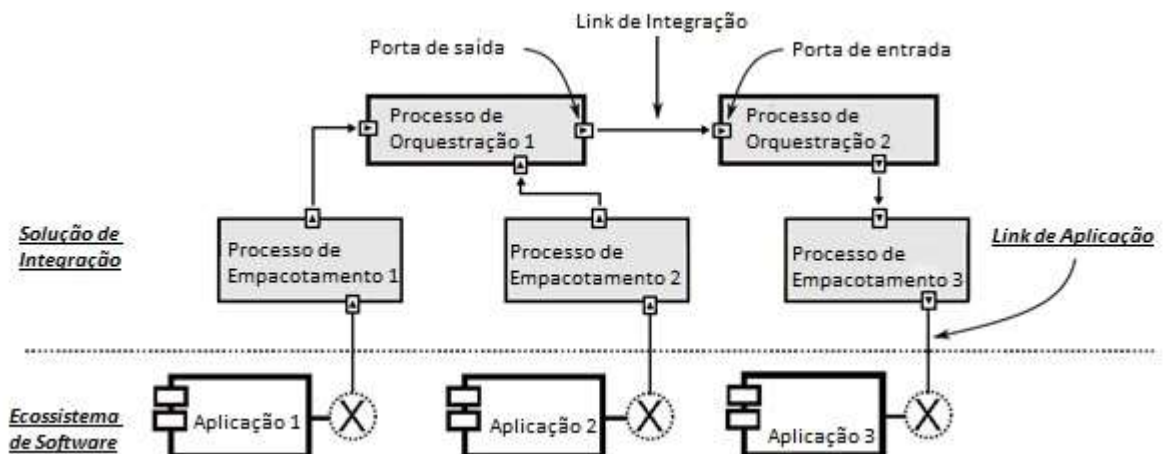
Fonte: adaptado de Frantz *et al.* (2011).

Conforme a imagem acima, observa-se que a tecnologia Guaraná também possibilita a transformação de um modelo em código Java, o qual invoca uma

biblioteca em tempo de execução. Esta biblioteca fornece as classes necessárias para implementar os conceitos suportados pelo metamodelo. Neste processo ainda são necessários alguns componentes de ligação que devem ser configurados e implantados para executar em um ESB (Enterprise Service Bus) independentemente. Os processos que compõe a solução devem usar estes componentes de ligação para interagir com as aplicações que estão sendo integradas ou com outros processos (Frantz, 2012).

Uma solução de integração modelada com o Guaraná é composta por processos de empacotamento, responsáveis por interagir com as aplicações e processos de orquestração, responsáveis pela gestão do fluxo de mensagens. Os processos usam portas para realizar a comunicação com outros processos, ou então, usam links para promover a comunicação com as aplicações.

Figura 11: Modelo de uma solução de integração.



Fonte: adaptado de Frantz *et al.* (2011).

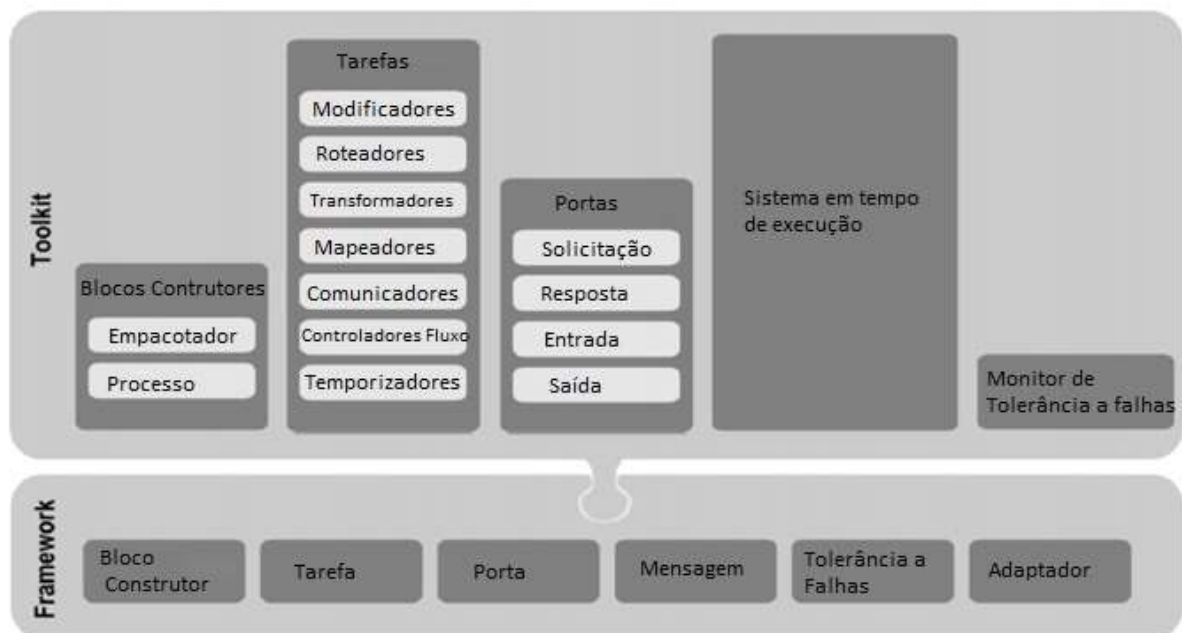
Para fornecer suporte à implementação de soluções de integração, a tecnologia Guaraná possui um Kit de Desenvolvimento de Software (SDK) chamado Guaraná SDK que é a implementação Java do Guaraná DSL.

O Guaraná SDK está organizado em duas camadas: a primeira camada, denominada *framework*, implementa todos os conceitos básicos da linguagem de domínio específico. Já a segunda camada, denominada *toolkit*, proporciona uma implementação concreta do *framework*, assim como um motor de execução multi-thread assíncrono para executar as soluções de integração. Frantz *et al.* (2014).

Em virtude da natureza distribuída das aplicações, as soluções de integração estão vulneráveis a erros que impactam diretamente no seu comportamento e resultados. Os erros acontecem em função de falhas, que podem ser permanentes (ex: defeitos no software) ou transientes (ex: indisponibilidade de um recurso necessário para a solução). Erros, quando não tratados resultam em falhas que são percebidas pelos usuários finais. O Guaraná FT é responsável pelo monitoramento e detecção de erros que possam ocorrer nas portas de comunicação presentes nos processos que compõe as soluções de integração. Frantz *et al.* (2014).

Outra propriedade encontrada no kit de ferramentas do Guaraná SDK, é o sistema em tempo de execução (*Runtime System*). Esta ferramenta utiliza um modelo de execução com base em regras, características esta que difere o Guaraná das outras tecnologias que utilizam um modelo de execução baseado em processos (Frantz, 2012).

Figura 12: Arquitetura do Guaraná SDK.



Fonte: adaptado de Frantz e Corchuelo (2012).

Em uma solução de integração, um processo pode ser visto como um processador de mensagens. As tarefas são extremamente dependentes de contexto, o que faz com que seja de pouco interesse pensar em uma coleção de propósito geral para tarefas. Os processos utilizam portas para comunicarem-se uns com os outros ou com as aplicações envolvidas em uma solução de integração, sendo que uma porta (*port*) é responsável por abstrair os detalhes necessários para interagir

com um comunicador (*communicator*), o qual, por sua vez, abstrai os detalhes necessários para interagir com um aplicativo ou com um processo. As portas (*ports*) podem ser ou portas de entrada ou de saída, dependendo da sua concepção. Já a comunicação entre uma tarefa de uma porta com uma tarefa de um processo ocorre por meio de um *slot*, porém com características um pouco diferentes e por isto denominado de *interslots*.

A Figura 13 mostra a modelagem de uma solução de integração para um sistema de ligações telefônicas utilizando o Guaraná: pelo sistema de ligações telefônicas as chamadas são filtradas, as que possuem custos são encaminhadas para débito na folha de pagamento do funcionário (*Payroll System - PS*), bem como encaminhada uma notificação ao funcionário através de email (*Email Server - MS*). Frantz *et al.* (2011),

Segundo os autores, esta solução é constituída por três processos de empacotamento (*wrapping*) e um processo de orquestração (*orchestration*).

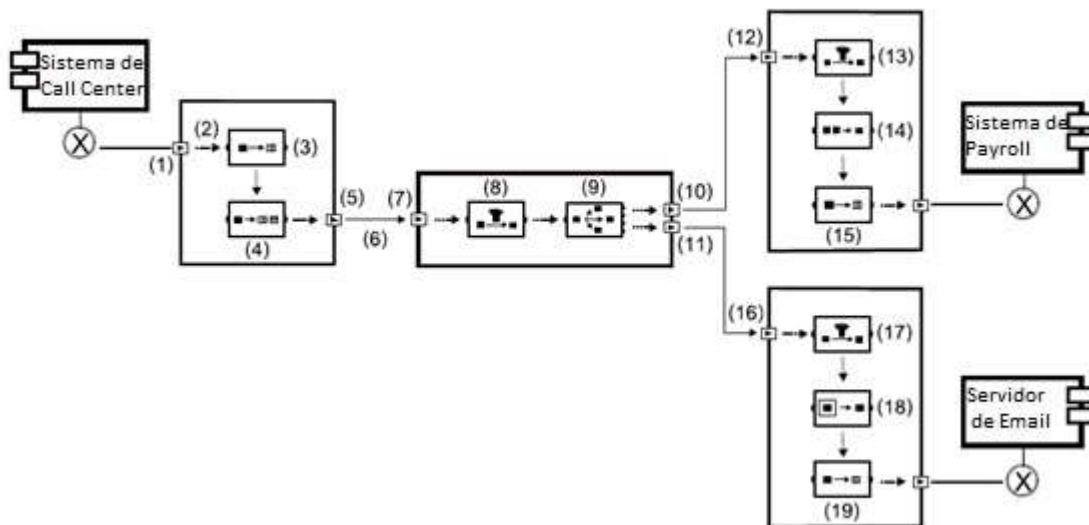
O fluxo de integração se inicia na porta de entrada 1 (*entry port*), a qual realiza a leitura de código CCS a cada período de tempo, buscando por um novo registro de chamada. Esta porta produz mensagens compostas por informações sobre a chamada telefônica e as transfere para o *slot* 2. A tarefa 3 é um tradutor (*translator*) que transforma as mensagens de entrada em mensagens de saída.

A tarefa 4 é um divisor (*splitter*) que separa as mensagens de entrada em partes menores, cada uma destas correspondente a somente uma chamada. A porta de saída 5 (*exit port*) escreve estas mensagens no link de integração 6 (*integration link*), assim cada processo de orquestração (*orchestration process*) pode ler estas mensagens através da porta de entrada 7 (*entry port*). O objetivo desse processo é filtrar 8 (*filter*) as ligações gratuitas, mantendo no fluxo apenas aquelas que tem custo.

A tarefa 9 é um replicador (*replicator*) que copia as mensagens recebidas para as portas de saída 10 e 11. As mensagens enviadas para a porta 10 são transferidas para a porta 12 que é uma parte do processo de empacotamento (*wrapping*) da aplicação PS. Primeiramente a mensagem é enviada para o filtro 12 (*filter*) para garantir que apenas mensagem com custos permaneçam no fluxo. Na tarefa 14 (*idempotent transfer*) é impedido que as mensagens duplicadas sejam enviadas para a aplicação.

A tarefa 15 é um tradutor (*translator*) que transforma a mensagem recebida em uma nova mensagem para a aplicação PS. As mensagens enviadas pela porta 11 são transferidas para a porta 16, que é parte do processo de empacotamento (*wrapping*) da aplicação MS. O filtro 17 (*filter*) garante que permaneçam no fluxo apenas as mensagens que tenham um email de funcionário, evitando assim que a aplicação MS receba mensagens que não possam ser notificadas ao usuário. Esta notificação ao usuário resume informações úteis de todas as chamadas, como por exemplo: número de destino, custo, duração, cidade de destino, estado. No entanto, essa notificação não possui quaisquer outros detalhes técnicos, pois são removidos na tarefa 18, que reduz o tamanho da mensagem (*slimmer*). A tarefa (19) é um tradutor que transforma as mensagens recebidas em mensagens a serem enviadas, no formato compreendido pela aplicação MS.

Figura 13: Solução de integração com o Guaraná.



Fonte: adaptado de Frantz *et al.* (2011).

Capítulo 3: Estudo de Caso

Este capítulo descreve o estudo de caso realizado neste trabalho baseado em Frantz (2012). Aqui é apresentada a simulação de uma solução de integração proposta para o problema abstrato denominado Café.

3.1 *Descrição do problema*

O estudo de caso a ser tratado neste trabalho denomina-se Café, e sua escolha leva em consideração o fato de que, do ponto de vista técnico, ele tornou-se um padrão para comparação de propostas de integração de aplicações. Ele consiste na especificação do fluxo de atendimento dos pedidos de clientes em um estabelecimento comercial do tipo Café. Frantz (2012).

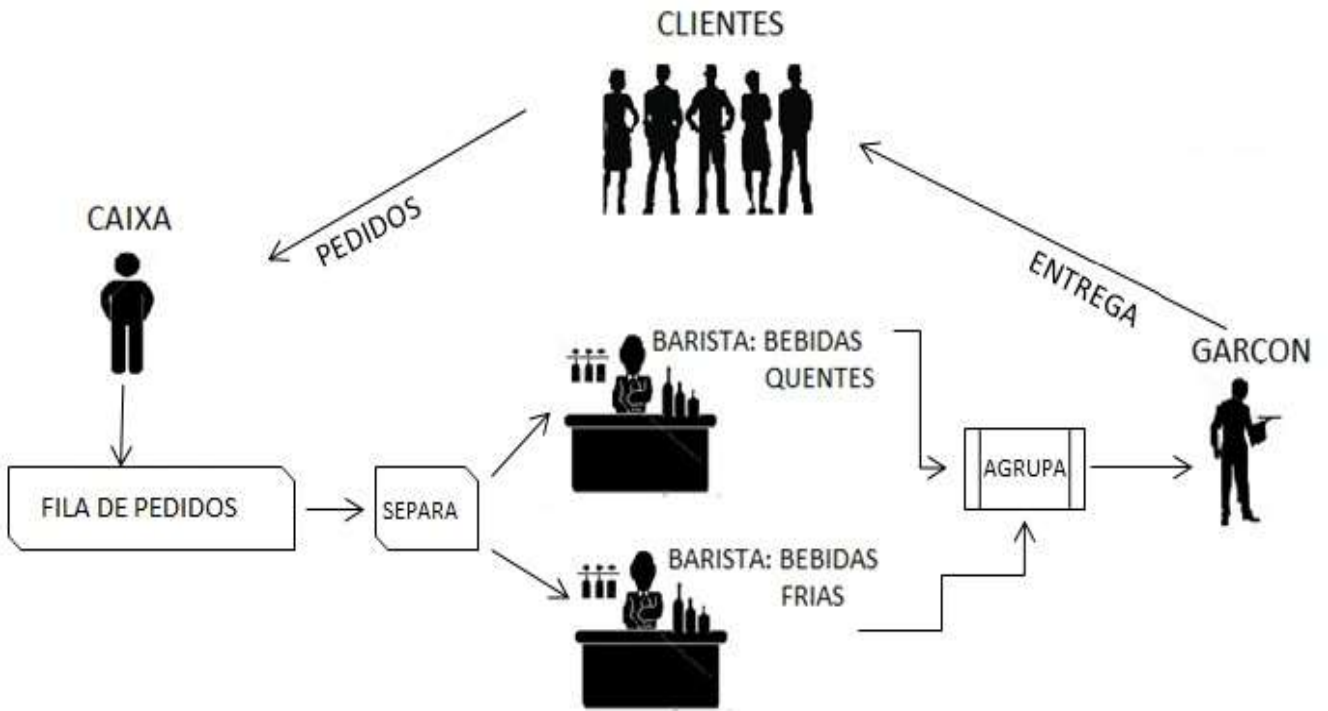
Basicamente, o funcionamento do Café pode ser assim descrito: primeiramente, ao entrar na loja o cliente dirige-se ao caixa para fazer seu pedido. Ao cadastrar o pedido do cliente, automaticamente o caixa coloca-o em uma fila de atendimento. O pedido do cliente pode ser relativo à bebida quente e/ou fria. Cada tipo de bebida é preparada por uma barista específico. O Barista recebe a ordem do preparo da bebida, a qual é integrante do pedido principal, bem como, atrelada a um número de bandeja.

Assim que a bebida solicitada é preparada, ela é colocada pelos Baristas na bandeja correspondente e então, poderá ser colocada na fila para ser entregue pelo garçom ao cliente. Os Baristas não tem visão de todos os itens do pedido do cliente, apenas das bebidas individuais da fila a serem preparadas.

Desta forma, os caixas podem continuar a fazer pedidos, mesmo que os Baristas estejam ocupados. Os caixas são os responsáveis por organizar a fila de pedidos a serem preparados e os Baristas, encarregados da fila de pedidos a serem entregues aos clientes.

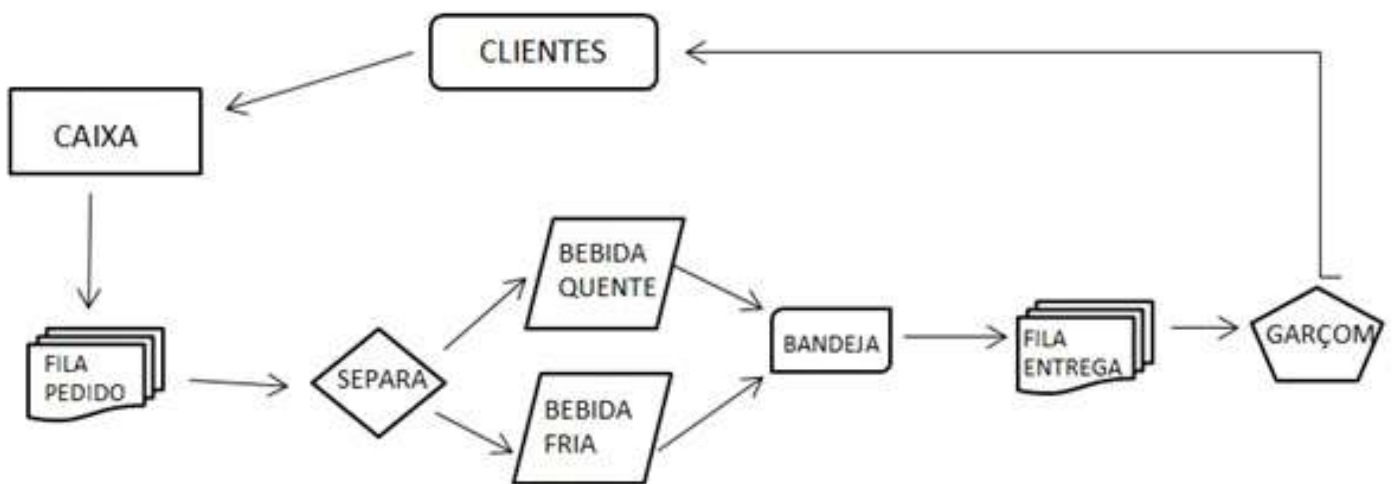
O caixa possui o controle do consumo do cliente, realizado através do pedido inicial, e conseqüentemente pode realizar o encerramento da conta. O objetivo é criar uma solução de integração na qual os pedidos são realizados pelos clientes, colocados pelo caixa em uma fila, preparados pelos Baristas específicos e depois, mediante uma fila de entrega, possam ser servidos aos clientes pelos garçons.

Figura 14: Modelo abstrato: Estudo de caso Café.



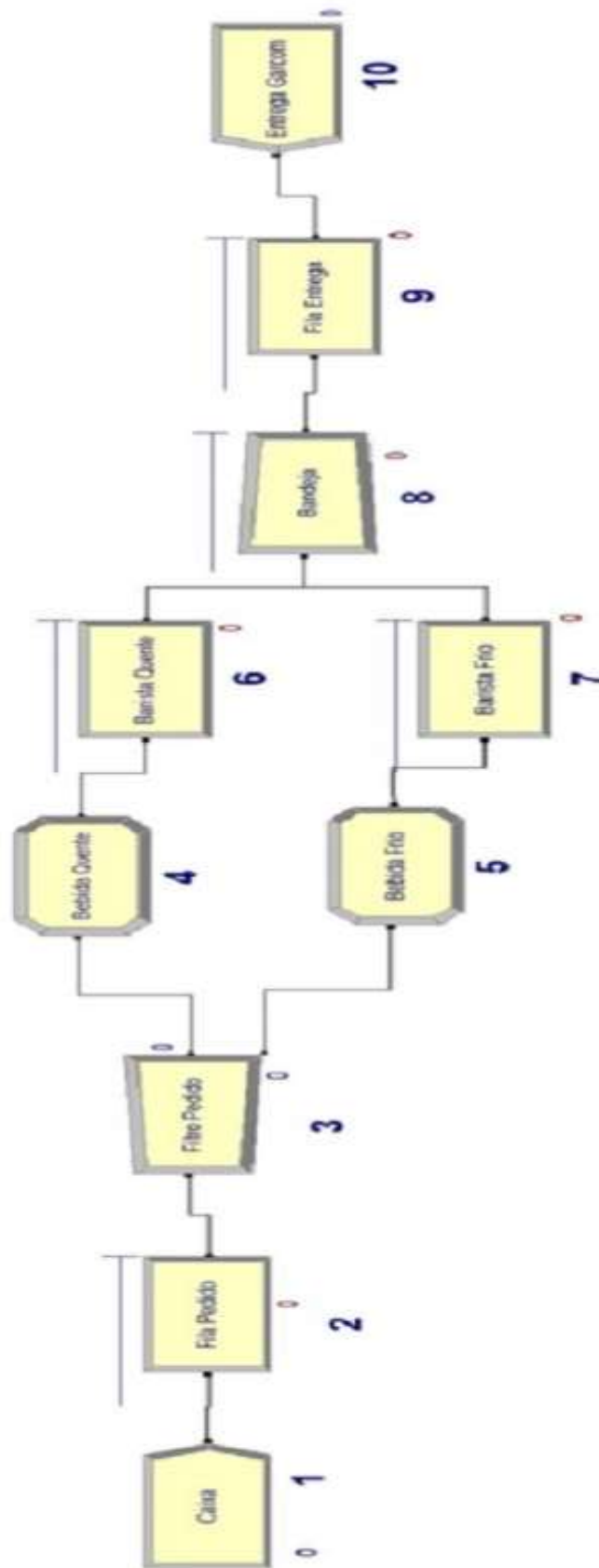
Fonte: autoria própria.

Figura 15: Modelo conceitual: Estudo de caso Café - Fluxograma.



Fonte: autoria própria.

Figura 16: Modelo computacional: Estudo de caso Café - ARENA®.



Fonte: autoria própria.

3.2 A proposta de solução

Primeiramente é importante destacar que a proposta do presente trabalho encontra-se de acordo com a área de aplicação da simulação conforme citado em Prado (1999), uma vez que a simulação computacional permite que sejam analisadas alterações em sistemas e plantas físicas de negócio, sem a necessidade de implanta-las no mundo real, reduzindo custos e garantindo eficiência do processo.

Desta forma, a representação do processo de simulação deu-se de acordo com o proposto por Chwif e Medina (2006): primeiramente foi criado o modelo abstrato conforme a Figura 14, depois desenvolvido o modelo conceitual mostrado na Figura 15 e, por fim, o modelo computacional de acordo com a Figura 16.

O modelo computacional adotado foi do tipo de simulação, classificado como dinâmico, determinístico e contínuo, onde foram observados os princípios essenciais do processo de simulação de acordo com autores Banks e Carson (1984, apud Sabbadini e Gonçalves, 2005).

Ainda convém destacar que a metodologia adotada para o trabalho seguiu os padrões determinados pelos principais autores desta área: Shannon (1998), Banks (1999), Banks e Carson (1984) e Harrel *et al.* (2002) conforme citado em Sabbadini e Gonçalves (2005).

Partindo-se agora para a apresentação da solução, elencamos os seguintes pontos:

- 1) A solução de integração foi desenvolvida com o software ARENA® 14.7 na versão Student, executada em computador do tipo notebook, equipado com o processador da família Intel Core I5 de 2.3 Ghz, com 4GB de memória RAM e sistema operacional Windows 7 de 64 bits.
- 2) O fluxo do modelo de simulação no ARENA® inicia-se em um bloco *create* (1) representando o caixa, os pedidos feitos no caixa são enviados para uma fila de atendimento representada por um bloco *process* (2). A fila de pedidos encaminha os pedidos para preparo, porém como cada pedido é composto por duas bebidas, eles são encaminhados para destinos distintos (Barista de Bebida Quente e Barista de Bebida Fria) passando

por um bloco *separate* (3). Necessariamente as entidades devem passar pelos blocos construtores de processos chamados *Assing* relativos à Bebida Quente (4) e Bebida Fria (5), que transferem a informação para os blocos *process* do Barista Quente (6) e do Barista Frio (7). Após o preparo das bebidas, as duas são agrupadas pelo bloco *Batch* (8) que representa a bandeja, e então, são encaminhadas para a fila de entrega dos pedidos (9) pelo garçom (10).

- 3) A simulação foi executada de acordo com os seguintes critérios: levou-se em consideração que, a cada 1 segundo pedidos eram encaminhados para a fila de pedidos. A fila de pedido despacha estes pedidos para o filtro de pedidos a cada 0.01 segundo. Os baristas levam 0.01 segundo para produzir uma bebida. A bandeja aguarda até que as duas bebidas estejam prontas para enviar para a fila de entrega, onde, a cada 0.02 segundo o garçom entrega o pedido aos clientes.
- 4) As rodadas de simulação foram realizadas levando-se em consideração alguns pontos:
 - a versão Student do ARENA® impõe como limitação um número máximo de 150 entidades fluindo pelo processo;
 - a propriedade do bloco *create* (1) chamada de *Entities per Arrival* determina quantas entidades serão enviadas para o fluxo a cada intervalo de tempo;
 - a propriedade *Max Arrivals* (máximo de envios) do mesmo bloco *create* (1) multiplica o número nela informado pelo valor existente na propriedade *Entities per Arrival* (entidades por envio).
 - A simulação do fluxo proposto foi realizada da seguinte forma: enviamos pedidos (mensagens) a cada segundo até atingirmos o limite de 100.000 (cem mil) a cada ciclo de simulação. Estes ciclos de envio correspondem ao número de mensagens enviadas a cada segundo até atingirmos o total de 100.000 mensagens, por exemplo: no primeiro cenário, a cada segundo enviamos 5 mensagens; no segundo cenário, enviamos 10 mensagens a cada segundo e assim sucessivamente, em intervalos de 5 em 5 mensagens até o total de 50 mensagens por segundo.

Os cenários simulados conforme descrição acima foram os seguintes:

- a) Cenário I – 1 (uma) mensagem enviada a cada segundo, até o total de 100.000 mensagens;
- b) Cenário II – 5 (cinco) mensagens enviadas a cada segundo, até o total de 100.000 mensagens;
- c) Cenário III – 10 (dez) mensagens enviadas a cada segundo, até o total de 100.000 mensagens;
- d) Cenário IV – 15 (quinze) mensagens enviadas a cada segundo, até o total de 100.000 mensagens;
- e) Cenário V – 20 (vinte) mensagens enviadas a cada segundo, até o total de 100.000 mensagens;
- f) Cenário VI – 25 (vinte e cinco) mensagens enviadas a cada segundo, até o total de 100.000 mensagens;
- g) Cenário VII – 30 (trinta) mensagens enviadas a cada segundo, até o total de 100.000 mensagens;
- h) Cenário VIII – 35 (trinta e cinco) mensagens enviadas a cada segundo, até o total de 100.000 mensagens;
- i) Cenário IX – 40 (quarenta) mensagens enviadas a cada segundo, até o total de 100.000 mensagens;
- j) Cenário X – 45 (quarenta e cinco) mensagens enviadas a cada segundo, até o total de 100.000 mensagens;
- k) Cenário XI – 50 (cinquenta) mensagens enviadas a cada segundo, até o total de 100.000 mensagens;

Capítulo 4: Resultados

Este capítulo apresenta os resultados obtidos mediante a execução da simulação da proposta de integração para o estudo de caso Café.

4.1 Apresentação dos Resultados

Na solução de integração construída com a ferramenta Guaraná para o estudo de Café, uma das variáveis analisadas por Frantz (2012) foi chamada de Mensagens Pendentes (*Pending Messages*), ou seja, o número de mensagens ainda não processadas dentro do fluxo quando o processamento de mensagens já conclui o envio destas.

Realizamos a mensuração desta mesma variável através da implementação da solução de integração para o problema abstrato Café com o software ARENA®, buscando alcançar o máximo de entidades no fluxo permitido pela versão Student, ou seja, 150 entidades.

De acordo com as rodadas de simulação realizadas nos parâmetros descritos na seção anterior, para a proposta de solução relativa ao problema Café, a versão student do ARENA® comportou um fluxo máximo de envio de mensagens por segundo de 50 entidades.

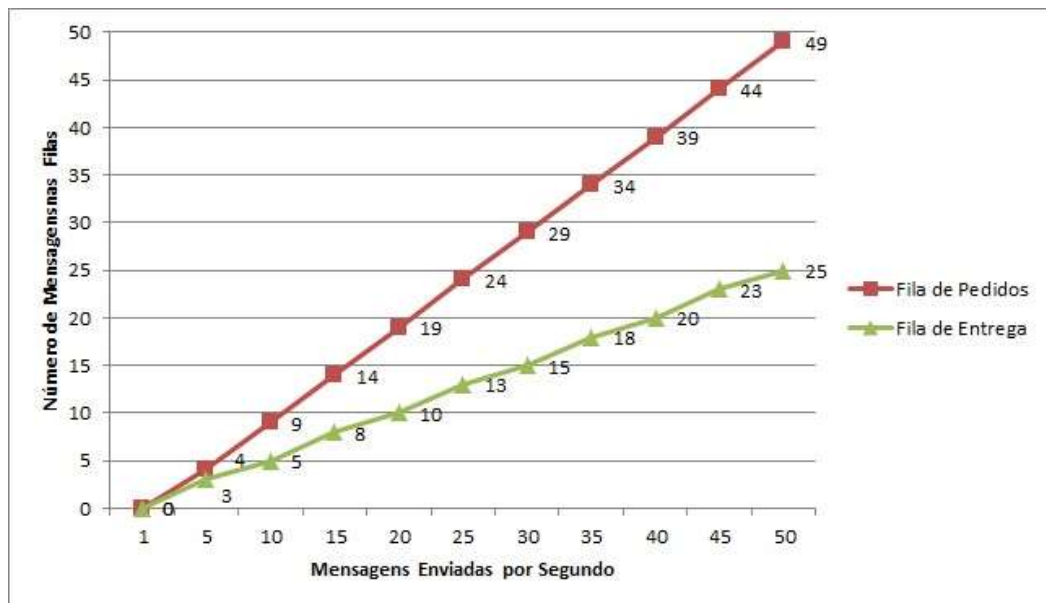
Já o número de mensagens pendentes pôde ser verificado por meio da propriedade *Number Waiting* do parâmetro *Queue* disponível no relatório do ARENA®. Esta propriedade informa o número de mensagens que permaneceram aguardando processamento na fila determinada do fluxo, ou seja, o tamanho da fila.

Com o auxílio da figura a seguir, podemos verificar visualmente que, a fila de Pedidos passou a acumular mensagens a partir do Cenário II, onde começaram a ser enviadas 5 mensagens por segundo para o fluxo. Quando foram enviadas 40 mensagens por segundo, nesta Fila de Pedidos já haviam 39 mensagens aguardando na fila e, quando foi testado o último cenário, ou seja, simulado o envio de 50 mensagens por segundo, esta fila apresentou um acúmulo de 49 mensagens.

Em relação a Fila de Entrega, observamos que o acúmulo de mensagens se iniciou também no segundo cenário quando foram enviadas 5 mensagens por segundo, aqui o número de mensagens aguardando processo foi de 3. Já quando

foram enviadas 25 mensagens por segundo no Cenário VI, havia 13 entidades aguardando na fila. Por fim no Cenário XI, quando foram enviadas 50 mensagens por segundo, 25 ficaram aguardando processamento ao fim da execução do processo de simulação.

Figura 17: Cenários de Envio de Mensagens por segundo.

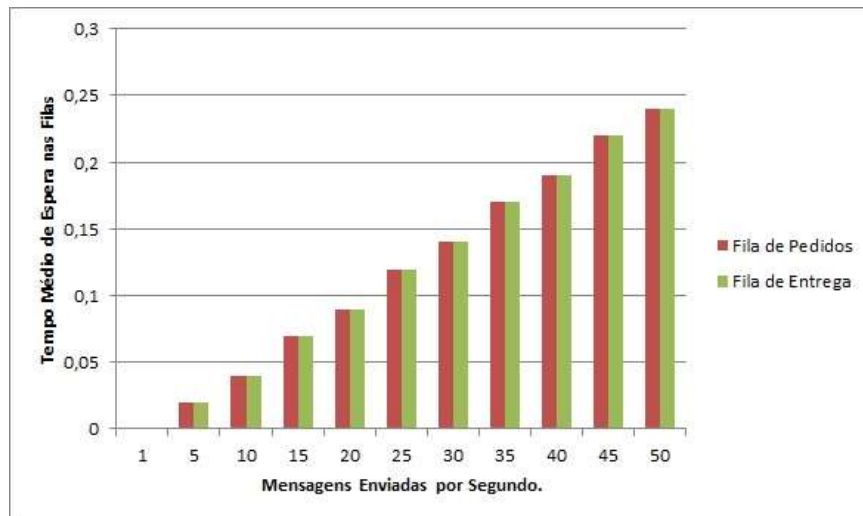


Fonte: autoria própria

Ainda outras duas variáveis nos chamaram a atenção quando da análise dos dados apresentados no relatório de execução do software ARENA®, a saber:

- Tempo médio de espera da entidade na fila: este valor é obtido através da propriedade *Waiting Time* do parâmetro *Queue*. Ele diz respeito ao tempo que cada entidade ficou esperando na fila específica; Com a análise desta variável pôde ser verificado que o tempo médio de espera das entidades tanto na Fila de Pedidos quanto na Fila de Entregas seguiu um comportamento exponencial e regular: no Cenário VII, quando são enviadas 30 mensagens por segundo para o fluxo, o tempo médio de espera nas filas foi de 0,14 segundos. Já quando foram enviadas 45 mensagens por segundo, este tempo passou para 0,22 segundos.

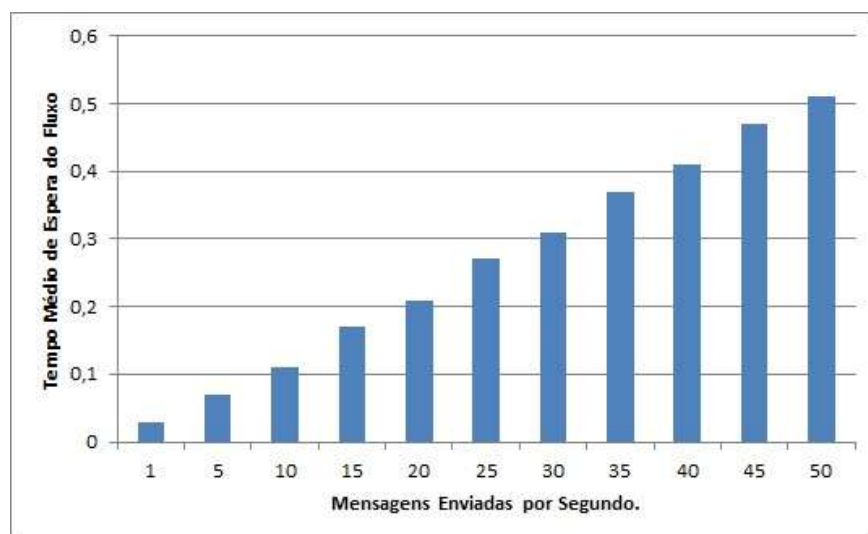
Figura 18: Tempo médio de espera da entidade na fila.



Fonte: autoria própria

- Tempo médio de espera total no fluxo: é informado na propriedade *Total Time Per Entity* do parâmetro *Process* e exibe o Tempo Total que a entidade ficou em processamento no fluxo inteiro, ou seja, o tempo total acumulado que uma entidade levou para percorrer o fluxo inteiro. Aqui também observamos um comportamento exponencial dos dados, onde no Cenário I tivemos um tempo total de espera de 0,03 segundos, depois no Cenário IV este tempo passou para 0,17 segundos e no Cenário XI já pode ser observado um tempo total de espera de 0,51 segundos.

Figura 19: Tempo médio de espera total no fluxo.



Fonte: autoria própria

Capítulo 5: Considerações Finais

5.1 Conclusão

O campo da Integração de Aplicações de Empresariais vem crescendo a cada dia, assim como, as tecnologias das ferramentas usadas nesta área. Como uma das principais vantagens desta técnica está à promoção do compartilhamento de informações dentro da organização garantindo melhor gestão da informação na busca pela competitividade de mercado.

Já o uso da Simulação Computacional torna possível que propostas de integração de aplicações sejam estudadas sem a necessidade de sua efetiva implementação. Assim, como principais vantagens da Simulação Computacional estão à economia de recursos e a verificação da viabilidade técnica do processo.

Este trabalho evidencia a parceria promissora das duas áreas, onde através de uma ferramenta de software, foi simulada uma solução de integração para um problema abstrato, possibilitando assim a análise e mensuração de seus resultados.

Ficou clara a necessidade de implementação conceitual tanto da proposta de integração, quanto do processo de simulação computacional, visando garantir a eficiência do projeto.

Cabe destacar a limitação da versão student do software ARENA®, a qual se refere não ao tamanho do projeto, mas sim, da quantidade de entidades que fluem através do processo simulado. É suportado um total de 150 entidades fluindo pelo processo, nada mais além disso.

Por fim, entendemos que o objetivo proposto foi alcançado com êxito, onde aplicamos de maneira efetiva e conjunta os conceitos e técnicas das áreas de Simulação Computacional e Integração de Aplicações Empresariais.

5.2 Contribuições

Como contribuições científicas decorrentes da realização do presente trabalho, podem ser elencadas como principais:

- 1) Construção de uma proposta de integração de um modelo de simulação com o software ARENA®, equivalente ao modelo conceitual elaborado com a ferramenta Guaraná DSL;
- 2) Simulação da solução desenvolvida, demonstrando ser possível a implementação com o ARENA® do estudo de caso Café;
- 3) Descoberta de alguns fatores limitantes para uso do software Arena®, como por exemplo, o número de mensagens no fluxo limitado a 150.

5.3 Trabalhos Futuros

Com o objetivo de dar continuidade ao estudo realizado no presente trabalho, pretendemos realizar a comparação de resultados da simulação de uma solução de integração modelada com o ARENA® e os resultados dos testes obtidos com a implementação da mesma solução modelada com o uso da ferramenta Guaraná DSL.

A proposta futura é traçar um comparativo de resultados do software ARENA® e da ferramenta Guaraná DSL, referente a implementação de uma solução de integração para o problema abstrato Café.

Porém, conforme comprovado através deste estudo, será necessária a utilização da versão comercial do ARENA®, uma vez que a versão student apresenta limitação de tamanho de fluxo de projeto.

Referências Bibliográficas

ARENA. Disponível em: <http://www.paragon.com.br/software/arena/>. Acesso em: 05/11/2014.

ARENA SIMULATION. Disponível em: <https://www.arenasimulation.com/>. Acesso em: 05/11/2014.

CHWIF, L.; Medina, A.C. Modelagem e Simulação de Eventos Discretos: Teoria e Aplicações. São Paulo, Ed. dos Autores, 2006.

CUMMINS, Fred A. Integração de Sistemas – Arquiteturas para Integração de Sistemas e Aplicações Corporativas. Editora Campus Ltda – Rio de Janeiro, 2002.

FRANTZ, Rafael Z.; Sawicki, Sandro; Roos-Frantz, Fabricia; Corchuelo, Rafael; Basto-Fernandes, Vitor; Hernández, Inma. Desafios para a Implantação de Soluções de Integração de Aplicações Empresariais em Provedores de Computação em Nuvem. Salão do Conhecimento, Universidade Regional do Noroeste do Estado do Rio Grande do Sul, 2014. Disponível em:

<https://www.revistas.unijui.edu.br/index.php/salaconhecimento/article/view/4088>.

Acesso em: 05/11/2014.

FRANTZ, Rafael Z. Enterprise Application Integration: An Easy-to-Maintain Model-Driven Engineering Approach. Doctoral Dissertation. University of Seville. 2012. Disponível em: <http://fondosdigitales.us.es/tesis/tesis/2065/enterprise-application-integration-easy-maintain-model-driven-engineering-approach/>.

Acesso em: 05/11/2014.

FRANTZ, Rafael Z.; Reina-Quintero, A. M. and Corchuelo, R. A Domain-Specific language to design enterprise application integration solutions. International Journal of Cooperative Information Systems, 20(2):143–176, 2011. Disponível em: <http://www.gca.unijui.edu.br/members/rzfrantz/publications/ijcis-2011.pdf>. Acesso em: 15/11/2014.

FRANTZ, R.Z. e Corchuelo, R. A software development kit to implement integration solutions. Em Proceedings of the 27th Annual ACM Symposium on Applied Computing, páginas 1647–1652. ACM, 2012. Disponível em: <http://www.gca.unijui.edu.br/members/rzfrantz/publications/sac-2012.pdf>. Acesso em: 13/04/2014.

FREITAS FILHO, P.J. Introdução à Modelagem e Simulação de Sistemas. 2ª edição, Editora Visual Books, 2008.

GAVIRA, M.O.; Taveres, C.G.; Roveri, E.; Belhot, R.V. O Emprego de System Dynamics no ensino e treinamento. XXX Congresso Brasileiro de Ensino de Engenharia, Piracicaba - SP, 2002. Disponível em: http://www2.eesc.usp.br/aprende/images/arquivos/o_emprego_de_system_dynamics.pdf. Acesso em: 26/03/2015.

GAVIRA, Muriel de Olivera.; Simulação Computacional como uma ferramenta de aquisição de conhecimento. Tese de Mestrado apresentada no Curso de Engenharia de Produção. Escola de Engenharia de São Carlos da Universidade de São Paulo. 2003. Disponível em: www.teses.usp.br/teses/disponiveis/18/18140/tde-20052003.../Gavira1.pdf. Acesso em: 02/06/2015.

GUARANÁ. Guaraná Enterprise Application Integration. Disponível em: <http://www.guarana-project.net/>. Acesso em: 20/04/2015.

HAUGG, Igor G.; Frantz, Rafael Z.; Roos-Frantz, Fabricia, Sawicki, Sandro. Uma Proposta para Comparação de Provedores de Computação em Nuvem desde uma Perspectiva de Integração de Aplicações. Salão do Conhecimento. XXII Seminário de Iniciação Científica. Universidade Regional do Noroeste do Estado do Rio Grande do Sul. 2014. Disponível em: <https://www.revistas.unijui.edu.br/index.php/salaconhecimento/article/view/3814/3196> . Acesso em: 20/06/2015.

KELTON, W. D.; Sadowski, R. P.; Zupick, N. Simulation with Arena. Fifth Edition, New York: McGraw- Hill, 2009.

LANDA, Fabiane Teixeira de. Utilização de Simulação de Eventos Discretos para Análise da Operação de Locomotivas de Auxílio nos Trens de Carga: Caso Mrs. Monografia apresentada ao Curso de Engenharia da Produção. Universidade Federal de Juiz de Fora. Minas Gerais. 2007. Disponível em: www.ufjf.br/ep/files/2014/07/2007_3_Fabiane.pdf. Acesso em: 20/05/2015.

LEMOS, Ana Carina Dri. Aplicação de uma Metodologia de Ajuste do Sistema Kanban em um Caso Real Utilizando a Simulação Computacional. Dissertação de Mestrado em Engenharia de Produção. Universidade Federal de Santa Catarina, 1999. Disponível em: <https://repositorio.ufsc.br/xmlui/bitstream/handle/123456789/81207/174797.pdf?sequence=1&isAllowed=y>. Acesso em: 20/05/2015.

MIRANDA, Juliano Coelho; Silveira, Aimée Piedade; Silva; Cristiane Rodrigues da; Belarmino, Daiana Guimarães. Monografia apresentada ao curso de Ciência da Computação do Centro Universitário do Sul de Minas – UNIS/MG. Varginha, 2006.

Disponível em:

http://www.cienciadacomputacao.unis.edu.br/files/2010/10/013_Arena.pdf. Acesso em: 20/05/2015.

OLIVEIRA, José Benedito de. Simulação Computacional: Análise de Um Sistema de Manufatura em Fase de Desenvolvimento. Dissertação de Pós-Graduação de Engenharia da Produção. Universidade Federal de Itajubá, 2007. Disponível em:

www.iepg.unifei.edu.br/arnaldo/.../Benedito%20de%20Oliveira.pdf. Acesso em: 15/04/2015.

PORTO, A.J.V.; Lobão, E.C. Evolução das Técnicas de Simulação. Revista Produção, Abepro, v. 9, n. 1, p. 12-22, 1999.

PRADO, Darci Santos do. Usando o Arena em Simulação. DG: Editora de Desenvolvimento Gerencial. Belo Horizonte, 1999.

SABBADINI, Francisco Santos; Gonçalves, Antônio Augusto. Simulação computacional aplicada na área de saúde. III Semana de Tecnologia - Curso de Sistemas de Informação - Unesa. 2005. Disponível em:

www.aedb.br/seget/arquivos/artigos05/22_simulacao_seget.pdf. Acesso em: 23/03/2015.

SOUZA, Cesar Alexandre de; Saccol, Amarolinda Zanela. Sistemas ERP no Brasil. Editora Atlas, 2003.

TRIVELATO, Gilberto da Cunha. Técnicas de Modelagem e Simulação de Sistemas Dinâmicos. Instituto Nacional de Pesquisas Espaciais. São José dos Campos, 2003.

Disponível em:

<http://mtc-m05.sid.inpe.br/col/sid.inpe.br/jeferson/2003/07.08.08.27/doc/INPE%20-%209665%20-%20NTC.pdf>. Acesso em: 28/05/2015.