



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

RENATO GRACIANO MAZARIN

SISTEMA DE GERENCIAMENTO PARA CEMITÉRIOS - SISCEM

Assis
2011

RENATO GRACIANO MAZARIN

SISTEMA DE GERENCIAMENTO PARA CEMITÉRIOS - SISCEM

Trabalho de Conclusão de Curso apresentado
ao Instituto Municipal de Ensino Superior de
Assis, como requisito do Curso Superior de
Tecnologia em Processamento de Dados.

Orientador: Prof. Esp. Célio Desiró

Área de Concentração: Desenvolvimento de Sistemas

Assis
2011

FICHA CATALOGRÁFICA

MAZARIN, Renato Graciano

Sistema de Gerenciamento para Cemitérios - SisCem / Renato Graciano Mazarin. Fundação Educacional do Município de Assis - FEMA - Assis, 2011.
58p.

Orientador: Célio Desiró.

Trabalho de Conclusão de Curso - Instituto Municipal de Ensino Superior de Assis - IMESA.

1. Gerenciamento de Cemitério. 2. Cemitério Municipal.

CDD: 001.61
Biblioteca da FEMA

SISTEMA DE GERENCIAMENTO PARA CEMITÉRIOS - SISCEM

RENATO GRACIANO MAZARIN

Trabalho de Conclusão de Curso apresentado
ao Instituto Municipal de Ensino Superior de
Assis, como requisito do Curso Superior de
Tecnologia em Processamento de Dados,
analisado pela seguinte comissão examinadora:

Orientador: Prof. Esp. Célio Desiró

Analisador (1): Prof. Msc. Douglas Sanches da Cunha

Assis
2011

DEDICATÓRIA

À minha família pelo esforço, dedicação e compreensão em todos os momentos desta e de outras caminhadas.

Aos colegas do curso de Tecnologia em Processamento de Dados que demonstraram companheirismo durante a árdua caminhada no decorrer desse curso.

Em especial, ao meu pai e melhor amigo "Mazarin" e minha mãe e melhor amiga "Toninha" que acreditaram e continuam acreditando em mim. Pai, Mãe, a sua benção! Tenham sempre a minha eterna gratidão. Mãe, saudades!!

AGRADECIMENTOS

Agradeço primeiramente a Deus, pois sem Ele, nada seria possível e não estaríamos aqui reunidos, desfrutando, juntos, destes momentos que nos são tão importantes.

Aos professores do curso de Tecnologia em Processamento de Dados da FEMA pela paciência e dedicação ao longo desses três anos.

Ao Prof. Célio, pela paciência, pelo apoio e pelo acompanhamento desse trabalho.

Ao Prof. Douglas (o Dodô), pelo papo amigo e pelas aulas de VB que muito contribuíram para a minha formação acadêmica e profissional.

Ao Prof. Almir, pelas longas horas de ensinamento da ferramenta utilizada para o desenvolvimento desse trabalho.

Agradecimento especial à Prof.^a Marisa Atsuko Nito pela calorosa recepção no 1º ano depois de longos 12 anos de afastamento; ainda ecoa em meu pensamento aquele “Que bom que você voltou!”.

RESUMO

O presente estudo analisou um projeto de informatização dos serviços prestados pelo Cemitério Municipal de Assis, tomando, em primeiro plano, a face administrativa como parâmetro de inicialização dos trabalhos com implementação de um sistema gerencial em formato WEB. Este trabalho foi desenvolvido em conjunto com os profissionais que atuam junto ao Cemitério Municipal que, através de entrevistas, repassaram as necessidades daquela instituição e que depois foram transformadas em análise e implementação, gerando assim o SisCem – Sistema de Gerenciamento para Cemitérios.

Palavras-chave: SisCem, Gerenciamento de Cemitérios, Plataforma .NET, UML, Crystal Reports.

ABSTRACT

The present study examined a project to computerize the services provided by the Municipal Cemetery of Assis, whilst in the foreground, the administrative side as a start parameter of the work with implementing a management system in WEB format. This work was developed in conjunction with the professionals who work with the Municipal Cemetery, which, through interviews, passed along the needs of this institution and which have been transformed into the analysis and implementation to generate the SisCem - Management System for Cemeteries.

Keywords: SisCem, Management of Cemeteries, .NET Platform, UML, Crystal Reports.

LISTA DE ILUSTRAÇÕES

Figura 1. Principais Componentes do SQL Server 2005.....	33
Figura 2. Diagrama de Use Case Geral	39
Figura 3. Diagrama de Use Case – Ator Operador	39
Figura 4. Evento Cadastrar Usuário	40
Figura 5. Evento Manter Usuário.....	41
Figura 6. Evento Cadastrar Permissão	42
Figura 7. Evento Manter Permissão	43
Figura 8. Evento Cadastrar Evento	44
Figura 9. Evento Manter Evento.....	45
Figura 10. Evento Cadastrar Funerária.....	46
Figura 11. Evento Manter Funerária.....	47
Figura 12. Evento Cadastrar Responsável pelo Atestado de Óbito	48
Figura 13. Evento Manter Responsável pelo Atestado de Óbito	49
Figura 14. Evento Emitir Relatórios.....	50
Figura 15. Evento Reservar Túmulo.....	51
Figura 16. Evento Cadastrar Sepultamento	52
Figura 17. Evento Cadastrar Falecido.....	53
Figura 18. Evento Cadastrar Óbito.....	54
Figura 19. Evento Cadastrar Túmulo	55
Figura 20. Diagrama de Classe.....	56

LISTA DE TABELAS

Tabela 1. Tabela comparativa entre edições do Crystal Reports – Report Design ...	31
Tabela 2. Tabela comparativa entre edições do Crystal Reports – Application Development	32
Tabela 3. Lista de eventos	38

SUMÁRIO

1 - INTRODUÇÃO	13
2 - OBJETIVOS	14
3 - JUSTIFICATIVA	15
4 - MOTIVAÇÃO.....	16
5 - PERSPECTIVAS DE CONTRIBUIÇÃO	17
6 – METODOLOGIA DE TRABALHO	18
7 – FERRAMENTAS UTILIZADAS.....	19
7.1 – Tecnologia .NET.....	19
7.2 – UML - Unified Modeling Language (<i>Linguagem de Modelagem Unificada</i>).....	20
Elementos UML.....	21
Diagrama de Caso de Uso	21
Caso de Uso.....	21
Ator.....	22
Descrição do Caso de Uso.....	22
Diagrama de Classe	23
Classe	23
Atributos	23
Operações.....	23
Modelos.....	24
Associações de Classe	24
Generalização	24
Associações	24
Agregação	25
Composição	25
Outros Ítems do Diagrama de Classe	25
Interfaces.....	25
Tipos de dados.....	26
Enumerações	26
Pacotes	26
Diagramas de Seqüência	26
Diagramas de Colaboração.....	27

Diagrama de Estado.....	27
Estado	28
Diagrama de Atividade	28
Atividade.....	28
Elementos Auxiliares	29
Diagramas de Componente.....	29
Diagramas de Distribuição	30
7.3 – Gerador de relatórios Crystal Reports	30
7.4 – Banco de dados SQL Server 2005	32
8 – DESENVOLVIMENTO DO PROJETO	34
8.1 – Análise de Requisitos	34
Reserva de túmulos (venda) com a informação dos seguintes dados	34
Dados do cliente.....	34
Cadastro de sepultamentos com a informação dos seguintes dados	35
Dados do sepultamento	35
Dados do Túmulo	35
Dados do falecido.....	36
Dados do óbito	36
Cadastros Operacionais com a informação dos seguintes dados.....	36
Dados do Responsável pelo Atestado de Óbito	36
Dados da Funerária.....	37
8.2 – Lista de Eventos	37
8.3 – Diagramas de Use Case	38
8.4 – Diagrama de Classe	56
CONSIDERAÇÕES FINAIS	57
REFERÊNCIAS BIBLIOGRÁFICAS.....	58

1. INTRODUÇÃO

Com processos mais ágeis, eficientes e totalmente voltados a resultados, o setor privado começa servir de espelho à área pública. Na busca por identificar quais soluções apóiam a gestão das empresas, o governo se deparou com os sistemas de gestão empresarial (ERPs) e surgiu o conceito de GRP, ou Government Resource Planning. (<http://www.decisionreport.com.br>)

O sistema prevê a interligação das diversas áreas do governo, permitindo ao gestor ter uma visão mais global. A única forma de melhorar os serviços ao cidadão é integrar o back office e o GRP cumpre essa função.

Com essa visão, pretende-se desenvolver um sistema, voltado à administração pública, de Gerenciamento de Cemitérios, aqui chamado de SisCem.

A informatização do sistema administrativo do Cemitério permitirá agilidade na operacionalização de serviços, facilitando busca e emissão de documentos. O programa disporá de cadastro de funcionários, perpetuadores e responsáveis pelas sepulturas e falecidos. Além disso, fornecerá registros da disponibilidade de vagas nos cemitérios, reutilização de covas e emissão de documentos.

O sistema será desenvolvido sob a plataforma .NET, mais precisamente com a ferramenta “Microsoft Visual Web Developer 2008”. Trata-se de uma ferramenta moderna, que proporciona uma interface fácil e agradável ao usuário devido à amplitude de recursos gráficos. Para armazenamento das informações, o aplicativo usado será o banco de dados SQL Server 2005, que acompanha o pacote de instalação do Microsoft Visual Studio Team System 2008, o qual se caracteriza como uma interface simples, objetiva, atendendo com total eficiência e exatidão a implementação do sistema. Para gerar os relatórios será utilizada a ferramenta Crystal Reports que gera relatórios eficientes e precisos, além de oferecer uma boa visualização dos dados desejados.

2. OBJETIVOS

O objetivo deste trabalho de conclusão de curso é criar um sistema de gerenciamento para propiciar à administração do Cemitério Municipal de Assis um melhor controle das informações geradas em todo seu âmbito gerencial e administrativo. A proposta principal é, além de colaborar com a administração, proporcionar uma melhor interface de comunicação entre a gerência e os usuários na solução de problemas ligados ao fornecimento de informações, sejam elas quais forem.

3. JUSTIFICATIVA

A administração do Cemitério Municipal de Assis solicitou um sistema para auxiliar nas tarefas de rotina e administrativas de um cemitério público. Com a informatização dos serviços pretende-se aprimorar a interação da administração com os usuários, em sua maioria visitantes, fornecendo um serviço mais pessoal e organizado.

4. MOTIVAÇÃO

Com o advento da internet a procura por interatividade de serviços tem sido intensa e a oferta aquém das necessidades. O Cemitério Municipal de Assis não conta com um sistema de gerenciamento via web, o que certamente contribuiria para um melhor atendimento à população que utiliza seus serviços tanto informativos quanto profissionais.

5. PERSPECTIVAS DE CONTRIBUIÇÃO

O sistema visa uma melhor organização das informações para que todos os envolvidos obtenham mais informações e que as mesmas sejam detalhadas, facilitando o dia a dia.

Desenvolvendo-se o SisCem, criar-se-á uma nova perspectiva administrativa, uma vez que o Cemitério Municipal de Assis ainda não conta com um sistema informatizado de Administração.

Com o SisCem, espera-se contribuir com o gerenciamento do órgão municipal, com o fornecimento de controles detalhados e precisos, além de relatórios eficientes e de fácil visualização, para que tarefas rotineiras tornem-se menos cansativas e com menor probabilidade de erros, além de proporcionar maior praticidade no tratamento dos serviços.

6. METODOLOGIA DE TRABALHO

Para a elaboração deste trabalho será consultada toda a gama de informações disponíveis, através de livros, sites especializados e tutoriais referentes aos softwares a serem utilizados para a elaboração do sistema.

Na primeira etapa será realizada entrevista para levantamento de todas as necessidades do cliente, onde serão sanadas as dúvidas prementes. Na segunda etapa será desenvolvido o software com base na documentação elaborada na primeira etapa.

7. FERRAMENTAS UTILIZADAS

7.1. Tecnologia .NET - Segundo o site <http://www.comquest.com.br/net.asp>

A tecnologia .NET da Microsoft está revolucionando os rumos da Internet e da Computação Distribuída. Essa tecnologia alcança dimensões nunca antes imaginadas para seus aplicativos.

O que é .NET

.NET é uma plataforma de software que conecta informações, sistemas, pessoas e dispositivos. A plataforma .NET conecta uma grande variedade de tecnologias de uso pessoal e de negócios, de telefones celulares a servidores corporativos, permitindo o acesso a informações importantes, onde e sempre que forem necessárias.

Desenvolvido sobre os padrões de Web Services XML, .NET possibilita que sistemas e aplicativos, novos ou já existentes, conectem seus dados e transações independentes do sistema operacional, tipo de computador ou de dispositivo móvel que sejam utilizados, ou que linguagem de programação tenha sido utilizada na sua criação.

O .NET é um "ingrediente" presente em toda a linha de produtos Microsoft, oferecendo a capacidade de desenvolver, implementar, gerenciar e usar soluções conectadas através de Web Services XML, de maneira rápida, barata e segura. Essas soluções permitem uma integração mais rápida e ágil entre os negócios e o acesso a informações a qualquer hora, em qualquer lugar e em qualquer dispositivo.

A idéia fundamental por trás do Microsoft .NET é uma mudança de foco na informática, passando de um mundo de aplicativos, Web sites e dispositivos isolados para uma infinidade de computadores, dispositivos, transações e serviços que se conectam diretamente e trabalham em conjunto para fornecerem soluções mais amplas e ricas.

As pessoas terão o controle sobre como, quando e que informações serão fornecidas a elas. Os computadores, sistemas e serviços

serão capazes de colaborar e interoperar mutuamente em favor do usuário, e as empresas poderão oferecer seus produtos e serviços aos clientes certos, na hora certa, da forma certa, combinando processos de maneira muito mais granular do que é possível hoje.

7.2. UML - Unified Modeling Language (*Linguagem de Modelagem Unificada*)

A Linguagem de Modelagem Unificada (Unified Modelling Language - UML) é uma linguagem de diagramação ou notação para especificar, visualizar e documentar modelos de sistemas de software Orientados à Objeto. A UML não é um método de desenvolvimento, o que significa que ela não diz para você o que fazer primeiro e em seguida ou como desenhar seu sistema, mas ele lhe auxilia a visualizar seu desenho e a comunicação entre objetos. A UML é controlada pelo Grupo de Gerenciamento de Objeto (Object Management Group - OMG) e é um padrão da indústria para descrever graficamente software.

A UML é voltada para o desenho de software Orientado à Objeto e tem um uso limitado para outros paradigmas de programação.

A UML é composta por muitos elementos de modelo que representam as diferentes partes de um sistema de software. Os elementos UML são usados para criar diagramas, que representam uma determinada parte, ou um ponto de vista do sistema.

- Diagrama de Caso de Uso mostra atores (pessoas ou outros usuários do sistema), casos de uso (os cenários onde eles usam o sistema), e seus relacionamentos
- Diagrama de Classe mostra classes e os relacionamentos entre elas
- Diagrama de Seqüência mostra objetos e uma seqüência das chamadas do método feitas para outros objetos.
- Diagrama de Colaboração mostra objetos e seus relacionamentos, colocando ênfase nos objetos que participam na troca de mensagens
- Diagrama de Estado mostra estados, mudanças de estado e eventos num objeto ou uma parte do sistema

- Diagrama de Atividade mostra atividades e as mudanças de uma atividade para outra com os eventos ocorridos em alguma parte do sistema
- Diagrama de Componente mostra os componentes de programação de alto nível (como KParts ou Java Beans).
- Diagrama de Distribuição mostra as instâncias dos componentes e seus relacionamentos.

Elementos UML

Diagrama de Caso de Uso

Diagramas de Caso de Uso descrevem relacionamentos e dependências entre um grupo de Caso de Uso e os Atores participantes no processo.

É importante observar que Diagramas de Caso de Uso não são adequados para representar o desenho, e não podem descrever os mecanismos internos de um sistema. Diagramas de Caso de Uso são feitos para facilitar a comunicação com os futuros usuários do sistema, e com o cliente, e são especialmente úteis para determinar os recursos necessários que o sistema deve ter. Diagramas de Caso de Uso dizem o quê o sistema deve fazer, mas não fazem — e não podem — especificar como isto será conseguido.

Caso de Uso

Um Caso de Uso descreve — do ponto de vista dos atores — um grupo de atividades num sistema que produz um resultado concreto e tangível.

Casos de Uso são descrições de interações típicas entre os usuários de um sistema e o sistema propriamente dito. Eles representam a interface externa do sistema e especificam um conjunto de exigências do que o sistema deve fazer (lembre-se: somente o quê, não como).

Quando trabalhar com Casos de Uso, é importante lembrar-se de algumas regras simples:

- Cada Caso de Uso está relacionado com no mínimo um ator
- Cada Caso de Uso possui um iniciador (ator)

- Cada Caso de Uso liga-se a um resultado relevante (um resultado com “valor de negócio”)

Casos de Uso também podem ter relacionamentos com outros Casos de Uso. Os três tipos mais comuns de relacionamento entre Casos de Uso são:

- <<inclui-se>> que especifica que um Caso de Uso toma lugar dentro de outro Caso de Uso
- <<estende>> que especifica que em determinadas situações, ou em algum ponto (chamado um ponto de extensão) um Caso de Uso será estendido por outro.
- Generalização especifica que um Caso de Uso herda as características do “Super” Caso de Uso, e pode sobrepor algumas delas ou adicionar novas de maneira semelhante à herança entre classes.

Ator

Um ator é uma entidade externa (fora do sistema) que interage com o sistema participando (e freqüentemente iniciando) um Caso de Uso. Atores podem ser pessoas reais (por exemplo usuários do sistema), outro sistema de computador ou eventos externos.

Atores não representam as pessoas físicas ou sistemas, mas sua regra. Isto significa que quando uma pessoa interage com o sistema de diferentes maneiras (assumindo diferentes regras) ela será representada por diversos atores. Por exemplo, uma pessoa que fornece suporte ao cliente por telefone e recebe ordens do cliente para o sistema pode ser representado por um ator da “Equipe de Suporte” e um ator “Representante de Vendas”

Descrição do Caso de Uso

Descrição do Caso de Uso são narrativas de texto do Caso de Uso. Elas usualmente tomam a forma de uma nota ou um documento que é de alguma maneira ligado ao Caso de Uso, e explana o processo ou atividades que tomarão lugar no Caso de Uso.

Diagrama de Classe

Diagramas de Classe mostram as diferentes classes que fazem um sistema e como elas se relacionam. Os Diagramas de Classe são chamados diagramas “estáticos” porque mostram as classes, com seus métodos e atributos bem como os relacionamentos estáticos entre elas: quais classes “conhecem” quais classes ou quais classes “são partes” de outras classes, mas não mostram a troca de mensagens entre elas.

Classe

Uma Classe define os atributos e os métodos de um conjunto de objetos. Todos os objetos desta classe (instâncias desta classe) compartilham o mesmo comportamento, e possuem o mesmo conjunto de atributos (cada objeto possui seu próprio conjunto). O termo “Tipo” é algumas vezes usado ao invés de Classe, mas é importante mencionar que estes dois termos não são a mesma coisa, e Tipo é um termo mais genérico.

Em UML Classes são representadas por retângulos, com o nome da classe, e podem também mostrar os atributos e operações da classe em dois outros “compartimentos” dentro do retângulo.

Atributos

Na UML, atributos são mostrados com pelo menos seu nome, e podem também mostrar seu tipo, valor inicial e outras propriedades. Atributos podem também ser exibidos com sua visibilidade:

- + indica atributos públicos
- # indica atributos protegidos
- - indica atributos privados

Operações

Operações (métodos) também são exibidas com pelo menos seu nome, e podem também mostrar seus parâmetros e valores de retorno. Operações podem, como os Atributos, mostrar sua visibilidade:

- + indica operações públicas
- # indica operações protegidas

- - indica operações privadas

Modelos

Classes podem ter modelos, um valor que é usado para uma classe ou tipo não especificado. O tipo de modelo é especificado quando uma classe é iniciada (isto é um objeto é criado). Modelos existem no C++ moderno e foram introduzidos no Java 1.5 onde eles são chamados de Genéricos.

Associações de Classe

Classes podem relacionar-se (ser associada com) com outras de diferentes maneiras:

Generalização

Herança é um dos conceitos fundamentais da programação Orientada à Objeto, na qual uma classe “herda” todos os atributos e operações da classe da qual deriva, e pode sobrescrever/modificar alguns deles, bem como adicionar mais atributos e operações próprios.

EM UML, uma associação Generalização entre duas classes coloca-as numa hierarquia representando o conceito de herança de uma classe derivada de uma classe base. Em UML, Generalizações são representadas por uma linha conectando duas classes, com uma seta no lado da classe base.

Associações

Uma associação representa um relacionamento entre classes, e fornece a semântica comum e a estrutura para muitos tipos de “conexões” entre objetos.

Associações são mecanismos que permitem objetos se comunicarem entre si. Elas descrevem a conexão entre diferentes classes (a conexão entre os objetos atuais é chamada conexão do objeto, ou link).

Associações podem ter uma regra que especifica o propósito da associação e pode ser uni ou bidirecional (indicando se os dois objetos participantes do relacionamento podem mandar mensagens para o outro, ou se apenas um deles sabe sobre o outro). Cada ponta da associação também possui um valor de multiplicidade, que dita como muitos objetos, neste lado da associação, podem relacionar-se com o outro lado.

Em UML, associações são representadas como linhas conectando as classes participantes do relacionamento, e podem também mostrar a regra e a multiplicidade de cada um dos participantes. A multiplicidade é exibida como um intervalo [min...máx] de valores não negativos, com uma estrela (*) no lado máximo representando infinito.

Agregação

Agregações são um tipo especial de associação no qual as duas classes participantes não possuem em nível igual, mas fazem um relacionamento “todo-parte”. Uma Agregação descreve como a classe que possui a regra do todo, é composta (tem) de outras classes, que possuem a regra das partes. Para Agregações, a classe que age como o todo sempre tem uma multiplicidade de um.

Em UML, Agregações são representadas por uma associação que mostra um rombóide no lado do todo.

Composição

Composições são associações que representam agregações muito fortes. Isto significa que Composições formam relacionamentos todo-parte também, mas o relacionamento é tão forte que as partes não podem existir independentes. Elas existem somente dentro do todo, e se o todo é destruído as partes morrem também.

Em UML, Composições são representadas por um rombóide sólido no lado do todo.

Outros Itens do Diagrama de Classe

Diagramas de Classe podem conter diversos outros itens além das classes.

Interfaces

Interfaces são classes abstratas que significam instâncias que não podem ser diretamente criadas delas. Elas podem conter operações mas não podem conter atributos. Classes podem derivar de interfaces (através da realização de uma associação) e instâncias podem então ser feitas destes diagramas.

Tipos de dados

Tipos de dados são primitivos uma vez que são tipicamente construídos numa linguagem de programação. Exemplos comuns são inteiros e lógicos. Eles não podem ser relacionados à classes, mas classes pode se relacionar com eles.

Enumerações

Enumerações são uma lista simples de valores. Um exemplo típico é uma enumeração para dias da semana. As opções de uma enumeração são chamadas Literais de Enumeração. Como tipos de dados, elas não podem ter relacionamentos para classes mas classes podem relacionar-se com elas.

Pacotes

Pacotes representam um espaço de nomes numa linguagem de programação. Num diagrama eles são usados para representar partes de um sistema que contém mais de uma classe, talvez centenas de classes.

Diagramas de Seqüência

Diagramas de Seqüência mostram a troca de mensagens (isto é chamada de método) entre diversos Objetos, numa situação específica e delimitada no tempo. Objetos são instâncias de classes. Diagramas de Seqüência colocam ênfase especial na ordem e nos momentos nos quais mensagens para os objetos são enviadas.

Em Diagramas de Seqüência objetos são representados através de linhas verticais tracejadas, com o nome do Objeto no topo. O eixo do tempo é também vertical, aumentando para baixo, de modo que as mensagens são enviadas de um Objeto para outro na forma de setas com a operação e os nomes dos parâmetros.

Mensagens pode ser síncronas, o tipo normal de mensagem de chamada onde o controle é passado para o objeto chamado até o método ter terminado sua execução, ou assíncronas onde o controle é passado diretamente para o objeto chamado. Mensagens síncronas possui uma caixa vertical no lado do objeto chamado para mostrar o controle do fluxo do programa.

Diagramas de Colaboração

Diagramas de Colaboração mostram as interações que ocorrem entre os objetos participantes numa situação específica. Isto é mais ou menos a mesma informação mostrada pelos Diagramas de Seqüência, mas neste a ênfase é colocada em como as interações ocorrem no tempo, enquanto os Diagramas de Colaboração colocam os relacionamentos entre os objetos e sua topologia em destaque.

Em Diagramas de Colaboração as mensagens enviadas de um objeto para outro são representadas por setas, mostrando o nome da mensagem, parâmetros, e a seqüência da mensagem. Diagramas de Colaboração são especialmente indicados para mostrar um fluxo ou situação específica do programa e são um dos melhores tipos de diagrama para rapidamente demonstrar ou explicar um processo na lógica do programa.

Diagrama de Estado

Diagramas de Estado mostram os diferentes estados de um Objeto durante sua vida, e o estímulo que faz com que o Objeto mude seu estado.

Diagramas de Estado vêem Objetos como máquinas de estado ou automatismos finitos que podem ser um de um conjunto de estados finitos e que podem mudar seu estado através de um de um conjunto finito de estímulos. Por exemplo, um tipo de Objeto ServidorRede pode estar em um dos seguintes estados durante sua vida:

- Pronto
- Ouvindo
- Trabalhando
- Parado

E os eventos que podem fazer com que o Objeto mude de estado são:

- Objeto é criado
- Objeto recebe mensagem ouvir
- Um Cliente solicita uma conexão através da rede
- Um Cliente termina um pedido
- O pedido é executado e terminado

- Objeto recebe mensagem parar
- etc.

Estado

Estados são os blocos construídos dos Diagramas de Estado. Um Estado pertence a exatamente uma classe e representa um resumo dos valores dos atributos que uma classe pode tomar. Um Estado UML descreve o estado interno de um objeto para uma classe em particular

Observe que nem toda mudança em um dos atributos de um objeto pode ser representada por um Estado mas somente aquelas mudanças que podem afetar significativamente o trabalho do objeto

Existem dois tipos especiais de Estados: Inicial e Final. Eles são especiais porque nenhum evento pode fazer com que um Objeto retorne para seu estado Inicial, e da mesma maneira nenhum evento pode tirar um Objeto de seu estado Final uma vez que ele já o tenha alcançado.

Diagrama de Atividade

O Diagrama de Atividade descreve a seqüência de atividades num sistema com a ajuda as Atividades. Diagramas de Atividades são formas especiais de Diagramas de Estado, que somente (ou principalmente) contém Atividades.

Diagramas de Atividade são similares as Diagramas de Fluxo de procedimentos, com a diferença de que todas as Atividades são claramente anexas aos Objetos.

Diagramas de Atividade são sempre associados a uma Classe, uma Operação ou um Caso de Uso.

Diagramas de Atividade suportam Atividades seqüenciais bem como paralelas. A execução paralela é representada pelos ícones Forquilha/Esperar, e para as Atividades executadas em paralelo, não é importante a ordem na qual elas se executam (elas podem ser executadas ao mesmo tempo ou uma após a outra).

Atividade

Uma Atividade é um passo simples num processo. Uma Atividade é um estado no sistema com atividade interna e, pelo menos, uma transição de saída. Atividades

podem também ter mais de uma transição de saída se elas possuem condições diferentes.

Atividades podem formar hierarquias, isto significa que uma Atividade pode ser composta por diversas Atividades em “detalhe”, na qual as transições de entrada e saída devem corresponder às transições de entrada e saída do diagrama de detalhe.

Elementos Auxiliares

Existem dois elementos em UML que não possuem nenhum valor real semântico para o modelo, mas auxiliam a elucidar partes do diagrama. Estes elementos são

- Linhas de texto
- Notas de Texto e âncoras
- Caixas

Linhas de texto são úteis para adicionar informações curtas de texto ao diagrama. São textos livres e não possuem nenhum significado para o Modelo propriamente dito.

Notas são úteis para adicionar informações mais detalhadas sobre um objeto ou situação específica. Elas possuem a grande vantagem de poderem ser ancoradas a Elementos UML para mostrar que a nota “pertence” a um objeto específico ou situação.

Caixas são retângulos de forma livre que podem ser usados para agrupar itens tornando os diagramas mais legíveis. Eles não possuem nenhum significado lógico no modelo.

Diagramas de Componente

Diagramas de Componente mostram os componentes do software (sejam componentes de tecnologias como KParts, componentes CORBA ou Java Beans ou apenas seções do sistema que são claramente distintas) e os artefatos de que eles são feitos como arquivos de código fonte, bibliotecas de programação ou tabelas de bancos de dados relacionais.

Componentes podem possuir interfaces (isto é classes abstratas com operações) que permitem associações entre componentes.

Diagramas de Distribuição

Diagramas de distribuição mostram as instâncias dos componentes de tempo de execução e suas associações. Eles incluem NÓS que são recursos físicos, tipicamente um computador simples. Eles também mostram interfaces e objetos (instâncias da classe).

7.3. GERADOR DE RELATÓRIOS CRYSTAL REPORTS

O **Crystal Reports** é a ferramenta para geração de relatórios, líder mundial absoluta, projetada para aumentar a produtividade e a agilidade das informações.

Acessando mais de 30 bancos de dados diferentes, nativamente ou através de drivers de conectividade, você será capaz de fornecer informações, números e gráficos avançados, imprimir relatórios com alta qualidade e distribuí-los em diversos formatos, (Word, Excel, web, etc.), inclusive para usuários que não têm o **Crystal Reports** instalado em seus computadores.

Isso tudo através de vários assistentes capazes de ajudá-lo nos passos básicos da montagem de um relatório: mala-direta, etiquetas, sub-relatórios e relatórios com análise de dados, além de outros, proporcionando ganho de tempo para tomada de decisões.

Além disso, os relatórios do **Crystal Reports** podem ser incluídos em programas desenvolvidos com DataFlex e Visual DataFlex, além de outras ferramentas e linguagens.

Com Crystal Reports você pode:

- Transformar rapidamente qualquer dado em informação poderosa e interativa.
- Criar, modificar e visualizar relatórios.
- Permitir acesso e interação com os relatórios via portal, dispositivo wireless e documentos.
- Exportar os relatórios para diversos formatos.
- Integrar os relatórios dentro da sua aplicação desenvolvida em VDF ou outra linguagem.

Simplem em seus fundamentos e em sua operação, poderoso em suas finalidades. O **Crystal Reports** reúne num só produto ferramentas para desenvolvimento (interno e externo), funções e fórmulas para controle de objetos (linguagem Crystal e Basic), formatações das mais variadas (através de janelas padronizadas) e controles de imagens, gráficos e mapas, deixando seus relatórios com um visual apropriado para qualquer tipo de situação.

A tabela abaixo ilustra algumas diferenças importantes entre as várias edições do Crystal Reports:

REPORT DESIGN	CONECTIVIDADE COM FONTES DE DADOS			
		S*	P**	D***
	PC-based e Microsoft® ODBC/OLE DB para MS Access e SQL Server	•	•	•
	XML		•	•
	OLAP		•	•
	Enterprise database servers (via ODBC e nativo)		•	•
	Dados customizados, definidos pelo usuário através de JavaBeans™, ADO, .NET e COM			•
	EDIÇÃO DE RELATÓRIOS			
	Editor visual de relatórios	•	•	•
	Templates personalizáveis para formatação rápida e consistente	•	•	•
Repositório para reutilização de objetos em diversos relatórios		•	•	
ENTREGA DE RELATÓRIOS WEB				
Crystal Enterprise Express para entrega rápida de relatórios na web		•		

Tabela 1. Tabela comparativa entre edições do Crystal Reports – Report Design

APPLICATION DEVELOPMENT	INTEGRAÇÃO COM APLICAÇÕES	S*	P**	D***
	API's de visualização de relatórios (Java, .NET e COM SDKs)			•
	API's (Java, .NET e COM SDKs) para usuários criarem e modificarem relatórios em tempo de execução			
	DISTRIBUIÇÃO DE APLICAÇÕES			
	Componentes de relatórios (3) do Crystal Reports para Java, .NET e COM, para visualização, impressão e exportação de relatórios			•
	Crystal Enterprise Embedded para processamento de relatórios grandes em um web server			

(*) Standard Edition (**)Professional Edition (***)Developer Edition

Tabela 2. Tabela comparativa entre edições do Crystal Reports – Application Development

7.4. BANCO DE DADOS SQL SERVER 2005

O SQL Server 2005 é uma plataforma de base de dados abrangente que oferece ferramentas de gestão de dados de classe empresarial com integração de Business Intelligence (BI). O motor de base de dados SQL Server 2005 proporciona um armazenamento mais seguro e fiável tanto para dados relacionais como estruturados, permitindo-lhe criar e gerir aplicações de dados de elevada disponibilidade e desempenho para a sua empresa.

O motor de dados do SQL Server 2005 é o núcleo desta solução de gestão de dados empresariais. Mas o SQL Server 2005 concentra também o que existe de mais avançado em análise, reporting, integração e notificação. Isto permite à sua empresa criar e implementar soluções de BI eficientes em termos de custos que ajudam a sua equipa a transportar dados para qualquer parte da sua empresa através de scorecards, dashboards, Web services e dispositivos móveis.

A estreita integração com o Microsoft Visual Studio, o Microsoft Office System e um pacote de novas ferramentas de desenvolvimento, incluindo o Business Intelligence Development Studio, torna o SQL Server 2005 um caso à parte. Quer seja um programador, um administrador de base de dados, um trabalhador da informação ou

um decisor, o SQL Server 2005 oferece soluções inovadoras que o ajudam a extrair mais valor dos seus dados.

O esquema seguinte ilustra as principais componentes do SQL Server 2005 e demonstra como esta solução é uma peça chave do Windows Server System na integração com a plataforma Microsoft Windows – incluindo o Microsoft Office System e o Visual Studio –, possibilitando-lhe assim a distribuição de dados em qualquer ponto da sua organização.

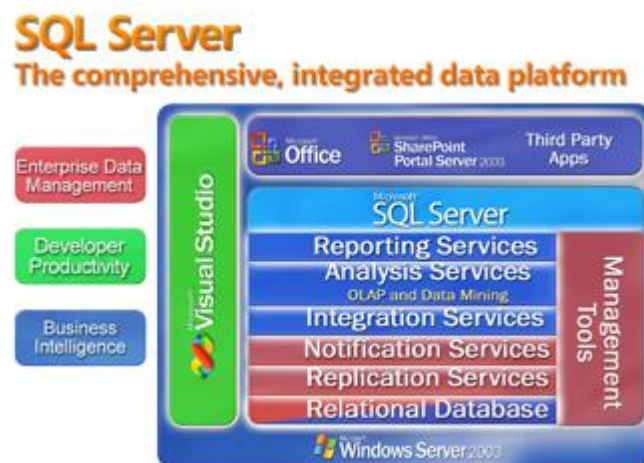


Figura 1. Principais Componentes do SQL Server 2005

8. DESENVOLVIMENTO DO PROJETO

8.1. Análise de Requisitos

Primeiramente fez-se o levantamento dos requisitos levando-se em conta, primordialmente, as necessidades apontadas pelos responsáveis pelo gerenciamento do Cemitério Municipal de Assis e que, posteriormente, passaram por análise minuciosa cabendo dessa forma ao analista/programador a crítica das necessidades reais do sistema onde se procurou contribuir para melhores entendimentos por parte do usuário final e melhor facilidade de desenvolvimento para o programador.

Inicialmente o desenvolvimento dos trabalhos contará com um ator principal, denominado administrador e um ator secundário, mas não menos importante, denominado operador. Será, então, criada uma interface de acesso ao sistema onde deverão ser informados o “Nome de Usuário” e a “Senha”.

Ao administrador serão atribuídas as seguintes funções que podem ser consideradas especiais, pois somente a ele serão permitidas tais aplicações:

- Cadastro de usuário do sistema
- Manutenção do cadastro de usuário
- Controle de permissões a usuários
- Controle de eventos
- Relatórios mais específicos

As permissões concedidas ao operador do sistema serão (estas também serão privilégio do administrador do sistema):

- Reserva de túmulos (venda) com a informação dos seguintes dados:
 - Dados do cliente:
 - Nome
 - CPF

- Endereço
 - Bairro
 - Cidade
 - CEP
 - Estado
 - Data da compra
 - Número da sepultura
- Cadastro de sepultamentos com a informação dos seguintes dados:
 - Dados do sepultamento:
 - Data de cadastro no sistema
 - Código do sepultamento (utilizando-se o número informativo de sepultura – número seqüencial fornecido pela administração do cemitério)
 - Rua (endereço interno de localização da sepultura)
 - Número da quadra de localização da sepultura
 - Número do setor de localização da sepultura
 - Número do lote de localização da sepultura
 - Terreno perpétuo
 - Dados do Túmulo
 - Rua (endereço interno de localização do túmulo)
 - Número da quadra de localização do túmulo
 - Número do setor de localização do túmulo
 - Número do lote de localização do túmulo
 - Terreno perpétuo

- Dados do falecido:
 - Nome
 - Data de nascimento
 - Idade
 - Nome do pai
 - Nome da mãe
 - Nome do cônjuge
 - Nacionalidade
 - Naturalidade (cidade e estado)

- Dados do óbito
 - Número de atestado de óbito
 - Número da folha
 - Número do livro
 - Data do óbito
 - Declarante
 - Responsável pelo Atestado de Óbito
 - *Causa mortis*
 - Serviço funerário responsável

- Cadastros Operacionais com a informação dos seguintes dados:
 - Dados do Responsável pelo Atestado de Óbito
 - Nome
 - CPF
 - Número de Registro de Classe

- Dados da Funerária
 - Razão Social
 - CNPJ
 - Endereço
 - Cidade
 - Estado
 - CEP
 - Fone
 - Ramal
 - Pessoa para Contato

8.2. Lista de Eventos

Nº	DESCRIÇÃO	EVENTO	USE CASE	RESPOSTA
01	Administrador cadastra usuário	Dados do usuário	Cadastrar usuário	Msg_01
02	Administrador mantém usuário	Dados da manutenção	Manter usuário	Msg_02
03	Administrador cadastra permissão	Dados do usuário	Cadastrar permissão	Msg_03
04	Administrador mantém permissão	Dados do usuário	Manter permissão	Msg_04
05	Administrador cadastra evento	Dados do usuário	Cadastrar evento	Msg_05
06	Administrador mantém evento	Dados do usuário	Manter evento	Msg_06

07	Administrador cadastra serviço funerário	Dados da funerária	Cadastrar funerária	Msg_07
08	Administrador mantém funerária	Dados da funerária	Manter funerária	Msg_08
09	Administrador cadastra responsável pelo atestado de óbito	Dados do responsável	Cadastrar responsável pelo Atestado de Óbito	Msg_09
10	Administrador mantém responsável pelo atestado de óbito	Dados do responsável	Manter responsável pelo Atestado de Óbito	Msg_10
11	Administrador mantém banco de dados	Dados da manutenção	Manter banco de dados	Msg_11
12	Administrador emite relatório	Dados do relatório	Emitir relatório	Msg_12
13	Operador procede reserva de túmulo	Dados do cliente	Reservar túmulo	Msg_13
14	Operador cadastra sepultamento	Dados do sepultamento	Cadastrar sepultamento	Msg_14
15	Operador cadastra falecido	Dados do falecido	Cadastrar falecido	Msg_15
16	Operador cadastra óbito	Dados do óbito	Cadastrar óbito	Msg_16

Tabela 3. Lista de eventos

8.3. Diagramas de Use Case

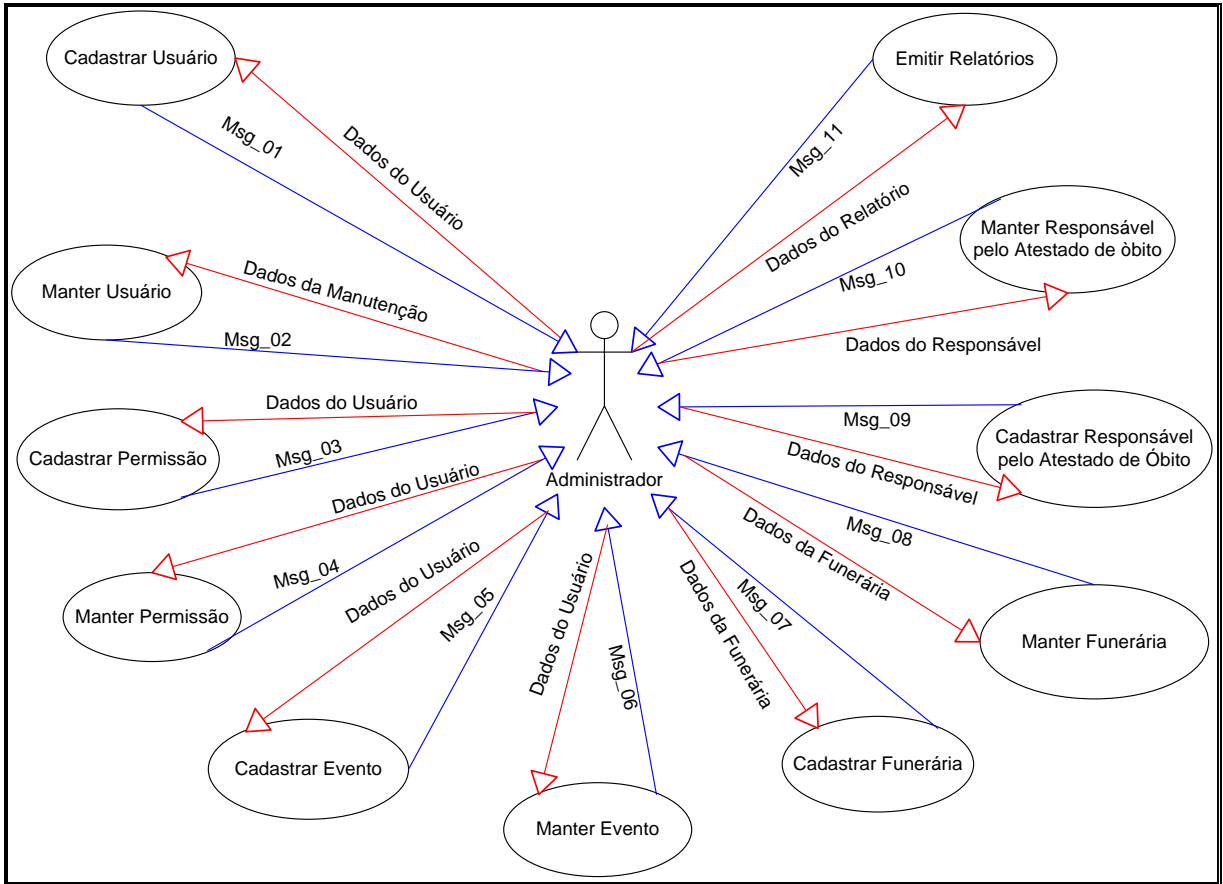


Figura 2. Diagrama de Use Case Geral

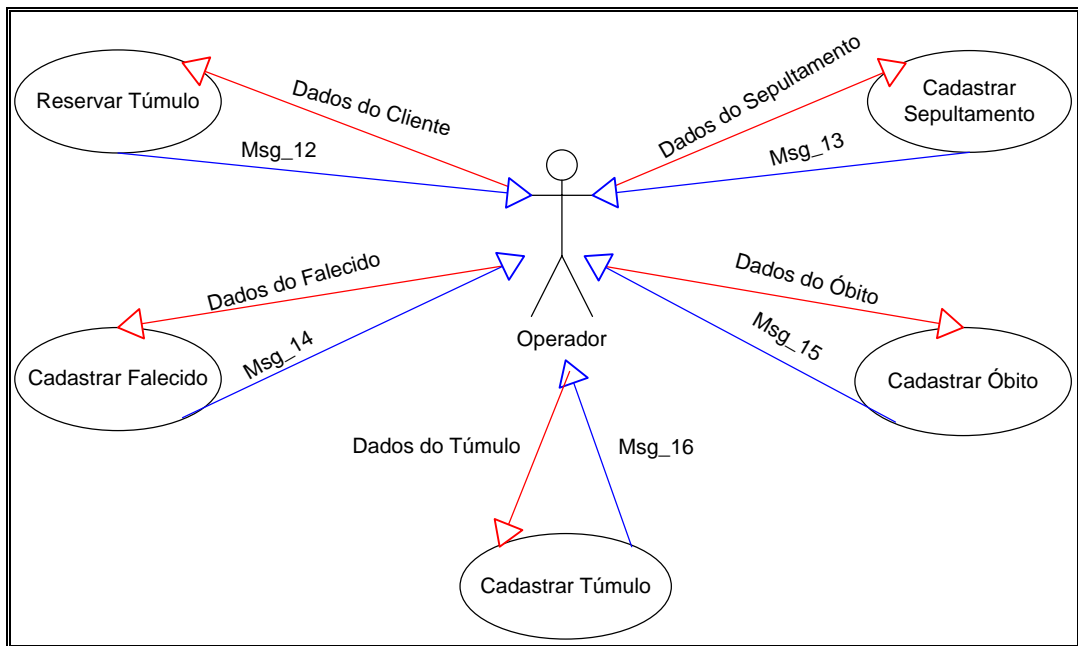
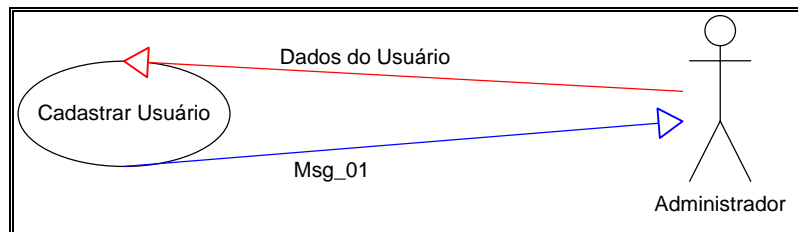


Figura 3. Diagrama de Use Case – Ator Operador

Evento 1

**Figura 4. Evento Cadastrar Usuário**

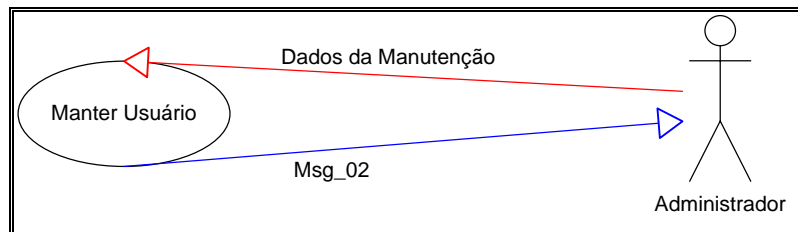
Curso Normal

- 1 – Verificar dados completos do usuário
- 2 – Preencher cadastro do usuário
- 3 – Atender solicitação do cadastro
- 4 – Verificar todos os campos do formulário de cadastro
- 5 – Msg_01 = “Cadastro efetuado com sucesso!”

Curso alternativo

- 1 – Verificar se há falta de dados do usuário
- 2 – Msg_01 = “Verificar dados e tentar novamente!”

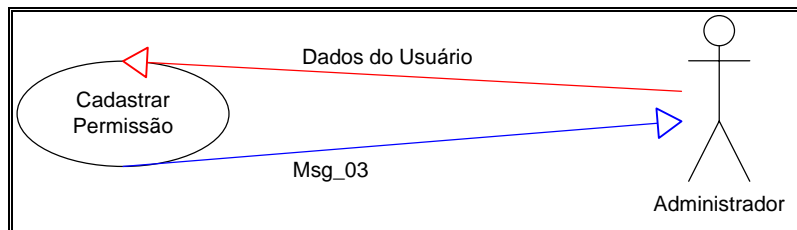
Evento 2

**Figura 5. Evento Manter Usuário**

Curso Normal

- 1 – Verificar alteração de dados do usuário
- 2 – Atender solicitação do cadastro
- 3 – Verificar todos os campos do formulário de cadastro
- 4 – Msg_02 = “Alteração de dados efetuada com sucesso!” ou “Manutenção efetuada com sucesso!”

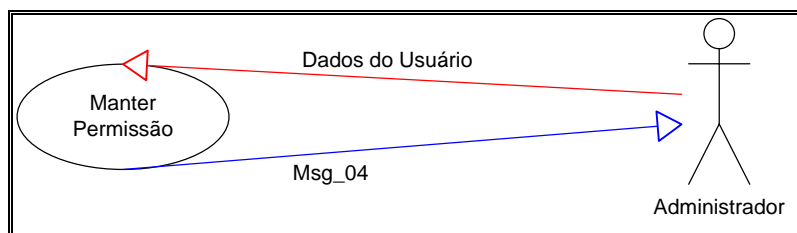
Evento 3

**Figura 6. Evento Cadastrar Permissão**

Curso Normal

- 1 – Estabelecer nível de permissão ao usuário
- 2 – Verificar dados do usuário
- 3 – Atender solicitação do cadastro
- 4 – Msg_03 = “Alteração efetuada com sucesso!”

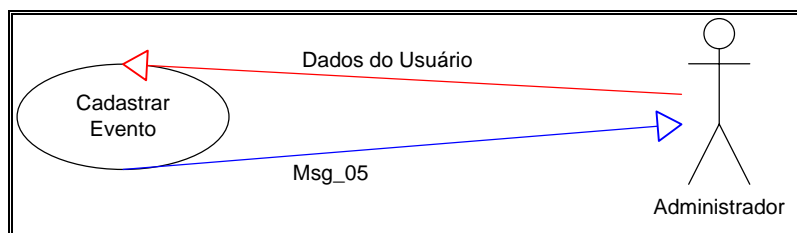
Evento 4

**Figura 7. Evento Manter Permissão**

Curso Normal

- 1 – Verificar dados de permissão
- 2 – Verificar dados do usuário
- 3 – Atender solicitação do cadastro
- 4 – Msg_04 = “Alteração efetuada com sucesso!”

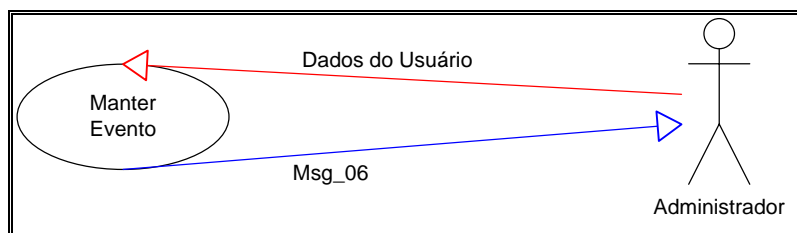
Evento 5

**Figura 8. Evento Cadastrar Evento**

Curso Normal

- 1 – Cadastrar evento
- 2 – Verificar dados do evento
- 3 – Atender solicitação do cadastro
- 4 – Msg_05 = “Cadastro efetuado com sucesso!”

Evento 6

**Figura 9. Evento Manter Evento**

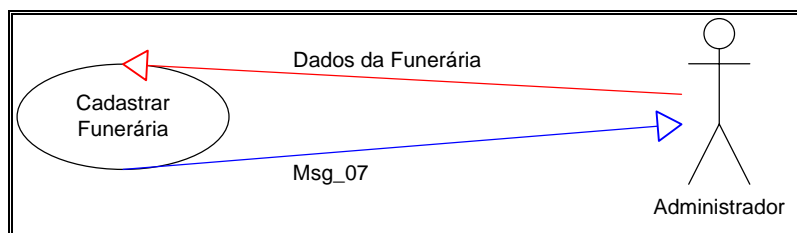
Curso Normal

1 – Verificar dados do evento

2 – Atender solicitação do cadastro

3 – Msg_06 = “Manutenção efetuada com sucesso!”

Evento 7

**Figura 10. Evento Cadastrar Funerária**

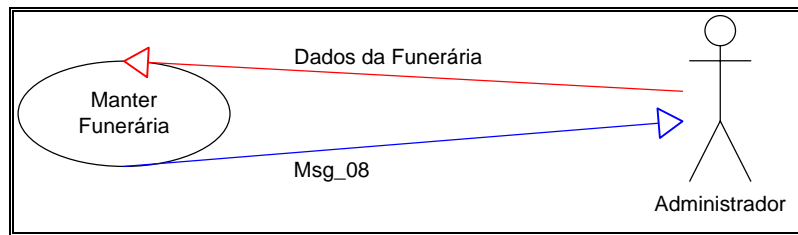
Curso Normal

- 1 – Verificar dados completos da funerária
- 2 – Preencher cadastro da funerária
- 3 – Atender solicitação do cadastro
- 4 – Verificar todos os campos do formulário de cadastro
- 5 – Msg_07 = “Cadastro efetuado com sucesso!”

Curso alternativo

- 1 – Verificar se há falta de dados da funerária
- 2 – Msg_07 = “Verificar dados e tentar novamente!”

Evento 8

**Figura 11. Evento Manter Funerária**

Curso Normal

- 1 – Verificar alteração de dados da funerária
- 2 – Atender solicitação do cadastro
- 4 – Verificar todos os campos do formulário de cadastro
- 5 – Msg_08 = “Manutenção efetuada com sucesso!”

Curso alternativo

- 1 – Verificar se há falta de dados da funerária
- 2 – Msg_08 = “Verificar dados e tentar novamente!”

Evento 9

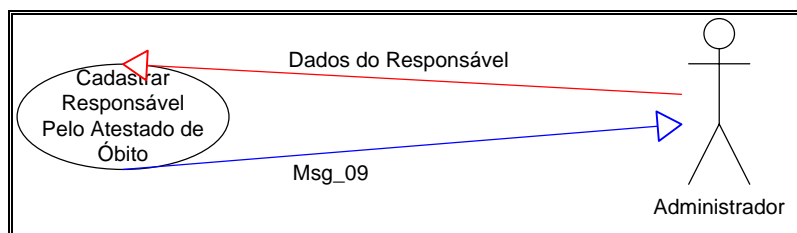


Figura 12. Evento Cadastrar Responsável pelo Atestado de Óbito

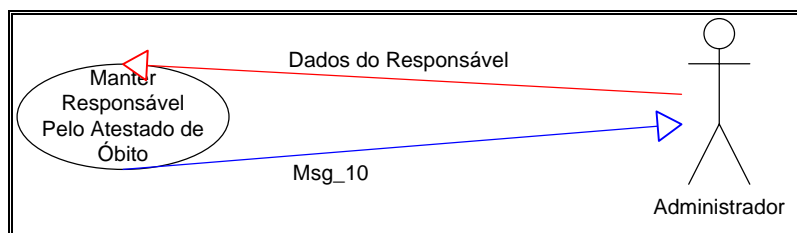
Curso Normal

- 1 – Verificar dados completos do responsável
- 2 – Preencher cadastro de responsável pelo atestado de óbito
- 3 – Atender solicitação do cadastro
- 4 – Verificar todos os campos do formulário de cadastro
- 5 – Msg_09 = “Cadastro efetuado com sucesso!”

Curso alternativo

- 1 – Verificar se há falta de dados do responsável
- 2 – Msg_09 = “Verificar dados e tentar novamente!”

Evento 10

**Figura 13. Evento Manter Responsável pelo Atestado de Óbito**

Curso Normal

- 1 – Verificar alteração de dados do responsável
- 2 – Atender solicitação do cadastro
- 4 – Verificar todos os campos do formulário de cadastro
- 5 – Msg_10 = “Manutenção efetuada com sucesso!”

Curso alternativo

- 1 – Verificar se há falta de dados do responsável
- 2 – Msg_10 = “Verificar dados e tentar novamente!”

Evento 11

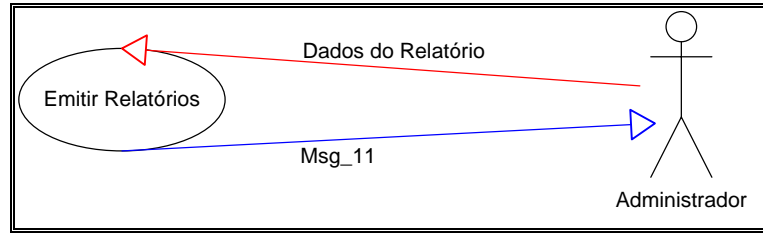
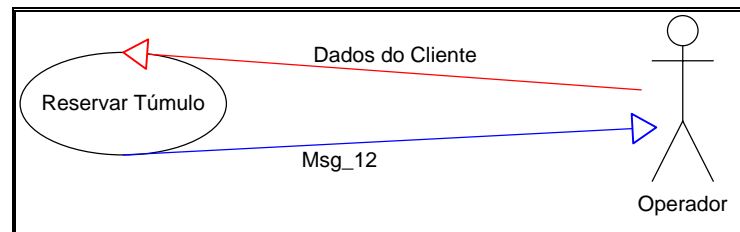


Figura 14. Evento Emitir Relatórios

Evento 12

**Figura 15. Evento Reservar TÍTULO**

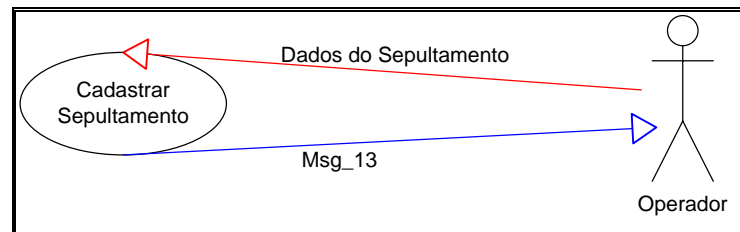
Curso Normal

- 1 – Verificar dados completos do cliente
- 2 – Preencher cadastro de cliente
- 3 – Atender solicitação do cadastro
- 4 – Verificar todos os campos do formulário de cadastro
- 5 – Msg_12 = “Cadastro efetuado com sucesso!”

Curso alternativo

- 1 – Verificar se há falta de dados do cliente
- 2 – Msg_12 = “Verificar dados e tentar novamente!”

Evento 13

**Figura 16. Evento Cadastrar Sepultamento**

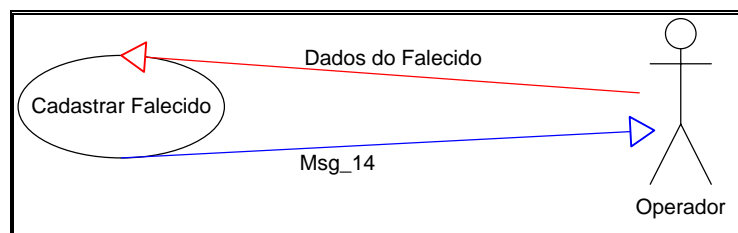
Curso Normal

- 1 – Verificar dados completos do sepultamento
- 2 – Preencher cadastro de sepultamento
- 3 – Atender solicitação do cadastro
- 4 – Verificar todos os campos do formulário de cadastro
- 5 – Msg_13 = “Cadastro efetuado com sucesso!”

Curso alternativo

- 1 – Verificar se há falta de dados do sepultamento
- 2 – Msg_13 = “Verificar dados e tentar novamente!”

Evento 14

**Figura 17. Evento Cadastrar Falecido**

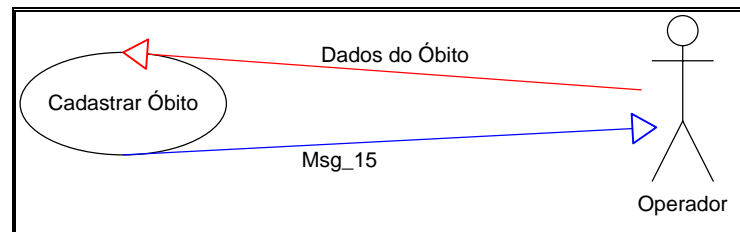
Curso Normal

- 1 – Verificar dados completos do falecido
- 2 – Preencher cadastro de falecido
- 3 – Atender solicitação do cadastro
- 4 – Verificar todos os campos do formulário de cadastro
- 5 – Msg_14 = “Cadastro efetuado com sucesso!”

Curso alternativo

- 1 – Verificar se há falta de dados do falecido
- 2 – Msg_14 = “Verificar dados e tentar novamente!”

Evento 15

**Figura 18. Evento Cadastrar Óbito**

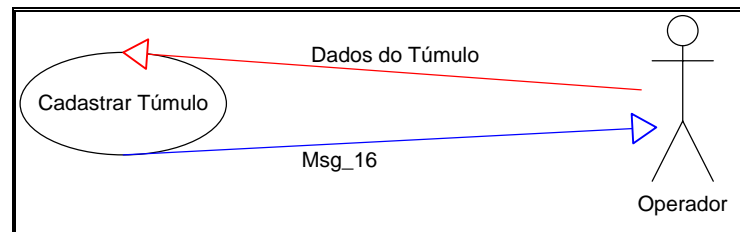
Curso Normal

- 1 – Verificar dados completos do óbito
- 2 – Preencher cadastro de óbito
- 3 – Atender solicitação do cadastro
- 4 – Verificar todos os campos do formulário de cadastro
- 5 – Msg_15 = “Cadastro efetuado com sucesso!”

Curso alternativo

- 1 – Verificar se há falta de dados do óbito
- 2 – Msg_15 = “Verificar dados e tentar novamente!”

Evento 16

**Figura 19. Evento Cadastrar Túmulo**

Curso Normal

- 1 – Verificar dados completos do túmulo
- 2 – Preencher cadastro de túmulo
- 3 – Atender solicitação do cadastro
- 4 – Verificar todos os campos do formulário de cadastro
- 5 – Msg_16 = “Cadastro efetuado com sucesso!”

Curso alternativo

- 1 – Verificar se há falta de dados do túmulo
- 2 – Msg_16 = “Verificar dados e tentar novamente!”

8.4. DIAGRAMA DE CLASSE

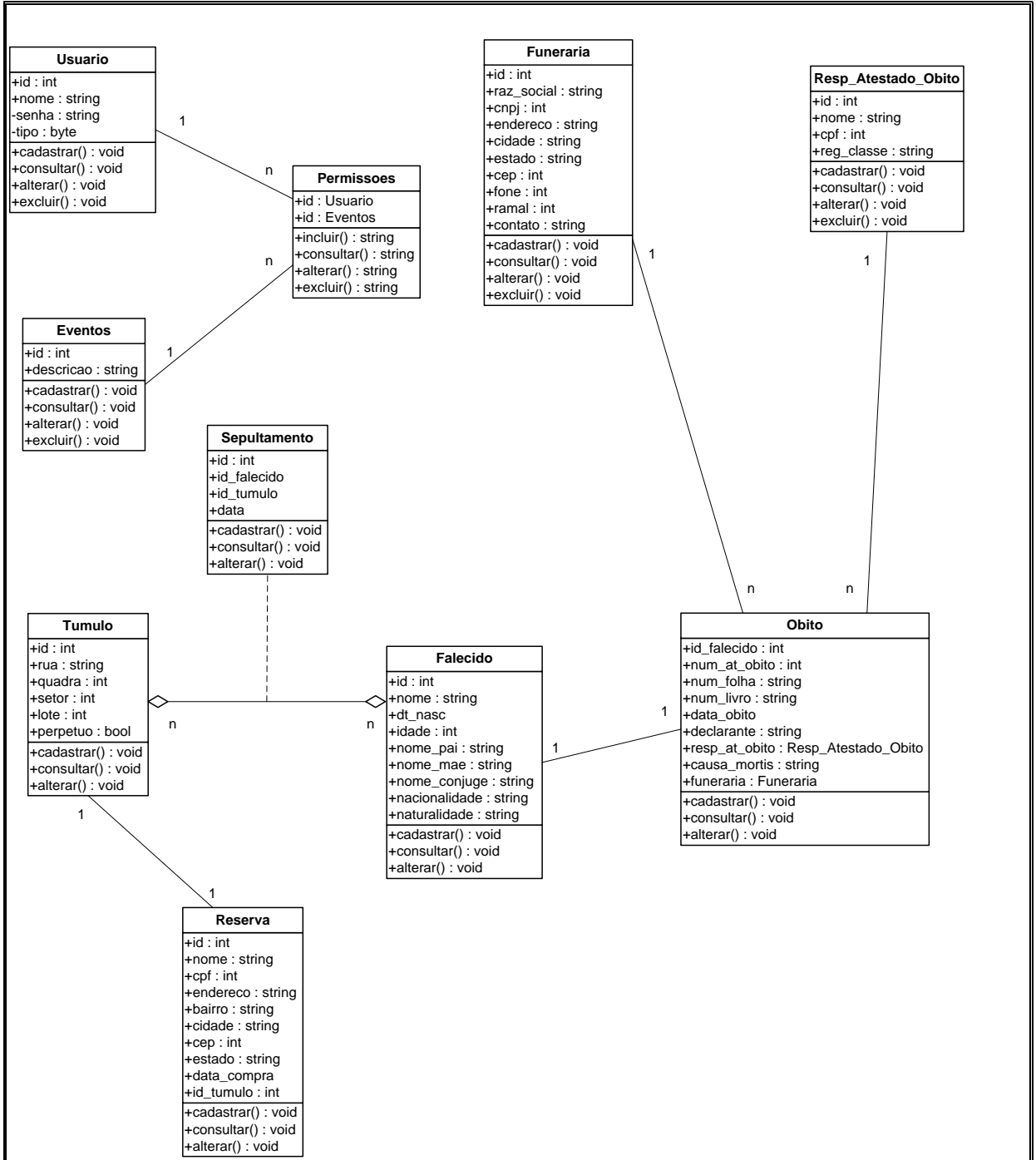


Figura 20. Diagrama de Classe

CONSIDERAÇÕES FINAIS

Estudos mostram que o caminho para a melhoria dos serviços públicos está na "terceirização" desses serviços, porém, atrelados a sistemas gerenciais informatizados que geram maior facilidade na demonstração de resultados, gerando, assim, nominal economia de recursos financeiros. Com a implantação de um sistema gerencial informatizado nos Cemitérios Municipais que oferecem serviços públicos, o erário público seria o maior beneficiário com as economias acarretadas, bem como a população em geral que teria, em contrapartida, a excelência na execução de serviços.

Com a implantação do SisCem seriam visíveis as melhorias no que diz respeito ao armazenamento de dados e à agilidade no fornecimento de informações pois, em se tratando de um sistema de fácil aprendizado e operacionalização, não demandaria em contratação de pessoal especializado, sendo absorvida a mão de obra já existente nessas instituições municipais.

Nesse sentido, o presente projeto pretendeu contribuir para a qualificação e efetivação dos serviços prestados pelo Cemitério Municipal de Assis facilitando o gerenciamento e a administração através de um sistema informatizado e eficiente.

REFERÊNCIAS BIBLIOGRÁFICAS

RIOS, Emerson. Organização da informática na empresa. Rio de Janeiro: Editora Campus, 1985.

RUMBAUGH, James. Modelagem e Projetos baseados em objetos. Rio de Janeiro: Editora Campus, 1994.

Site Decision Report

<http://www.decisionreport.com.br/publique/cgi/cgilua.exe/sys/start.htm?infoid=1590&sid=2>

Arquivo capturado em 04 de abril de 2008.

SOUKUP, Ron. Desvendando o Microsoft SQL Server 6.5. Rio de Janeiro: Editora Campus, 1998.

STELLMAN, Andrew; GREENE, Jennifer. Use a Cabeça! C#. Rio de Janeiro: Alta Books, 2008.

TROELSEN, Andrew. Profissional C# e a Plataforma .net 3.5 - Curso Completo. Rio de Janeiro: Alta Books, 2009.

WILLE, Christoph. Apresentando C#. São Paulo: Berkeley Brasil, 2001.