PROJETO INTERDISCIPLINAR

TEMA: SAÚDE E BEM ESTAR

Em atendimento às requisições das disciplinas:

* Engenharia de Software I
* Desenvolvimento WEB I
* Design Digital

Araras – SP

2021

Equipe de desenvolvimento:

Alunos do curso de Tecnologia em Desenvolvimento de Software Multiplataforma - 1º Semestre

Participantes:

* Alexandre Guidio Paquola
* Cleber Luís Aranha
* Juliana Barbosa Tangerino
* Marcos Vinicius da Silva Guedes
* Sabrina Borges Ribeiro

SUMÁRIO (REVISAR SEMPRE)

1 - INTRODUÇÃO.....................................................................................................13

2 - OBJETIVOS.........................................................................................................14

3 - JUSTIFICATIVA...................................................................................................15

4 - MOTIVAÇÃO........................................................................................................16

5 - PERSPECTIVAS DE CONTRIBUIÇÃO ...............................................................17

6 – METODOLOGIA DE TRABALHO .......................................................................18

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

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. ESCOLHA DO PROJETO

SOFTWARE ONLINE PARA ADMINISTRAÇÃO DE CEMITÉRIOS COM PESQUISA PÚBLICA DE SEPULTAMENTOS

2. NOME DO PROJETO

ENTE QUERIDO

3. OBJETIVOS

O objetivo deste trabalho é documentar e detalhar um sistema de gerenciamento para propiciar à administração do Cemitério 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, bem como disponibilizar uma tela de consultas públicas das pessoas sepultadas e da localização física das respectivas sepulturas.

4. JUSTIFICATIVA

Durante os anos de 2020 e 2021 houve uma pandemia de COVID-19 também conhecida como pandemia de corona vírus. Uma doença respiratória causada pelo corona vírus da síndrome respiratória aguda grade (SARS-CoV-2), classificada em 20/01/2020 pela Organização Mundial da Saúde (OMS) como Emergência de Saúde Pública de Âmbito Internacional e em 11/03/2020 como pandemia.

Os sintomas são altamente variáveis: há pessoas infectadas não apresentando nenhum sintoma e muitas outras morrendo em decorrência das mesmas e também em decorrência de outras doenças já pré-existentes que foram agravadas pelo corona vírus.

É uma doença altamente contagiosa espalhando-se principalmente pelo ar quando as pessoas estão perto umas das outras.

A principal medida preventiva recomendada foi o distanciamento social.

Em função dessa pandemia milhões de pessoas morreram em decorrência deste vírus.

Como a doença é altamente contagiosa a grande maioria dos mortos teve que ser sepultada rapidamente, isso, aliado a recomendação do distanciamento social não permitiu que parentes e amigos “velassem” seus entes queridos e em contra partida houve a sobrecarga de tarefas dos cemitérios, que tiveram de se adequar aos novos padrões de segurança para evitar o contágio de seus colaboradores e a grande demanda de serviços ocasionada pela pandemia.

É fundamental o controle de sepultamentos e a disponibilidade de consultas de dados aos entes queridos.

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.

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 ENTE QUERIDO, criar-se-á uma nova perspectiva administrativa, uma vez que muitos Cemitérios (principalmente públicos e em pequenas cidades) ainda não contam com um sistema informatizado de Administração. Com o ENTE QUERIDO, espera-se contribuir com o gerenciamento do órgã, 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 serão utilizados os conceitos e conteúdo das aulas ministradas durante o primeiro semestre do Curso de Desenvolvimento de Software Multiplataforma nas disciplinas listadas no preâmbulo deste documento.

Inicialmente será realizada uma entrevista com os funcionários do Cemitério Municipal da cidade de Leme para levantamento das necessidades e requisitos do software onde serão sanadas as dúvidas prementes.

Na etapa seguinte será elaborado toda documentação necessária em atendimento às requisições das disciplinas envolvidas no projeto com detalhamento das rotinas e funções que deverão ser desenvolvidas pelo sistema para atendimento às necessidades das instituições que queiram fazer uso desta ferramenta.

***?***

***?***

***?***

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 Sequencia mostra objetos e uma sequência 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 que 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 frequentemente 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, mostras 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 ―todoparte‖.

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 o 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 Sequencia**

Diagramas de Sequencia 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 Sequencia colocam ênfase especial na ordem e nos momentos nos quais mensagens para os objetos são enviadas.

Em Diagramas de Sequencia 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 Sequencia, 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 sequência 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 explanar 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, 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 sequência 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 aos 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 sequenciais 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 possui 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).

***?***

***?***

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 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.

A utilização do sistema se dará pelos seguintes atores:

* Administrador do Sistema;
* Administrador do Cemitério;
* Operador do Sistema;
* Usuário público.

Administrador do Sistema: responsável pela inclusão de novos cemitérios que utilizarão o sistema e pela inclusão dos seus respectivos Administradores do Cemitério.

Administrador do Cemitério: usuário “master” de cada cemitério. Ele será o responsável pelas operações da instituição a que estiver vinculado, gerenciando os operadores do sistema, estipulando taxas, cadastrando funerárias, convênios firmados com terceiros, emitindo relatórios administrativos e financeiros.

Operador do Sistema: usuário vinculado à uma determinada instituição que será responsável pelas operações básicas e diárias do cemitério, tais como: consultas, registro dos sepultamentos e recebimento de taxas e serviços.

Usuário Público: público em geral, o qual terá acesso apenas a uma pesquisa simplificada dos sepultamentos.

Será, então, criada uma interface de apresentação da ferramenta com 2 (duas) opções principais de acesso:

* Tela de “Login” ao sistema onde deverão ser informados o “Nome de Usuário” e a “Senha” para acesso aos serviços;
* Pesquisa pública. Nesta opção será permitida a consulta de sepultamentos simplificada por qualquer pessoa que acesse ao site, sem prévio cadastro.

Ao Administrador do Sistema serão atribuídas as funções de inclusão das instituições interessadas na utilização do sistema e dos seus respectivos administradores, denominados aqui de Administrador do Cemitério.

TERMINAR DE DESCREVER AS FUNÇÕES DE CADA ATOR:

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 sequencial 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