

**UNIVERSIDADE FEDERAL DE SERGIPE**  
**NÚCLEO DE SISTEMAS DE INFORMAÇÃO**  
**CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

# **RELATÓRIO DO ESTÁGIO SUPERVISIONADO**

**RAFAEL DOS SANTOS MENDONÇA**  
**ISRAEL MENESES SANTOS**

**ITABAIANA – SERGIPE**  
**JUNHO DE 2010**

# **RELATÓRIO DO ESTÁGIO SUPERVISIONADO**

**RAFAEL DOS SANTOS MENDONÇA  
ISRAEL MENESES SANTOS**

Relatório de estágio apresentado ao Curso de Sistemas de Informação como parte das exigências da disciplina Estágio Supervisionado, sob a orientação pedagógica do Prof. Marcos Barbosa Dósea e orientação técnica do Analista de Centro de Processamento de Dados da Universidade Federal de Sergipe Diego Vasconcelos e Carmo.

**ITABAIANA – SERGIPE  
JUNHO DE 2010**

## **RESUMO**

O Centro de Processamento de Dados da Universidade Federal de Sergipe disponibiliza uma variedade de sistemas que são utilizados para atender as demandas dos departamentos e demais setores administrativos da universidade. Esses sistemas além de não atender completamente as necessidades da instituição, utilizam bases de dados sem integração gerando sérios problemas de consistência e duplicidade de informações. Esses problemas levaram a instituição a adquirir um novo sistema integrado que atende a maior parte das necessidades da instituição, porém existem alguns módulos que não estão totalmente adequados à realidade da instituição, por exemplo, o módulo de restaurante.

O módulo de restaurante existente no novo sistema, faz o gerenciamento e a administração necessária para o controle de estoque do restaurante universitário, porém não faz o controle da venda de refeições nem o controle de acesso ao mesmo. Este relatório de estágio tem como objetivo discutir o trabalho realizado para implementação dessas funcionalidades no novo sistema.

## SUMÁRIO

<b>1. INTRODUÇÃO .....</b>	<b>6</b>
1.1 Local do estágio .....	6
1.2 Objetivos .....	6
1.3 Estrutura do relatório .....	7
<b>2. REFERÊNCIAL TEÓRICO.....</b>	<b>8</b>
<u>2.1 Rational Unified Process (RUP) .....</u>	<u>8</u>
<u>2.2. BPM (Business Process Management) .....</u>	<u>14</u>
<u>2.3. Modelagem de dados .....</u>	<u>15</u>
<u>2.4. SIPAC- Módulo Restaurante .....</u>	<u>16</u>
<b>3. FERRAMENTAS UTILIZADAS .....</b>	<b>17</b>
<u>3.1 Talend.....</u>	<u>17</u>
<u>3.1.1 Análise Técnica .....</u>	<u>19</u>
<u>3.2. Postgres .....</u>	<u>19</u>
<u>3.3. JBOSS .....</u>	<u>20</u>
<u>3.4. JSF.....</u>	<u>20</u>
<u>3.5. BizAgi.....</u>	<u>21</u>
<b>4. ATIVIDADES DO ESTÁGIO.....</b>	<b>22</b>
<u>4.1. Modelos de Negócio .....</u>	<u>24</u>
<u>4.2. Protótipos .....</u>	<u>26</u>
<u>4.3. Casos de uso.....</u>	<u>32</u>
<u>4.4. Diagrama de Entidade e Relacionamento do Modulo Restaurante .....</u>	<u>34</u>
<b>5. CONCLUSÃO .....</b>	<b>36</b>
<b>6. REFERENCIAS BIBLIOGRAFICAS.....</b>	<b>37</b>

## ÍNDICE DE FIGURAS

<i>Figura 1: Estrutura do RUP.....</i>	<i>09</i>
<i>Figura 2: Fases do RUP.....</i>	<i>12</i>
<i>Figura 3: Ambiente de desenvolvimento do Talend.....</i>	<i>18</i>
<i>Figura 4: BizAgi Ambiente de Desenvolvimento .....</i>	<i>22</i>
<i>Figura 5: Modelo de negócio representando a compra de tickets .....</i>	<i>25</i>
<i>Figura 6: Modelo de negócio representando a entrada do cliente no restaurante. ....</i>	<i>26</i>
<i>Figura 7: Cartões avulsos .....</i>	<i>27</i>
<i>Figura 8: Cadastro de visitantes.....</i>	<i>27</i>
<i>Figura 9: Relatório para Consulta de Refeições por Local .....</i>	<i>28</i>
<i>Figura 10: Relação de estudantes isentos .....</i>	<i>28</i>
<i>Figura 11: Relatório para consulta de refeições.....</i>	<i>29</i>
<i>Figura 12: Relatório para fechamento de caixa .....</i>	<i>29</i>
<i>Figura 13: Relatório de refeições por matricula .....</i>	<i>33</i>
<i>Figura 14: Relatório para consulta de refeições por categoria de usuário.....</i>	<i>33</i>
<i>Figura 15: Cartões para militares .....</i>	<i>31</i>
<i>Figura 16: Cartões de servidores .....</i>	<i>31</i>
<i>Figura 17: Cartão de visitantes .....</i>	<i>31</i>
<i>Figura 18: DER schema restaurante.....</i>	<i>35</i>

## **1. INTRODUÇÃO**

Este trabalho descreve a implantação de novas funcionalidades no módulo Restaurante no novo sistema adquirido pela Universidade Federal de Sergipe. Nele não foi necessário fazer importação de dados, já que as informações do sistema legado eram desnecessárias. Mas foram necessárias fazer a engenharia para implantação no módulo das funcionalidades para venda de vale-refeições e controle da catraca eletrônica.

### ***1.1 Local do estágio***

O estágio foi realizado no CPD do Campus Universitário Professor Alberto Carvalho – UFS (Universidade Federal de Sergipe). Onde foi disponibilizando pela UFS dois computadores para a realização do trabalho. Este estágio período de 15 de março de 2010 a 21 de julho de 2010 tendo uma duração 20 horas semanais.

### ***1.2 Objetivos***

O objetivo geral do estágio foi a alteração do modulo de restaurante do novo sistema da UFS para adequá-lo as necessidades da instituição.

Os objetivos específicos do Estágio Supervisionado foram:

- Levantamento dos requisitos para implementação das alterações no novo sistema
- Criação de modelos de negócio e casos de uso.
- Criação de protótipos.
- Criação de novo modelo de dados para sistema.
- Configuração de ambiente de desenvolvimento.

- Implementação das novas funcionalidades.
- Realização de teste nas funcionalidades desenvolvidas;
- Treinamento dos usuários.

### ***1.3 Estrutura do relatório***

Este relatório está organizado como descrito a seguir. Na Seção 2 é apresentada o referencial teórico para o desenvolvimento do projeto. As ferramentas e tecnologias usadas são apresentadas na Seção 3. As atividades do estágio, bem como os resultados alcançados na Seção 4. Finalmente, na Seção 5 as conclusões das atividades do estágio.

## 2. REFERÊNCIAL TEÓRICO

Nesta seção são apresentados os principais conceitos necessários ao entendimento das atividades realizadas durante o estágio. O processo de desenvolvimento adotado foi o *Rational Unified Process* apresentado na Seção 2.1. Para elaboração do modelo de negócio do sistema foi utilizado o BPM (*Business Process Management*) apresentado na Seção 2.2 e na Seção 2.3 são apresentadas as principais ferramentas e tecnologias utilizadas para o desenvolvimento das funcionalidades propostas.

### 2.1 *Rational Unified Process* (RUP)

O desenvolvimento de software de qualidade tem se tornado cada vez mais complexo nos últimos anos, devido ao surgimento de novas tecnologias e a necessidade de realização de mudanças nos software para atender às exigências dos clientes. [13]

O *Rational Unified Process* (RUP) é um dos principais processos de desenvolvimento de software usados no mercado. Ele tem com uma de suas principais características a busca constante por qualidade. O RUP define um guia que deve ser seguido no desenvolvimento de software. No entanto ele pode ser adaptado para ser utilizado em diversas situações. [13]

O RUP utiliza seis boas práticas de desenvolvimento de software para garantir o sucesso dos projetos que utilizem o processo. Essas boas práticas são abordagens comprovadas comercialmente que, quando combinadas, atacam a origem dos principais problemas que ocasionam o fracasso de um projeto de software, evitando que tais problemas ocorram ou antecipando-os de forma que o projeto possa ser reavaliado. As seis boas práticas adotadas pelo RUP são [13]:

**Desenvolver software iterativamente** - O RUP utiliza um ciclo de vida iterativo e incremental como modelo de desenvolvimento. Nessa abordagem um conjunto de



atividades de modelagem de negócios, requisitos, análise e projeto, implementação, testes e implantação são desenvolvidos de forma seqüencial a cada iteração, mas em diferentes proporções dependendo da fase projeto.

**Gerenciar requisitos** - O gerenciamento de requisitos é uma abordagem sistemática para localizar, documentar, organizar e controlar os requisitos variáveis do sistema. Requisitos são funções ou capacidades que o sistema deve atender.

**Utilizar arquiteturas baseadas em componentes** - A arquitetura de um software define a organização de um sistema de software ou estrutura de elementos que o compõem. Determina como esses elementos interagem entre si e como se combinam em subsistemas para compor um sistema maior.

**Modelar o software visualmente** - Modelos são utilizados para descrever sistemas de uma forma particular e simplificada da realidade. Através de modelos podemos entender e visualizar de uma maneira mais simples e clara o que deve ser desenvolvido.

**Verificar a qualidade do software de forma contínua** - A qualidade do produto está ligada à qualidade do software desenvolvido, assim como seus componentes, arquitetura, etc.

**Controlar as mudanças do software** - O controle de mudanças permite que mudanças de requisitos possuam um melhor monitoramento, facilita a organização e comunicação entre equipes trabalhando paralelamente e permite um maior controle sobre os produtos gerados no processo.

A estrutura do RUP é organizada em duas dimensões, como mostrado na Figura 1. O eixo vertical é a parte estática do processo, apresentando as principais disciplinas que compõem o processo. O eixo horizontal representa o tempo, o aspecto dinâmico do processo, e apresenta o aspecto do ciclo de vida do processo em termos de ciclos, fases, iterações e marcos.

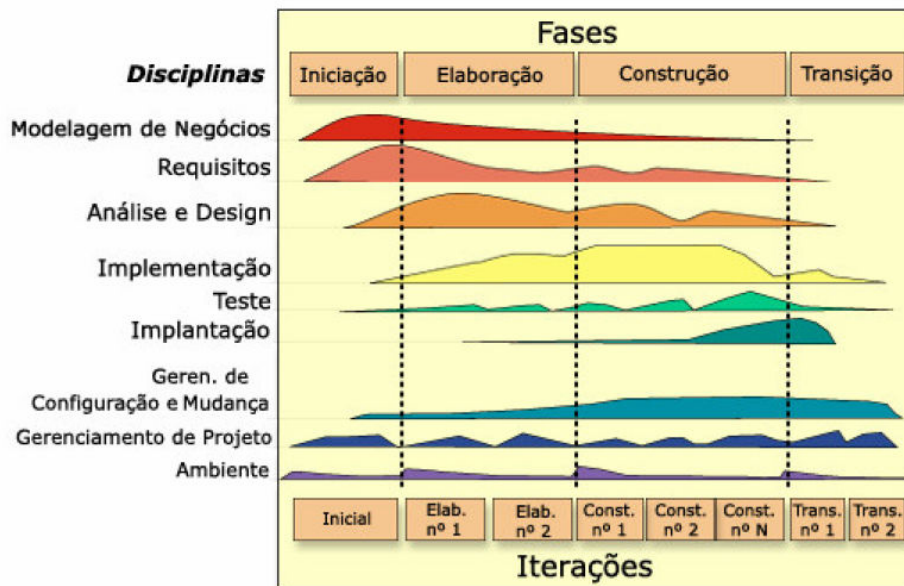


Figura 2- Estrutura do RUP

O eixo horizontal representa o tempo e mostra os aspectos do ciclo de vida do processo à medida que se desenvolve. O eixo vertical representa as disciplinas, que agrupam as atividades de maneira lógica, por natureza. A parte estática do RUP, representada pelo eixo vertical, é descrita através dos conceitos de papéis, atividades, artefatos, fluxos de trabalho e disciplinas [14].

Um papel define o comportamento e as responsabilidades assumidas por uma pessoa ou um conjunto de pessoas trabalhando em equipe. Cada papel é responsável pela criação, modificação e controle de determinados artefatos que são gerados a partir da realização de certas atividades ligadas ao papel em questão [14].

Uma atividade é um trabalho desempenhado por um indivíduo. O indivíduo que realiza a atividade deve assumir o papel responsável pela realização de tal atividade. As atividades estão diretamente ligadas a criação ou manutenção de artefatos e são executadas repetidas vezes em diferentes iterações [14].

Um artefato é um pedaço de informação que é produzido, modificado e utilizado por um processo. Artefatos são utilizados como entrada para a realização de uma atividade e são resultados das saídas das atividades. Um artefato pode ser um

documento, um modelo, um arquivo executável, código-fonte ou qualquer outro tipo de documentação que seja importante para o projeto [14].

Um fluxo de trabalho, ou disciplina, descreve uma seqüência de atividades e interações entre papéis de forma a produzir um conjunto de artefatos. Esses fluxos são descritos de forma a apresentar todos os papéis, atividades e artefatos envolvidos, mostrando como esses papéis interagem para produzir os artefatos em um nível mais detalhado [14].

Uma disciplina mostra todas as atividades que devem ser realizadas para produzir um determinado conjunto de artefatos. Essas disciplinas são descritas em nível geral por um resumo de todos os papéis, atividades e artefatos envolvidos. As principais disciplinas do RUP são:

### **Disciplina de Modelagem de Negócios**

O objetivo dessa disciplina é Compreender o negócio, isso significa que a equipe devem compreender a estrutura e a dinâmica do negócio, os atuais problemas na organização e possíveis melhorias. Eles também devem garantir um entendimento comum da organização-alvo entre os clientes, usuários finais e desenvolvedores.

### **Disciplina de Requisitos**

Esta disciplina explica como levantar pedidos das partes interessadas, denominados "stakeholders", e transformá-los em requisitos que devem ser implementados no sistema.

### **Disciplina de Análise e Projeto**

O objetivo da análise e projeto é mostrar como o sistema vai ser realizado.

### **Disciplina de Implementação**

A finalidade da implementação é definir a organização do código em termos de subsistemas de implementação organizados em camadas, implementar classes e

objetos em termos de componentes, testar os componentes desenvolvidos como unidades e integrar os resultados produzidos pela equipe ao sistema executável.

### **Disciplina de Teste**

A finalidade do Teste é localizar e expor os pontos fracos do software.

### **Disciplina de Implantação**

A Disciplina Implantação descreve as atividades que garantem que o produto de software será disponibilizado aos usuários finais.

### **Disciplina de Ambiente**

O ambiente enfoca as atividades necessárias para configurar o processo para um projeto. Ele descreve as atividades necessárias para desenvolver as diretrizes de apoio a um projeto.

### **Disciplina de Configuração e Gerência de Mudança**

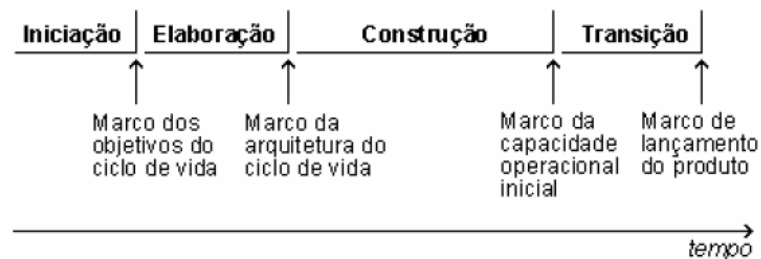
A gestão de configuração e Gerência de Mudança é pelo controle nas alterações dos produtos gerados no decorrer do projeto

### **Disciplina de Gerência de Projeto**

Esta disciplina concentra-se principalmente sobre os aspectos importantes de um processo de desenvolvimento iterativo, ela leva em consideração questões como gerência de risco, planejamento das interações, além de outros aspecto relativos ao gerenciamento do projeto.

A estrutura dinâmica do RUP representa o desenvolvimento do processo no decorrer do tempo. O processo é caracterizado pelo desenvolvimento iterativo e por uma evolução incremental e com foco na redução dos riscos. O ciclo de vida de um projeto utilizando o RUP é dividido em uma sucessão de pequenos ciclos do modelo clássico, em cascata, chamado de modelo iterativo [2].

Existem quatro marcos principais no RUP que dividem as iterações em quatro fases: iniciação, elaboração, construção e transição. Cada fase é finalizada quando um dos marcos principais é realizado, como mostrado na Figura 2..



**Figura 2- Fases do RUP**

### **Iniciação**

A principal meta da fase de iniciação é a definição dos objetivos do ciclo de vida do projeto. Nesta fase é estabelecido o escopo do projeto a ser desenvolvido, critérios de aceitação e o que deve ou não deve estar no produto.

O foco da fase de iniciação é concentrado nas atividades voltadas para modelagem de negócios, requisitos, gerência de projeto e ambiente. O marco para o final da fase é o marco dos objetivos do ciclo de vida, que avalia a viabilidade do projeto.

### **Elaboração**

A principal meta da fase de elaboração é a definição de uma arquitetura estável para o sistema. A arquitetura definida deve ser estável o suficiente para diminuir os riscos do projeto e dar suporte à determinação dos custos e programação para a conclusão do desenvolvimento.

A fase de elaboração é focada nos requisitos, análise e projeto, além das atividades de gerência de projetos e ambiente, mas já é realizada a implementação do protótipo da arquitetura. O marco para o final da fase é o marco da arquitetura do ciclo de vida que estabelece uma base para a arquitetura do sistema.

### **Construção**

Na fase de construção os últimos requisitos são detalhados e é concluído o desenvolvimento do sistema. Na fase de construção as atividades de análise e implementação são o foco principal. O marco para o final da fase é o marco da capacidade operacional inicial, que determina se o produto está pronto para ser implantado.

### **Transição**

A fase de transição deve assegurar que o software estará disponível para os usuários finais. Nesta fase os objetivos estão voltados para a realização de testes dos usuários, treinamentos, distribuição e vendas, engenharia voltada para implantação, atividades de ajustes, etc.

Na fase de transição o foco é voltado para testes e gerência de configuração. O marco final desta fase é o marco de lançamento do produto.

## **2.2. BPM (Business Process Management)**

BPM é um conceito que une gestão de negócios e tecnologia da informação para melhorar os resultados das organizações utilizando processos de negócios e modelos que descrevem as etapas dos processos.

Para o desenvolvimento desses modelos, são utilizados métodos, técnicas e ferramentas para analisar, modelar, publicar, otimizar e controlar processos envolvendo várias áreas da empresa e fontes de informações. Ele tem como objetivo prover a junção entre os processos de negócio e as estratégias de marketing, além dos objetivos e a cadeia de valores das organizações, para gerar modelos de negócios que realmente mostrem como os processos funcionam e assim modificá-los de modo a atender às necessidades da empresa.

A utilização deste vem crescendo muito nos últimos anos, isso tudo graças a sua utilidade e rapidez para melhorar e aperfeiçoar os processos de negócio dentro das

empresas. Existem algumas vertentes do BPM cujo uso depende das necessidades de cada empresa.

Utilizando ferramentas de BPM, os gestores podem analisar processos se baseando em dados reais, o que permite a visualização de vários problemas na empresa, como por exemplo, pessoas que estão atrasando tarefas, o percentual que isso ocorre ou ainda processos ou tarefas que foram concluídas, entre muitos outros.

## **2.3. Modelagem de dados**

Além do Modelo de negócio, outro artefato muito importante para o desenvolvimento de qualquer aplicativo é o modelo de dados. Um modelo de dados é uma descrição dos tipos de informações que estão armazenadas em um banco de dados. Ele irá descrever toda estrutura dos dados que serão persistidos no banco. O modelo de dados não irá mostrar os dados que estão salvos no banco, o que ele faz é mostrar quais informações sobre os dados ficarão salvas no banco.

### **2.3.1 Modelo Conceitual**

Representa as regras de negócio sem levar em consideração limitações tecnológicas ou de implementação. O modelo gerando é de fácil entendimento tanto dos usuários quanto dos desenvolvedores, já que só possui os principais elementos que são manipulados pelo negócio.

### **2.3.2 Modelo lógico**

Nesse modelo já começa a aparecer a preocupação com limitações existentes nas tecnologias de banco usadas (Orientado a objetos, relacional, etc.). Nesse momento cria-se um modelo derivado do modelo conceitual só que agora respeitando o as limitações impostas pelo SGBD que será utilizado.

### **2.3.3. Modelo Físico**

É o modelo usado para estabelecer a organização mais adequada para os arquivos e os métodos de indexação mais eficientes. Considera características físicas do SGBD que será utilizado.

### **2.3.4. Modelo Entidade-Relacionamento**

Um dos principais objetivos da modelagem de dados é evitar a redundância dos dados persistidos e criar estruturas que permitam o acesso rápido e eficiente às informações armazenadas em banco. Existem várias ferramentas usadas para facilitar a criação de modelos de dados, a mais utilizada é o modelo entidade-relacionamento.

O Modelo Entidade-Relacionamento (MER) foi desenvolvido pelo professor Peter Chen, e tinha com o objetivo representar as estruturas de dados de uma forma mais natural e mais próxima do mundo real dos negócios.

O Modelo E-R propõe que a realidade seja visualizada sob três pontos de vista. Assim, há três conceitos fundamentais no Modelo E-R: Entidade, Atributo e Relacionamento.

Uma entidade corresponde à representação de todo e qualquer substantivo, concreto ou abstrato, sobre o qual se precisa armazenar e/ou recuperar informações. Um atributo é uma característica (ou informação) pertencente a uma entidade e relacionamentos. Um relacionamento no contexto do Modelo E-R é a forma como os objetos se relacionam.

O relacionamento é um conceito complexo, pois é representa as regras e políticas do negocio que esta sendo modelado.

## **2.4. SIPAC- Módulo Restaurante**

O módulo Restaurante Universitário no novo sistema adquirido pela Universidade Federal de Sergipe tem com principal objetivo a informatização das atividades realizadas nos restaurante da UFS, ele difere do sistema atual no que se refere a utilização da catraca eletrônica para controle de acesso dos usuários.



O sistema antigo atende as necessidades do restaurante em relação à venda de refeições, já o novo sistema conta com uma boa estrutura para o controle administrativo e gerencial dos recursos de um restaurante, porém difere no processo para venda das refeições. Dessa forma, foram identificadas necessidades de alterações no sistema que será implantado para incluir novas funcionalidades como impressão de cartões e algumas a geração de alguns relatórios de controle necessários para o funcionamento efetivo do sistema,

Apesar de possuir algumas funções que existem no aplicativo usado atualmente no restaurante, o novo sistema conta com uma boa estrutura para o controle dos recursos, como relatórios sobre despesas e receitas, além de relatórios sobre vendas de refeições e toda a parte para cadastro de despesa, valor de refeições. Isso significa que todo o sistema pode ser aproveitado, bastando incluir a funcionalidade que será necessária levando em vista o sistema atual.

### **3. FERRAMENTAS UTILIZADAS**

Para o desenvolvimento deste trabalho foi necessário o uso de algumas ferramentas que nos auxiliaram tanto na importação dos dados quanto em modificações diretamente no código que foram necessárias. As ferramentas usadas foram o Talend [1], o banco de dados Postgres [2], o JBOSS [8] e o JSF [7]. A seguir será mostrado um pouco de cada uma delas.

#### **3.3 Talend**

Para o desenvolvimento do nosso trabalho, foi necessário do uso de uma ferramenta ELT, Extract Load and Transform, que é uma maneira de manipulação de grande quantidade de dados e organizá-los na estrutura de um DW (Data Warehouse) [9].

Ela é uma ferramenta nova, lançada em 2006, mas que já faz grande sucesso entre as empresas e os desenvolvedores. Suas facilidades no uso, além da

possibilidade de alterações em partes do projeto utilizando a linguagem JAVA [10], a tornam uma ótima ferramenta para integrar os dados.

Uma ferramenta gratuita de código aberto que faz essa função de manipulação dos dados é o Talend Open Studio. Ela possui uma arquitetura distribuída, o que permite o uso de múltiplos servidores que executam o processo, o que diferencia ela de outras soluções existentes.

A interface gráfica do Talend é baseada no eclipse, que auxilia na transformação dos dados e facilita a visualização e usabilidade do sistema.

Ela é bastante usada para migrar dados, como também transformação de cargas completas de dados.

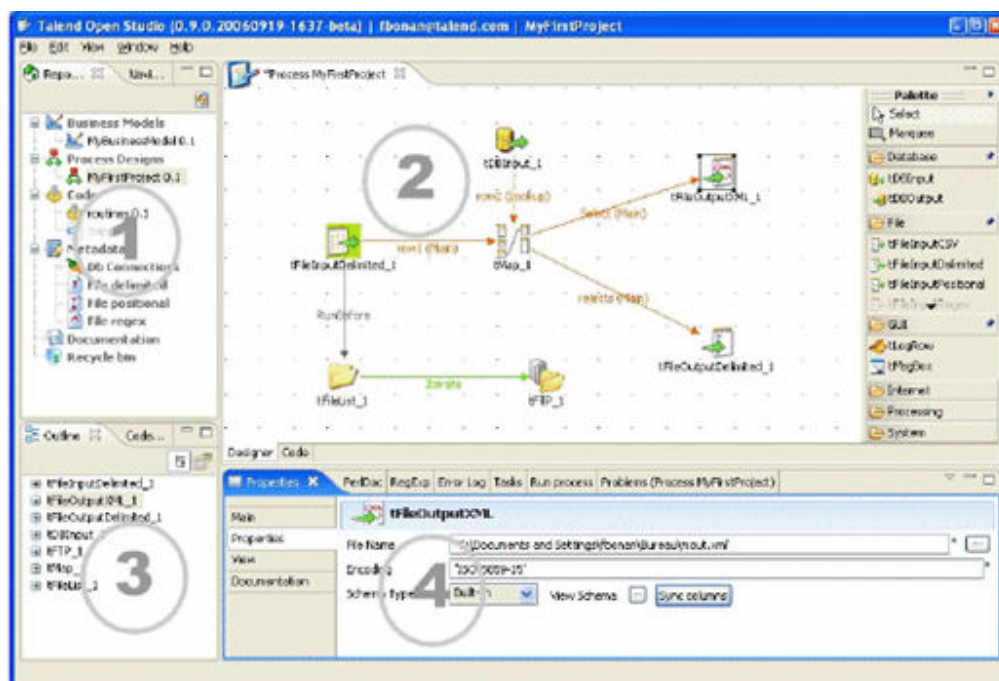


Figura 3: Ambiente de desenvolvimento do Talend

O sistema é composto por quatro painéis. O primeiro serve para armazenar as informações técnicas sobre o projeto aberto. É chamado de REPOSITORY e dá para se ter uma idéia geral do projeto.

O segundo painel é onde estão as ferramentas que podem ser usadas nos projetos. Dá para se visualizar a modelagem e as transformações feitas.

O terceiro painel dá informações específicas em relação ao componente ou diagrama selecionado. Também a partir dele se pode visualizar o código gerado pelo componente.

O quarto painel possui várias abas que podem nos dar das mais variadas informações. Nela se podem alterar as propriedades de componentes, visualizar a execução do programa, conferir logs do projeto, informando erros que podem vir a acontecer durante o projeto.

### **3.3.1 Análise Técnica**

O Talend tem a capacidade de fazer a extração de dados de várias fontes diferentes, e a migração para várias fontes também.

O Talend trabalha com processamento distribuído, podendo dar prioridade para processos mais importantes e agilizando o processo. Ela também possui alta escalabilidade, o que permite manipular grande quantidade de trabalho de maneira uniforme.

Ele possui um repositório de metadados para auxiliar o desenvolvimento dos projetos.

O Talend também dá suporte a agendamento de execução de tarefas, além disso, disponibiliza uma interface gráfica para auxiliar e facilitar os usuários na manipulação dos dados.

## **3.2. Postgres**

Lançado em 1989, só veio a ganhar repercussão mesmo a partir de 1996, quando inseriu a linguagem SQL no seu código. O Postgres é um projeto Open Source, um SGBD que usa objeto relacional, já é hoje entre os de código aberto, considerado por muitos, o mais avançado.

Seu sistema permite o uso de muitas coisas como consultas complexas, chaves estrangeiras, integridade transacional, controle de concorrência multi-versão, suporte ao modelo híbrido objeto relacional, gatilhos, views, linguagem procedural com suporte a várias linguagens de programação, indexação por texto, entre outros.

### **3.3. JBOSS**

Os componentes precisam de um ambiente de execução denominado container, o qual provê os serviços necessários para execução do mesmo. Neste projeto o servidor de aplicação utilizado para os componentes Enterprise Beans foi o JBOSS.

JBOSS é um servidor de aplicação feito na plataforma Java. Foi lançado em 1999, e é um servidor completo que implementa a especificação J2EE da Sun. Por ser feito na plataforma Java ele tem como uma das características ser portátil para qualquer plataforma que possua uma implementação da máquina virtual Java configurada corretamente.

A implantação de aplicações no JBOSS é muito simples, bastando copiar os arquivos para pasta do servidor, não é preciso reiniciar o serviço para fazer o deploy, essa característica é chamada de hot deployment.

O JBOSS já vem com algumas configurações padrões, all, default e minimal. Na configuração all todos os serviços disponíveis são iniciados, na configuração default, os serviços que permitem clustering e o serviço RMI/IIOP não são iniciados, e na configuração minimal são excluídos o Java Message Service (JMS), o container EJB, de Servlet/JSP entre outros, mas também existe a possibilidade de criação de novas configurações personalizadas alterando os arquivos de configurações existentes.

### **3.4. JSF**

O objetivo da plataforma J2EE[10] é reduzir o custo e a complexidade de desenvolver serviços multicamada. A tecnologia J2EE permite o desenvolvimento de aplicações em "n" camadas, separando em camadas de apresentação, negócio e dados. A tecnologia dá suporte a recursos de conexão de banco de dados compartilhada, "pooling", que são componentes para publicação de dados dinâmicos na WEB (JSP, Servlet), componentes de negócio e persistência (EJB), entre outros. O J2EE incorpora novos recursos a linguagem JAVA, sendo por isso, mais que uma metodologia. A plataforma define segundo a especificação da Sun, empresa criadora da linguagem Java, os seguintes tipos de contêiner: [7] [10]

- EJB (Enterprise Java Beans): um dos contêineres mais conhecidos da arquitetura, ele especifica um ambiente para enterprise beans como serviços de segurança, concorrência, gerenciamento de ciclo de vida, transação, entre outros serviços. Os beans rodam no servidor J2EE e eles são os componentes de negócio na aplicação. Existem três tipos de Enterprise Beans: session beans, entity beans e message-driven beans.

Session Beans representam os serviços que podem ser utilizados pelos clientes ou outros componentes. Entity Beans representam os dados persistentes que são usados na aplicação. Message-driven beans são usado para criar componentes de integração com outros sistemas, pois suas ações são executadas através de mensagens que podem vir de diversos sistemas que podem ser implementados com tecnologias diferentes.

- WEB: trata-se da implementação de contrato de componente da Web com a arquitetura J2EE, fornecendo ambiente para os componentes do tipo web que incluem desde segurança até gerenciamento de ciclo de vida e transação. Inclui Servlets (classes Java que podem ser carregadas dinamicamente e executar sob um servidor web) disponibilizadas por serviços de rede referentes aos pedidos e respostas enviados, com uso obrigatório do protocolo HTTP.

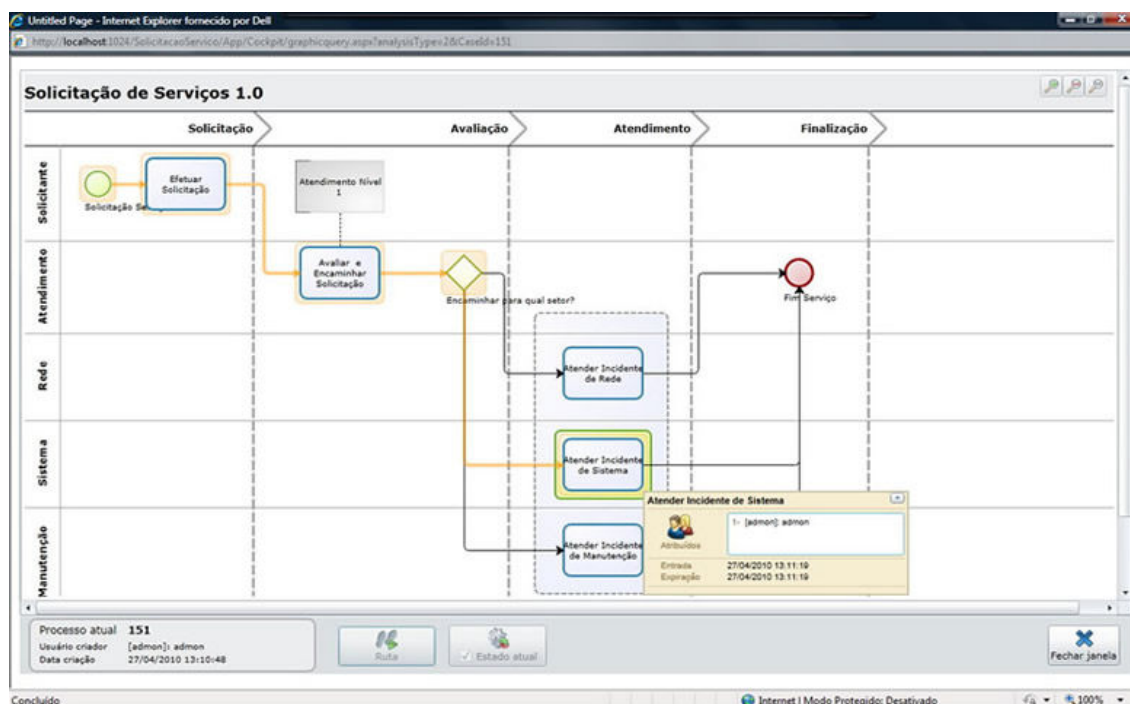
- Cliente da aplicação: Um contêiner que aceita e gerencia a execução de componentes do cliente da aplicação

### **3.5. BizAgi**

Para o desenvolvimento do nosso trabalho, tornou-se necessário a criação de alguns modelos de negócio que nos auxiliariam no desenvolvimento das novas aplicações. Foi utilizada a ferramenta BizAgi para criar os modelos.[12]

Ele foi concebido para projetar diagramas de processos em BPMN [11], definir regras de negócio, orquestrar outras aplicações, definir interface do usuário, otimização e balanceamento de carga de trabalho, portal web de trabalho, indicadores de desempenho de processos, monitor de atividades e alguns outros.

O uso desta ferramenta pode proporcionar uma modificação melhor dos dados dos processos automatizados.



**Figura 4: BizAgg Ambiente de Desenvolvimento**

## 4. ATIVIDADES DO ESTÁGIO

O objetivo inicial do estágio era a implantação do módulo Patrimônio do sistema SIPAC adquirido pela UFS. Foi realizado um estudo inicial e reuniões com usuário para melhor entendimento do módulo. Entretanto o módulo possuía um número muito grande de dependências forte com outros módulos do SIPAC ainda não convertidos, tornando-se inviável a continuidade do trabalho devido à restrição de tempo para realização do estágio.

Apesar de cumprir o desafio inicial para compreensão dos requisitos do sistema de Patrimônio, o conhecimento adquirido não foi utilizado para as demais atividades do estágio. Vale ressaltar o desafio para o entendimento do modelo de dados do sistema, composto por uma quantidade considerável de tabelas e relacionamento, além do

modelo de negócio e os requisitos do sistema de grande complexidade e definido por legislação federal para registro e controle de bens patrimoniais em instituições federais.

Dessa forma, foi necessário definir um outro módulo do sistema que pudesse ser implantado no tempo restante para conclusão das atividades do estágio. No período em que a equipe do CPD escolhia um novo módulo do sistema, houve uma realocação da equipe para que pudessemos contribuir com a implantação do módulo Protocolo do sistema SIPAC. Para essa tarefa foi realizado o estudo sobre a ferramenta Talend [1], apresentada na Seção 3.3. Essa atividade foi importante para auxiliar no início das atividades de importação dos dados do sistema legado de protocolos para o novo sistema adquirido pela UFS.

O novo módulo escolhido para dar continuidade de forma independente as atividades do estágio foi o módulo Restaurante do SIPAC. Foi iniciado um estudo para verificar a disponibilidade e a possibilidade da implantação desse sistema, chegando-se a conclusão que este, diferentemente do módulo Patrimônio, não possuía dependências fortes como outros módulos que fossem impeditivas a continuidade dos trabalhos. Foi definido que seria implantado a parte do módulo relacionado à catraca eletrônica, que já existia no sistema atual da UFS. O sistema novo não possui essa rotina de controle de uma catraca eletrônica, algo considerado fundamental pelos usuários do sistema. Cientes disso foi resolvido que seria necessário implementar essa nova rotina no sistema, mas para isso era importante entender um pouco mais do funcionamento da arquitetura do sistema.

Foram passados alguns treinamentos e manuais necessários para entender o sistema e poder modificá-lo. Além disso, também foi necessário entender como essa parte funcionava no sistema atual de restaurante da UFS. Foi feito um trabalho de entendimento que nos levou a conhecer esse módulo e assim conseguir competência suficiente para implementação das mudanças. O pouco tempo restante para conclusão do estágio foi utilizado para análise de requisitos e projeto das novas funcionalidades propostas para o sistema.

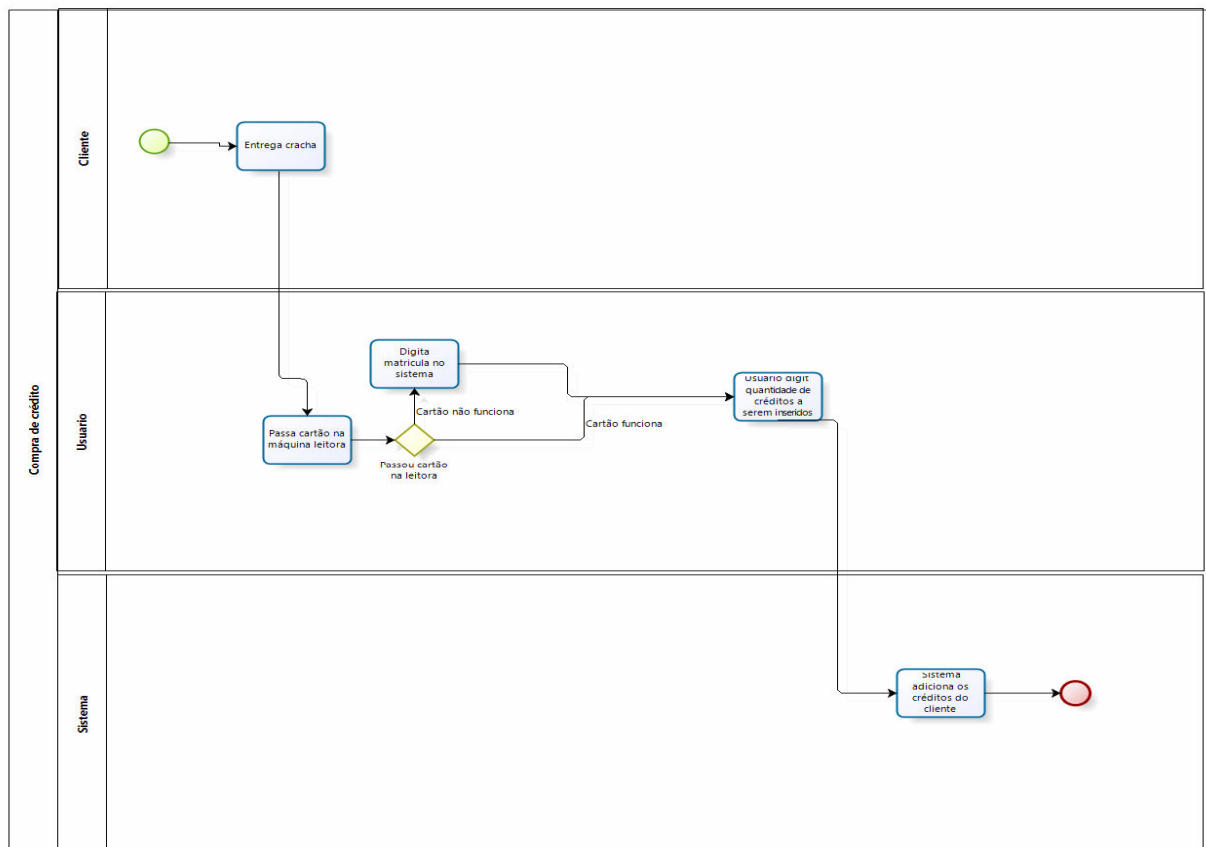
Na Seção 3.1 é apresentado o Modelo de Negócio do Sistema de restaurante, ferramenta bastante prática e simples para entendimento do funcionamento da organização. Na Seção 3.2 são apresentados os protótipos das telas que serão necessárias ao novo sistema. Na Seção 3.3 a descrição detalhada dos novos casos de uso incorporados ao sistema e na Seção 3.4 o Diagrama Entidade Relacionamento destacando-se as modificações que foram necessárias para o desenvolvimento do módulo.

#### **4.1. Modelos de Negócio**

Para facilitar o entendimento inicial do funcionamento do sistema, foram identificados e modelados dois processos de negócio que descrevem o funcionamento dos módulos que precisam ser desenvolvidos. Os processos de negócio identificados foram o Processo Compra de Tickets e o Processo Y. Ambos foram modelados na ferramenta BizAgi descrita na Seção 3.5.

A Figura 5 mostra como ocorre a rotina de um cliente, sendo este um aluno, professor, ou qualquer outro que tenha permissão para utilizar as dependências do restaurante universitário da UFS. O cliente chega ao local de compra de tickets para almoço, entrega o seu cartão universitário e informa ao usuário do sistema a quantidade de tickets que ele quer inserir em seu cadastro, além do pagamento. Logo após, o usuário passa o cartão na máquina de leitura, esta indica as informações pessoais do cliente. Caso isso não ocorra, o usuário digita o número de matrícula do cliente. Em seguida, o usuário digita a quantidade de tickets que devem ser inseridos no cadastro, caso não haja nenhum impedimento, os tickets são inseridos.



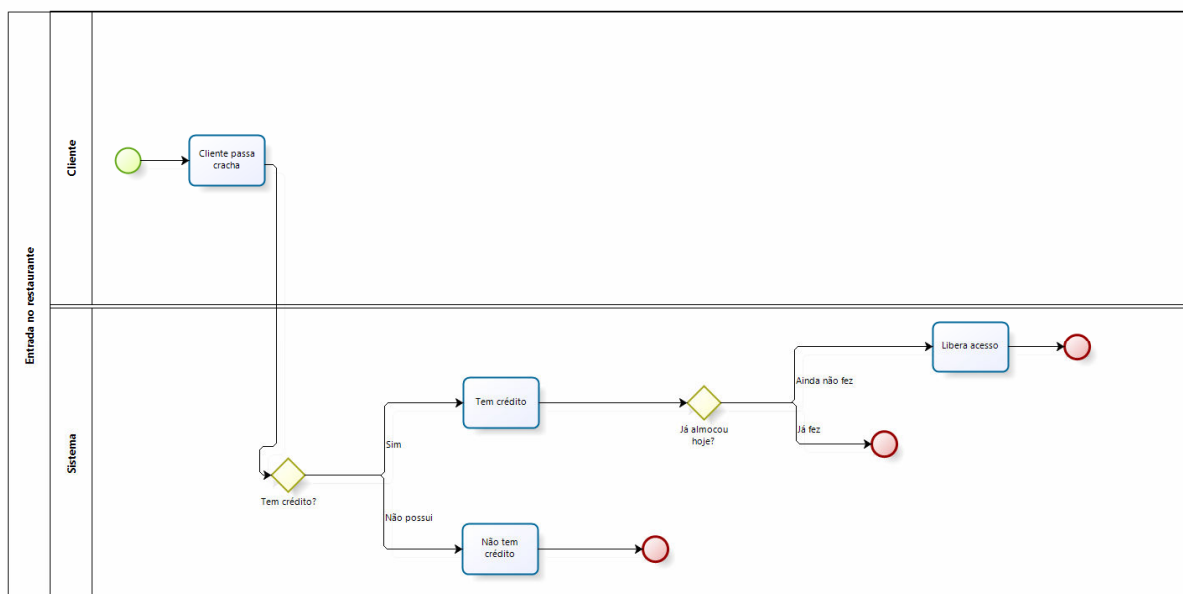


**Figura 5: Modelo de negócio representando a compra de tickets**

Também foi necessário um modelo que mostrasse como ocorre a etapa de um cliente ao entrar no restaurante. Este por sinal, podendo ser considerado uma das etapas mais importantes do projeto.

A etapa começa com o cliente chegando à catraca eletrônica, na entrada do restaurante. Ele passa o seu cartão universitário na maquina leitora, ligada a catraca, esta por sua vez acessa o sistema e busca as informações do cliente. Primeiro ela busca se o cliente possui realmente tickets em seu cadastro que lhe darão acesso ao restaurante, caso ele possua, o sistema busca agora informações de entrada no restaurante, se o cliente já fez um almoço no mesmo dia, a catraca não libera a entrada do usuário, se ele ainda não fez, a catraca é liberada e ele pode entrar no restaurante.

Caso ele não possua créditos, a entrada é rejeitada e a catraca não libera para entrada. A Figura 6 ilustra o processo descrito.



powered by  
Bizagi  
Process Modeler

Figura 6: Modelo de negócio representando a entrada do cliente no restaurante.

## 4.2. Protótipos

A prototipagem de sistemas de software pretende ser usada, principalmente, demonstrar os requisitos de um sistema que será desenvolvido, por esse motivo foram desenvolvidos protótipos das novas telas e relatórios que serão implementados.

A Figura 7 mostra o protótipo de Tela que será responsável pela geração de cartões avulsos. Essa é uma medida adotada pelo usuário quando existe algum problema com a máquina leitora de cartão na entrada do restaurante. Assim o funcionamento não é tão prejudicado e os dados são armazenados para relatórios.

Quantidade de Cartões

**Figura 7: Cartões avulsos**

A Figura 8 mostra o protótipo da Tela responsável pelo cadastramento de visitantes. No caso de alguém de fora da instituição que terá um tempo determinado de acesso ao restaurante. Nessa tela existem as funcionalidades necessárias de para manter um cadastro de visitante como incluir novo visitante, alterar um visitante já cadastrado, consulta de visitantes cadastrados e exclusão de registro de um visitante.

Matricula

Nome

Orgão

Identidade

CPF

**Figura 8: Cadastro de visitantes**

A Figura 9 mostra o relatório de totalização de refeições. Esse relatório é o responsável por informar o total de refeições por data e pelo tipo da refeição. Também será selecionado o restaurante que o usuário deseja obter os dados.



UNIVERSIDADE FEDERAL DE SERGIPE  
PROEST - PROREITORIA DE ASSUNTOS ESTUDANTIS  
RESUN - RESTAURANTE UNIVERSITÁRIO

**RELATÓRIO PARA CONSULTA DE REFEIÇÕES POR LOCAL**

PERÍODO DE: 01/01/2009 ATÉ 01/03/2009

REFEIÇÃO: JANTAR

LOCAL	QUANTIDADE	TOTAL
UFS	2123	1.262,00
TOTAIS	2123	1.262,00

**Figura 9: Relatório para Consulta de Refeições por Local**

A Figura 10 mostra o relatório que exibe a lista de todos os estudantes isentos do pagamento da taxa para entrada no restaurante.



UNIVERSIDADE FEDERAL DE SERGIPE  
PROEST - PROREITORIA DE ASSUNTOS ESTUDANTIS  
RESUN - RESTAURANTE UNIVERSITÁRIO

**RELAÇÃO DE ESTUDANTES ISENTOS**

MATRICULA NO RESUN	NOME	CURSO

**Figura 10: Relação de estudantes isentos**

A Figura 11 exibe o relatório que mostra a consulta de refeições que foram feitas em um intervalo de tempo. Exibe em relação ao valor das refeições.



UNIVERSIDADE FEDERAL DE SERGIPE  
PROEST - PROREITORIA DE ASSUNTOS ESTUDANTIS  
RESUN - RESTAURANTE UNIVERSITÁRIO

**RELATÓRIO PARA CONSULTA DE REFEIÇÕES**

**PERÍODO DE: 13/12/2008 ATÉ 13/01/2009**

**REFEIÇÃO: ALMOÇO**

VALOR PAGO	QUANTIDADE	TOTAL
0,00	1583	0,00
1,00	3381	3.381,00
2,00	84	168,00
<b>TOTAL DE REFEIÇÕES: 5048</b>		
<b>TOTAL ARRECADADO : R\$ 3.549,00</b>		

**Figura 11: Relatório para consulta de refeições**

O relatório exibido na Figura 12 mostra o fechamento do caixa. Esse relatório será gerado a partir de um intervalo de tempo determinado pelo usuário do sistema.



PROEST - PROREITORIA DE ASSUNTOS ESTUDANTIS  
RESUN - RESTAURANTE UNIVERSITÁRIO

**RELATÓRIO PARA FECHAMENTO DE CAIXA**

**PERÍODO DE: 13/12/2008 ATÉ 13/12/2009**

**REFEIÇÃO: ALMOÇO E JANTAR**

DATA	VALOR PAGO	QUANTIDADE	TOTAL
			0

**Figura 12: Relatório para fechamento de caixa**

A Figura 13 representa o relatório responsável por todas as informações sobre as refeições feitas por um usuário. O seu filtro é a matrícula do usuário.



Figura 13: Relatório de refeições por matrícula

O relatório para consulta de refeições por categoria de usuário é mostrado na Figura 14. Ela vai possuir filtros apenas de intervalo de tempo e do tipo da refeição.

**UNIVERSIDADE FEDERAL DE SERGIPE**  
**PROEST - PROREITORIA DE ASSUNTOS ESTUDANTIS**  
**RESUN - RESTAURANTE UNIVERSITÁRIO**

**RELATÓRIO PARA CONSULTA DE REFEIÇÕES POR CATEGORIA DE USUÁRIO**

**PERÍODO DE: 13/01/2009 ATÉ 13/03/2009**

**REFEIÇÃO: JANTAR**

**CATEGORIA**

**QUANTIDADE**

**TOTAL**

COLÉGIO DE APLICAÇÃO

0

0

DOCENTE

**TOTAIS**

0

0

Figura 14: Relatório para consulta de refeições por categoria de usuário

## Geração de cartões

As Figuras 15, 16 e 17 mostram os cartões que serão impressos para militares, servidores e visitantes para que estes possam ter acesso ao restaurante.

MATRÍCULA: 0000000  
NOME EDSON  
IDENTIDADE 0000000  
CPF: 0000000  
ORGÃO: SEGURANÇA

MATRÍCULA: 0000000  
NOME: MILTON  
IDENTIDADE: 0000000  
CPF: 0000000  
ORGÃO: SEGURANÇA

**Figura 15: Cartões para militares**

Cartões para os servidores:

MATRÍCULA: 0000000  
NOME SERAFIM  
CARGO: AUX. SERV. OPERAÇON

MATRÍCULA: 0000000  
NOME: GILSON  
CARGO: DIRETOR

**Figura 16: Cartões de servidores**

Cartão para visitantes:

MATRÍCULA: 0000000  
NOME ANA CRISTINA  
IDENTIDADE 0000000  
CPF: 0000000  
ORGÃO: CCV

MATRÍCULA: 0000000  
NOME: ELIZEU  
IDENTIDADE: 0000000  
CPF: 0000000  
ORGÃO: DIMOV

**Figura 17: Cartão de visitantes**

### 4.3. Casos de uso

Criar casos de uso é uma técnica usada para descrever e definir os requisitos funcionais de um sistema. Casos de uso descrevem um conjunto de cenários de interação com um usuário ou outro sistema. Eles podem ser facilmente adicionados e removidos do projeto de software assim que as prioridades mudam. Também podem servir como base para estimar, escalonar e validar esforços, além de serem de fácil entendimento pelos desenvolvedores e usuários.

A seguir são exibidos os casos de usos que foram descritos para as novas funcionalidades que serão implementadas no sistema.

#### Entrada no Restaurante

Mostra como irar ocorrer à entrada do usuário no restaurante universitário.

<b>Sumário</b>	Entrada do usuário no restaurante universitário
<b>Ator Primário</b>	Cliente.
<b>Ator Secundário</b>	-
<b>Pré-Condição</b>	-
<b>Pós-Condição</b>	-
Fluxo Principal	
1- Cliente pega o crachá 2- Cliente passa o crachá na máquina de leitura 3- Sistema capta as informações do cliente nas tabelas de usuários e exibi na tela. 4- Sistema testa se o cliente já fez um almoço no dia. Busca as informações na tabela. 5- Sistema confirma a entrada do cliente e libera a catraca eletrônica. Envia um pulso pra	



catraca que libera o acesso.
6- Cliente entra no restaurante.
7- Sistema confirma a entrada no dia e descontar um ponto nos créditos que o usuário tiver
Fluxos de Exceção
Linha 3: Usuário rejeitado por falta de crédito
Linha 4: Usuário rejeitado por já ter feito um almoço.

### Compra de Créditos

Mostra como irar ocorrer à compra de créditos dos usuários do restaurante.

<b>Sumário</b>	Compra de créditos dos usuários do restaurante
<b>Ator Primário</b>	Cliente.
<b>Ator Secundário</b>	Usuário
<b>Pré-Condição</b>	-
<b>Pós-Condição</b>	-
Fluxo Principal	
1- Usuário pega o crachá do cliente 2- Usuário passa o crachá na máquina de leitura 3- Sistema capta as informações do cliente e exhibi ao usuário 4- Usuário digita quantidade de crédito a ser inserido <b>5- Sistema confirma</b>	
Fluxos de Exceção	

<p>Linha 2: A máquina não consegue fazer a leitura e o usuário digita a matrícula no sistema, então Sistema capta as informações do cliente e exibi ao usuário.</p> <p>Linha 3: Usuário digita dados inválidos e Sistema retorna a operação para o início.</p>

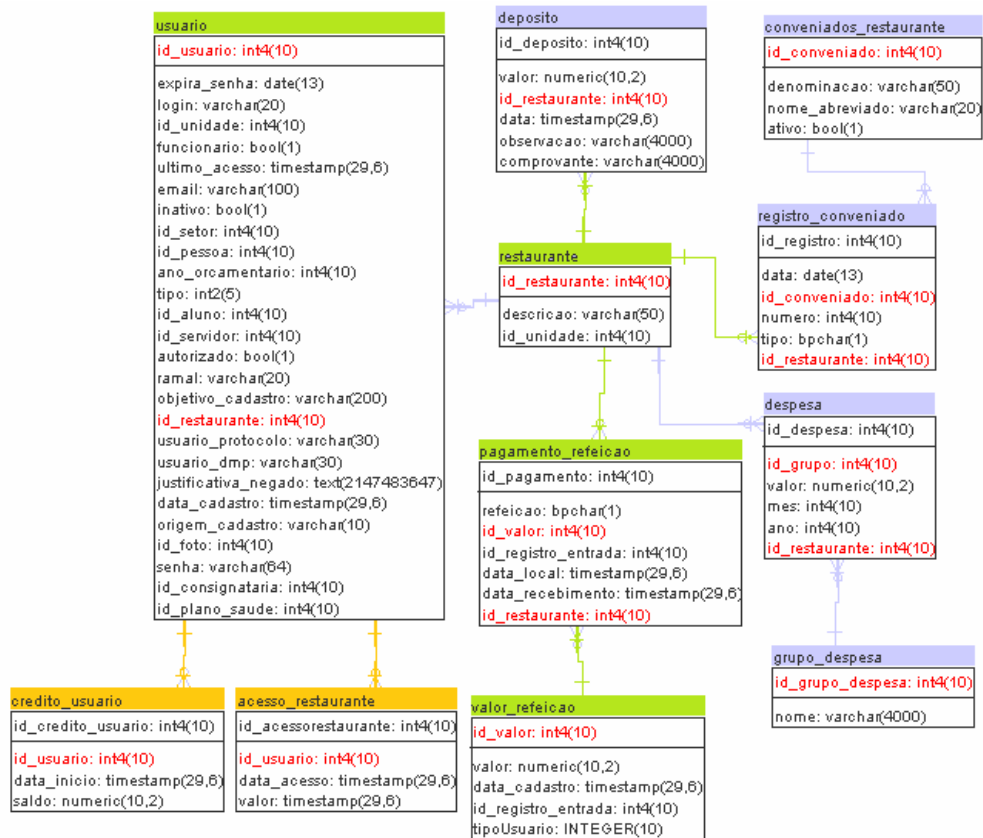
#### 4.4. Diagrama de Entidade e Relacionamento do Modulo Restaurante

No modelo de dados do banco original do sistema SIPAC, não existiam as estruturas de tabelas necessárias para armazenar dados importantes necessários para implementação das novas funcionalidades para utilização da catraca no restaurante, necessitando, dessa forma, que alterações nesse modelo fossem realizadas.

Foi feita uma análise completa do banco existente com base nos casos de uso e nas entrevistas com os usuários do sistema. Após o levantamento foi confirmada a necessidade de alterações no esquema restaurante do banco administrativo. As modificações visavam armazenar informações sobre créditos dos usuários e para essa finalidade foram incluídas duas nova da tabelas: UsuarioRefeicoes e usuario\_entradaCredito.

A tabela UsuarioRefeicoes armazenará o saldo corrente de um usuário e informações referentes a data de validade dos créditos, essa data poderá ser usada na criação de um histórico que poderá ser usado em relatórios.

A Tabela usuario\_entradaCredito irá armazenar informações sobre quantidade tickets comprados por um usuário e também salvará as datas das compras, isso também ajudará na criação de futuros relatórios.



**Figura 18: DER esquema restaurante.**

A Figura 18 mostra o DER do esquema restaurante, onde pode-se identificar pelas cores o relacionamento das novas tabelas com as existentes. As tabelas verdes e Laranjas terão uso direto nas novas funcionalidades as roxas não terão uso direto nas novas funcionalidades desenvolvidas, porém são utilizadas por outras telas do módulo de restaurante, além disso as tabelas verdes e roxas já existem no banco de dados enquanto as na cor laranja precisaram ser criadas. O relacionamento entre a tabela Usuário com as tabelas credito\_usuario é usado para fazer a associação de uma conta que armazenará os créditos comprados a um usuário cadastrado. A tabela Usuario também se relaciona com a tabela acesso\_usuario esse relacionamento irá guardar informações sobre a quantidade de acessos de um usuário ao restaurante. O Relacionamento entre as tabela pagamento\_refeicao e valor\_refeicao é usando para armazenar informações sobre a compra de refeições e pagamento\_refeicao se relaciona com restaurante indicando o restaurante onde foi realizada a compra.

## **5. CONCLUSÃO**

Grandes instituições sempre precisam de sistemas que se adéquem ao seu negócio e as melhorias contínuas requisitadas por seus usuários. O novo sistema obtido pela UFS tem como objetivo integrar todas as áreas e departamentos da instituição. O objetivo desse trabalho foi a implantação de um desses módulos, salvando as informações que já existiam e fazendo com que os dados já existentes não fossem perdidos.

Houve muitas dificuldades e alterações nos requisitos, como a troca dos módulos a serem implantados, mas mesmo com as dificuldades encontradas durante o este trabalho, o desenvolvimento desse projeto trouxe muitos ganhos em conhecimento para a equipe, pois foi necessário o estudo e a aplicação de vários conceitos abordados durante o curso, além do aprendizado de novas tecnologias e ferramentas. Esse conhecimento prático é importante na vida profissional de qualquer aluno e mostra a importância da disciplina de estágio supervisionado para uma melhor formação dos alunos.

## 6. REFERENCIAS BIBLIOGRAFICAS

- [1] Talend Open Studio. Disponível em <http://www.talendforge.org>. Último acesso em Junho de 2010.
- [2] Postgres SQL. Disponível em <http://www.postgresql.org.br/>. Último acesso em Junho de 2010.
- [3] World Wide Web. Disponível em [http://pt.wikipedia.org/wiki/World\\_Wide\\_Web](http://pt.wikipedia.org/wiki/World_Wide_Web). Último acesso em Junho de 2010.
- [4] MENDES, Antonio. Arquitetura de Software: desenvolvimento orientado para arquitetura. Editora Campus. Rio de Janeiro - RJ, 2002.
- [5] PAIM, R., CAULLIRAUX, H., CARDOSO, V. & CLEMENTE, R. Gestão de Processos: pensar, agir e aprender. Bookman., 2009.
- [6] HEUSER, CARLOS ALBERTO. Projeto de Banco de Dados. SAGRA-LUZZATTO. 4ª edição, 1998.
- [7] Java Server Faces. Disponível em <http://www.oracle.com/technetwork/java/javaee/>. Último acesso em Junho de 2010.
- [8] Jboss Application Server. Disponível em <http://www.jboss.org/>. Último acesso em Março de 2010.
- [9] KIMBALL, Ross. The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling (Second Edition). Wiley. 2002.
- [10] JAVA. Disponível em <http://www.jcp.org/en/home/index>. Último acesso em Junho de 2010.
- [11] BPMN. Disponível em <http://www.bpmn.org/>. Último acesso em Junho 2010.
- [12] BizAgi. Disponível em <http://www.bizagi.com/>. Último acesso em Junho de 2010.

[13] MASSONI, Tiago; SAMPAIO, Augusto; BORBA, Paulo. *A RUP-Based Software Process Supporting Progressive Implementation*. Idea Group Publishing, 2003.

[14] KRUCHTEN, Philippe. *The Rational Unified Process: An Introduction*. Addison-Wesley, 2003.