

UNIVERSIDADE FEDERAL DE CAMPINA GRANDE  
CENTRO DE CIÊNCIAS E TECNOLOGIA  
DEPARTAMENTO DE CIÊNCIAS E COMPUTAÇÃO

# Relatório Final do Estágio

Aluna: Emmanuelle Albuquerque dos Santos Coelho  
Matrícula: 29621021  
Disciplina: Estágio Supervisionado



Biblioteca Setorial do CDSA. Maio de 2021.

Sumé - PB

# 1. Apresentação

O objetivo do sistema SEBRAE pode ser traduzido pela busca do fortalecimento e consolidação dos pequenos negócios existentes, bem como o incentivo ao surgimento de novos.

Através da oportunidade de estagiar no setor de informática do SEBRAE-Pb, em João Pessoa, eu pude experimentar o primeiro contato com uma empresa, e por em prática os conhecimentos adquiridos ao longo do curso. A interação com profissionais, tanto da mesma área, como de áreas diferentes, e a participação nas decisões do setor, vieram a somar os fatos anteriormente descritos, os quais acredito constituirão o objetivo acadêmico para com um estágio.

Esse relatório encontra-se dividido em sete partes. A primeira delas corresponde à apresentação do objetivo acadêmico e da empresa, além de mencionar como o relatório está estruturado. A segunda traz uma descrição mais detalhada do ambiente de estágio, dando ênfase à função de informática e narrando os aspectos positivos e negativos nele presentes. Na terceira parte será feita uma apresentação completa dos problemas por mim analisados, para que na parte seguinte seja enunciada a minha proposta para solucioná-los.

A quinta parte deste relatório é destinada a uma narração cronológica das atividades desenvolvidas durante o estágio, indicando os métodos e técnicas utilizados, os tempos gastos e os produtos gerados por cada uma dessas atividades.

As conclusões e sugestões são deixadas para a sexta parte, onde serão citadas as situações de desconforto na aplicação dos conhecimentos adquiridos durante o curso. Finalmente, a última parte, que é destinada aos anexos e encontra-se subdividida em:

- A1. Plano de Estágio
- A2. Declaração da empresa aprovando o resultado do estágio
- A3. Resumo metodológico
- Produtos trabalhados nas atividades do estágio.

## **2. Ambiente de Estágio**

O setor de Informática da unidade de João Pessoa é formado por profissionais das áreas de suporte técnico, programação, administração de redes e de banco de dados, contando, ainda, com estagiários, cujo papel é prestar serviços de suporte a usuários e máquinas.

Graças à variedade na qualificação desses profissionais e ao empenho em sempre fornecer melhores serviços e disseminar informação, o setor de Informática é capaz não só de administrar a informática nessa unidade, como também monitorar e prestar suporte técnico às demais unidades que compõem o SEBRAE-Pb.

### **2.1 Aspectos Positivos**

Muitos foram os aspectos positivos presentes no Ambiente de estágio. Para começar, eu gostaria de mencionar o tratamento que é dado aos estagiários iniciantes, fato que ressalto como a maior virtude do setor. Já no primeiro dia, somos apresentados a todos os colegas, e a seguir às atividades por cada um deles realizada. Isso faz com que não nos sintamos tão perdidos e nervosos e possamos começar a interagir com todos. Em seguida, somos informados sobre as atividades do setor, e deixados em companhia dos estagiários mais antigos, para que possamos acompanhá-los, quando são requisitados, para prestar suporte a algum equipamento ou usuário. Desse modo, já começamos a conhecer os demais setores da empresa e alguns dos problemas mais frequentemente apresentados em cada um deles.

Um segundo ponto positivo é o apoio que recebemos dos que trabalham no setor. Qualquer dúvida que tenhamos é prontamente esclarecida por um colega, por mais banal que pareça ser e, assim, somos capazes de aprimorar os nossos conhecimentos e até mesmo adquirir novos.



Não poderia deixar de mencionar o clima de descontração e cumplicidade que impera no setor. A relação chefe – empregados é respeitosa e amigável. Contamos com o apoio e a confiança que o supervisor tem para conosco e o nosso trabalho.

Finalmente, é importante citar a oportunidade que eu tive de exercer ou auxiliar em outras funções como atualizar páginas na Internet, receber treinamento no novo sistema de atendimento aos clientes do SEBRAE-Pb, que estava para ser implantando na época e, também, colaborar para a alimentação do banco de dados da Biblioteca Virtual, que também estava para ser inaugurada, visando fornecer um maior apoio a esses mesmos clientes.

Todos esses aspectos positivos encontrados no meu ambiente de estágio fizeram-me amadurecer profissionalmente, dando-me a oportunidade de mostrar os conhecimentos que possuo, bem como adquirir novos em áreas que antes não possuía.

## **2.1 Aspectos Negativos**

O único aspecto negativo que sou capaz de citar, diz respeito às críticas que com alguma frequência eram feitas tanto aos estagiários como, ao setor de Informática. Apesar delas não atingirem diretamente minha pessoa, eu me sentia bastante decepcionada de saber que, apesar dos esforços para fornecer serviços de qualidade que satisfizessem os usuários e proporcionassem os merecidos elogios a todos nós, algumas pessoas preferiam ignorá-los e criticá-los a incentivá-los.

### 3. Descrição do Problema

O SEBRAE-Pb encontra-se dividido em diversos setores. Cada um desses setores possui um número diferenciado de profissionais que nele atuam. O mesmo pode ser afirmado com relação ao número de computadores que existem nos mencionados setores.

Diante de um volume tão grande de equipamentos espalhados ao longo de toda empresa, seria interessante que o setor de Informática tivesse cadastros de todos esses equipamentos, a fim de que pudesse manter um controle sobre a quantidade e a alocação daqueles pelos demais setores, uma vez que fica praticamente impossível manter toda essa informação apenas na memória.

A desvantagem da inexistência desses cadastros faz sentir-se, principalmente, quando os estagiários são encarregados de uma atividade de formatação de disco rígido, por exemplo. A formatação não é o problema, porque isso todos eles já estão cansados de fazer. A questão é: quais softwares deverão ser reinstalados depois que o disco já estiver formatado? Na ausência de um cadastro para consultar, o estagiário vai tentar lembrar de todos eles, ou, se já trabalhar no setor há muito tempo, saberá sem nem pensar o que deverá reinstalar no equipamento. A outra opção, que é a adotada principalmente pelos estagiários iniciantes, é perguntar, a um colega mais antigo, quais softwares são necessários no computador para que o usuário possa realizar as suas tarefas.

Em qualquer um dos casos, não há garantias de que o computador seja devolvido ao setor de origem e, pouco depois, o usuário ligue para a Informática solicitando a presença de um estagiário para instalar um programa que é crucial para o desempenho de seu papel. Essa situação causa inconvenientes tanto para o usuário, que tem seu trabalho interrompido por algum tempo, como para o estagiário, que deve se dirigir ao setor para terminar um serviço que já considerava terminado.

Uma outra questão que merece atenção está relacionada à quantidade e função dos softwares que o setor mantém, para que sejam instalados nos

equipamentos da empresa. Se cada setor é responsável por uma determinada tarefa, significa dizer que cada um tem diferentes necessidades. Sendo assim, os softwares instalados nos computadores variarão, dependendo da função e, também, do tipo de trabalho que o profissional executa. Como resultado, não existe uma padronização nem em nível de empresa, nem de setor, o que torna difícil, principalmente, para um estagiário iniciante, o conhecimento de quais softwares deverão ser instalados em cada equipamento, bem como, para que servem. O desconhecimento dos softwares que o setor de Informática possui e quando da existência de mais de uma versão, qual deverá ser instalada, é mais um motivo de preocupação.

Uma outra observação que eu gostaria de fazer, está relacionada aos serviços e atividades executadas pelos estagiários. Até o último dia do meu estágio, o setor não possuía nenhum registro dos serviços que os mesmos prestavam e muito menos das atividades por eles realizadas para equacionar os problemas. Como resultado, não há como gerar relatórios a respeito de quantos serviços são prestados por dia, quem foram os responsáveis por cada um deles, o que foi feito para resolver os problemas e o tempo que se levou para terminar cada um deles.

## 4. Proposta de Solução

Considerando os problemas descritos no item anterior, a proposta de solução é o desenvolvimento de um sistema que possa ser alimentado com informações que gerem os relatórios e as estatísticas de que tanto o setor precisa.

Primeiramente, deverão ser feitos cadastros de todos os equipamentos espalhados pela empresa, bem como dos setores aos quais pertencem. No cadastro dos computadores deverá haver informações sobre o código a ele atribuído, a quantidade de memória RAM, a capacidade do disco rígido, a existência ou não de dispositivos de CD\_ROM e DISKETT, a descrição dos atalhos não padrão localizados na área de trabalho e o sistema operacional que roda no computador. O cadastro de um setor será constituído por um código a ele atribuído e o seu nome.

Outros cadastros também deverão ser feitos, pois serão de fundamental importância para as respostas que serão produzidas pelo sistema. São eles:

- Cadastro de softwares utilizados pela empresa e presentes na Informática;
- Cadastro dos estagiários;
- Cadastro dos sintomas que possam vir a ser apresentados pelos equipamentos; e
- Cadastro das atividades que, possivelmente, serão utilizadas pelos estagiários na solução de problemas.

Todos esses cadastros citados acima serão compostos de dois campos. O primeiro é um código de identificação única, o que significa dizer que, para cada item cadastrado, haverá uma única ocorrência desse código. O outro campo do cadastro será uma referência ao nome do estagiário, descrição do sintoma, nome do software ou descrição da atividade, dependendo do tipo do cadastro.

É importante alertar para o fato de que os sintomas e as atividades acima mencionados não correspondem aos sintomas que são apresentados pelos equipamentos ou às atividades que são realizadas na tentativa de solucionar o problema. Tais cadastros foram introduzidos, a fim de que pudéssemos criar um

conjunto padronizado de sintomas e atividades que facilitassem as consultas ao sistema. Um exemplo bastante prático pode ser dado, pegando o caso de um estagiário que não está obtendo sucesso na solução de um determinado problema. Uma vez introduzido o sistema, ele poderá procurar pelo sintoma que o equipamento está apresentando e utilizá-lo para obter informações sobre que atividades foram realizadas por outros estagiários quando enfrentaram o mesmo problema. Se não houvesse a padronização dessas informações, a resposta a essa consulta seria impossível de ser obtida, visto que cada indivíduo registraria o sintoma e as atividades da forma que lhe fosse mais conveniente, fazendo com que o mesmo sintoma ou a mesma atividade, recebesse as mais variadas descrições, tornando inviável a utilização do sistema para tirar as dúvidas com relação a um determinado problema.

Uma outra característica do sistema é que ele permitirá o registro das ordens de serviço, assim como o seu controle. Para que isso aconteça, será feita a abertura de uma Ordem de Serviço, mediante o fornecimento do código do computador e do código do setor. Feito isso, o sistema gravará a data e a hora em que a ordem de serviço foi solicitada e gerará um código para ela, que deverá ser apresentado na tela. O registro é gravado no banco de dados e uma mensagem de confirmação da abertura é, também, lançado na tela.

A distribuição dos serviços entre os estagiários será registrada pelo sistema para uma posterior geração de relatórios sobre os serviços prestados por cada um durante um determinado período. Para tanto, o sistema requererá o código da ordem de serviço e o código do técnico que deverá lhe ser vinculado. Internamente, o sistema registrará a data e a hora em que a vinculação ocorreu. Quando o técnico terminar o serviço, o sistema deverá ser informado através do código da ordem de serviço e do técnico, para que possa registrar a data e a hora em que foi dada a desvinculação. O mesmo acontece se o estagiário for dispensado seja porque não obteve sucesso na solução do problema ou porque é chegado o fim do seu turno.

Uma ordem de serviço poderá ser vinculada e desvinculada de um técnico inúmeras vezes. Essas vinculações poderão ser feitas ao mesmo técnico

ou a técnicos diferentes, ficando a decisão a cargo do supervisor, e continuarão até que o problema do computador em questão tenha sido resolvido. Uma vez que isso aconteça, é chegada a hora de fechar a ordem de serviço. Para isso, basta fornecer o código da ordem de serviço, para que o sistema localize-a e registre a data e a hora de seu fechamento.

Serão permitidas quatro tipos de consulta:

- 1) Consulta a Ordens de Serviço Pendentes;
- 2) Consulta a Ordens de Serviço Por Setor;
- 3) Consulta a Ordens de Serviço Por Técnico; e
- 4) Consulta a Ordens de Serviço Por Computador.

O primeiro tipo de consulta será utilizada para obter-se os serviços pendentes como seu próprio nome já diz. Será mostrada uma lista com tais ordens de serviço, juntamente com a data e a hora em que foram abertas, o código do computador e o nome do setor aos quais está vinculada. Paralelamente, será apresentada uma lista, para cada uma dessas ordens, contendo os nomes dos técnicos que já foram vinculados a elas, caso já tenha ocorrido alguma vinculação, assim como a data e a hora das vinculações e desvinculações.

O segundo tipo permitirá a geração de relatórios sobre os serviços prestados a cada setor, podendo-se, assim, derivar estatísticas e definir quais são os setores que requerem mais frequentemente o auxílio da Informática e por que.

A terceira forma de consulta visa um acompanhamento, por parte do supervisor, do desempenho de cada estagiário. São gerados relatórios, listando todos os serviços aos quais um determinado estagiário foi vinculado. A partir daí, o supervisor poderá obter estatísticas e outros dados a respeito do desempenho do estagiário num dado período de tempo.

A última forma de consulta tem por objetivo fornecer um relatório de todas as Ordens de Serviço que foram abertas para um determinado computador. Desse modo, será possível fazer um balanço dos problemas mais frequentemente apresentados e encontrar formas de minimiza-los.

Nos três últimos tipos de consulta, é possível, ainda, fazer uma filtragem das ordens de serviço que constarão no relatório, baseado no estado em que elas

deverão se encontrar. Pode-se optar por aquelas em aberto, ou só pelas que já foram fechadas, bem como fazer a opção para que todas elas, independentemente do estado em que se encontram, constem no relatório.



## 5. Atividades Desenvolvidas

O desenvolvimento do sistema proposto como solução para o problema apontado no item três desse relatório foi dividido em quatro etapas ou atividades:

- 1) Definição dos Requisitos
- 2) Construção dos Modelos de Dados
- 3) Construção do Banco de Dados
- 4) Implementação Inicial

### 1ª ATIVIDADE: DEFINIÇÃO DOS REQUISITOS

A primeira etapa, definição dos requisitos, foi concluída em pouco mais de um mês, o que me fez perder um pouco de tempo em relação às demais atividades. O ritmo de trabalho foi intenso, levando em média 10 horas de trabalho por dia, tendo em vista que se fez necessária à leitura de dois livros sobre Análise Essencial, para que eu pudesse relembrar todas as informações necessárias para realizar essa atividade e a seguinte, que ainda fazia parte da análise do sistema. Além de estudar o trabalho realizado pelos estagiários para extrair características para o meu projeto, eu conversei com alguns deles sobre as minhas idéias e ouvi as suas opiniões sobre o que incluir ou não, já que eles seriam os mais interessados no assunto.

Durante essa fase, minha preocupação foi definir qual seria a essência do sistema, ou seja, quais seriam as atividades essenciais do sistema e que dados deveriam constar na sua memória essencial, entendendo-se por memória essencial, o conjunto de todos os dados que são necessários ao sistema para que ele possa executar as atividades essenciais.

Eu já tinha em mente um esboço dos estímulos aos quais devia responder e quais seriam os agentes responsáveis por gerar tais estímulos, mas o



*difícil era organizar todas essas idéias de forma a expressar, o mais claramente possível, o que eu pretendia com esse sistema.*

A técnica escolhida para fazer a análise do sistema foi a Análise Essencial, por ser, na minha opinião, a mais completa, visto que faz intenso uso de ferramentas de modelagem e oferece uma abordagem que considera o sistema, tanto do ponto de vista das funções e dos dados, como dos controles.

Ao final dessa etapa do projeto, eu elaborei uma lista com as atividades essenciais e os objetos de depósito de dados da memória essencial. A título de ilustração, a lista das atividades essenciais encontra-se em Anexos-A4, página 1.

## **2ª ATIVIDADE: CONSTRUÇÃO DOS MODELOS DE DADOS**

Essa segunda etapa do projeto era prevista para durar dois meses, mas como já havia perdido alguns dias, devido ao atraso com a primeira atividade, tive que aumentar a carga horária de trabalho para conseguir terminá-la dentro do prazo que havia previsto, sem levar em conta mais adiamentos. Dedicando o equivalente a 9 horas por dia, durante os dias úteis, e 11 horas nos fins de semana, essa atividade foi a mais trabalhosa. Precisava partir de todos os dados, que até então havia conseguido, e chegar ao produto final que seriam os modelos de dados.

Fazendo uso dos princípios de particionamento para modelos essenciais apresentado pelos autores Stephen M. McMenamim e John F. Palmer, em seu livro Análise Essencial de Sistemas, revisei o particionamento do sistema em atividades essenciais e segui para o reconhecimento dos eventos. A dificuldade, nesse ponto, foi reconhecer o que seria realmente um evento. Era preciso saber fazer a distinção entre o sistema e o seu ambiente para que eu não houvesse riscos de introduzir uma atividade do sistema ou até mesmo uma resposta como um evento. Outro fato que levei em consideração, na hora de reconhecer eventos,

foi que eu devia olhar um evento do ponto de vista do sistema, desse modo, as chances de cometer erros seriam mínimas.

Havendo encontrado os eventos, particionei o sistema segundo os mesmo, a fim de fazer testes para saber se as atividades essenciais que eu havia listado na primeira etapa do projeto podiam realmente ser qualificadas como tal. Esse particionamento também me ajudou a encontrar quais deveriam ser as respostas aos estímulos que iriam ativar tais atividades no sistema.

De posse dos eventos, atividades essenciais, respostas e depósitos de dados, pude avançar no projeto, dando início a modelagem dos dados que foi feita da seguinte maneira:

- **Memória Essencial:** sua modelagem foi feita através da modelagem dos objetos de depósitos de dados. A ferramenta utilizada foi o dicionário de dados. (vide Anexos-A4, páginas 3-6).
- **Estímulos e Resultados Externos:** foram modelados através do dicionário de dados.
- **Resposta Planejada, Acessos e Atualizações de Memória:** sua modelagem foi feita em forma de miniespecificação.

A modelagem dos dois últimos grupos de dados podem ser vistas em Anexos-A4 a partir da página 7. No decorrer dessas páginas, poderão também ser vistos os Diagramas de Fluxo de Dados e de Entidade e Relacionamento. O primeiro corresponde a uma forma gráfica de apresentar os fluxos de dados, os depósitos de dados e as entidades externas. O outro é uma ferramenta gráfica para modelar os objetos de depósitos de dados, também conhecidos como entidades, e para representar os relacionamentos entre eles.

Os símbolos existentes nas extremidades das linhas nos Diagramas de Entidade-Relacionameto representam os seguintes tipos de mapeamentos:

<u>Símbolo</u>	<u>Mapeamento</u>
—+—	1 para 1
—	1 para 1 ou várias
—○+	1 para 0 ou 1
—○	1 para 0 ou várias

### 3ª ATIVIDADE: CONSTRUÇÃO DO BANCO DE DADOS

Depois das atividades de análise, foi chegada a hora de iniciar o projeto do banco de dados. Apesar de parecer uma tarefa simples, havia muita coisa a ser feita, principalmente antes de obter o esquema de banco de dados.

Para a modelagem lógica dos dados, optei pelo Modelo Relacional, por já o haver estudado durante o curso de Banco de Dados I e já possuir um certo entrosamento com seus conceitos. Tal modelo procura representar as entidades e os relacionamentos, entre elas existentes, por conjuntos de tabelas e, como o próprio nome já sugere, baseia-se no conceito matemático de relação.

Partindo dos pequenos Diagramas de Entidade-Relacionamento que haviam sido construídos durante a atividade anterior, os passos para obter o conjunto completo de tabelas foram:

- Transformar todas as entidades em tabelas, fazendo com que os atributos que as descrevem tornassem-se as colunas das tabelas e;
- Transformar os relacionamentos do tipo M:N em tabelas que contivessem as chaves primárias de cada entidade envolvida no relacionamento.

As tabelas que representam as entidades que participam de relacionamentos do tipo 1:N, possuem uma coluna a mais, que é destinada à chave primária da outra entidade, tornando-se, assim, uma chave estrangeira para as primeiras.

O próximo passo foi utilizar a normalização para validar o projeto relacional, evitando anomalias de inclusão, remoção e atualização. Feito isso, o banco de dados já poderia passar de um projeto para algo concreto.

O software utilizado para a construção e manipulação do banco de dados foi o Microsoft SQL Server 7.0, que faz uso da linguagem SQL. Uma descrição completa dos comandos usados para a criação das tabelas é fornecida em Anexos-A4, páginas 48 e 49.

A duração dessa etapa do projeto foi de exatamente uma semana.

#### **4ª ATIVIDADE: IMPLEMENTAÇÃO INICIAL**

A quarta e última atividade consiste numa implementação inicial do software. O motivo pelo qual fiz questão de mencionar a palavra inicial, desde o plano de estágio, é porque eu sabia que para um sistema desse porte, a implementação exigiria muito mais do que um mês para ficar pronta e esse era um prazo que não teria até o dia da defesa.

O sistema foi implementado quase que em sua totalidade utilizando o Delphi 5.0. Foram feitos um total de 36 formulários, que atendem às necessidades de todas as atividades essenciais do sistema, com exceção da atividade Consulta Por Computador. Esta não pode ser implementada, devido à falta de tempo.

Uma vez que não foi possível imprimir todos os formulários e anexá-los ao presente relatório, optei por fazer uma breve descrição de como eles se apresentam, antes de demonstrar a linha de raciocínio que foi usada para a implementação de cada atividade.

##### **1º) Formulário Principal**

Esse formulário corresponde à janela principal do programa que permanece visível durante todo o tempo em que o programa estiver sendo utilizado. Ele possui um menu principal que oferece as seguintes opções:

- Cadastro;
- Atualização;
- Exclusão;
- Consulta e;
- Serviço.

Essas opções serão melhor detalhadas no decorrer da explicação dos demais formulários.

##### **2º) Formulário de Cadastro de Computador**

Acessando o menu *Cadastro* e escolhendo a opção *Computador*, esse formulário é criado e visualizado pelo usuário. Ele fornece os campos necessários para que o usuário entre com os dados para o cadastro de um computador, ou seja, para cadastrar um computador, o usuário deverá entrar com todos os campos existentes na tabela computador. Assim que o usuário tiver concluído a entrada de dados, ele clica no botão salvar e o cadastro é salvo se houver sido corretamente preenchido.

Internamente, o programa analisará campo por campo, efetuando a validação dos dados e retornando mensagens de erro caso haja alguma tentativa de ferir as regras de integridade e de domínio do banco de dados.

Trechos do código utilizado para efetuar o cadastro são mostrados em Anexos-A4, páginas 49 e 50.

### **3º) Formulários de Cadastro de Setor, Sintoma, Técnico, Software e Atividade**

Esses quatro formulários serão descritos ao mesmo tempo, porque são idênticos. Com exceção dos nomes que lhe são dados, as demais características são comuns a todos eles, inclusive o código executado para salvar o cadastro.

Eles são criados quando se seleciona o menu *Cadastro* e depois qualquer um de seus nomes. O único dado de entrada que será requisitado ao usuário é o nome do setor, no caso do cadastro de um setor, ou o nome de um técnico, no caso do cadastro de um técnico e assim por diante.

Em Anexos-A4, página 50, é ilustrado o código que é executado, quando o usuário clica no botão para salvar o cadastro, e apesar de fazer alusão ao caso do cadastro de um setor, permanece válido para qualquer um dos outros acima citados.

Se o registro for inserido com sucesso no banco de dados, o sistema retornará o código do elemento cadastrado.

### **4º) Formulário de Cadastro dos Softwares de um Computador**

No formulário destinado ao cadastro de computadores, existe um botão chamado Softwares. Esse botão permanece desativado até que o usuário entre com os dados de um computador e clique no botão salvar para incluí-los no banco de dados. Se a inclusão for feita com sucesso, o botão Softwares é habilitado para que o usuário possa clicar nele e ter acesso ao formulário que o permitirá listar os softwares que estão presentes no computador.

Trata-se de um formulário bastante simples que permite ao usuário selecionar um software de cada vez e salvar a vinculação dele com o computador cujo cadastrado foi recém feito.

A especificação do código que é executado para que a vinculação seja registrada no banco de dados aparece em Anexos-A4, páginas 50 e 51.

#### **5º) Formulário de Atualização de Cadastro de Computador**

O formulário para atualizar o cadastro de um computador é muito parecido com o de cadastro do mesmo, com pouquíssimas diferenças. A primeira diferença é que o único dado que o usuário precisa fornecer é o código do computador. Depois, ele clica no botão de pesquisa para obter os dados do cadastro.

Será permitida a alteração de qualquer campo, menos do código, que por segurança permanecerá indisponível. Terminada a alteração ou alterações desejadas, basta clicar em salvar, para que o banco de dados seja atualizado com as novas informações.

Uma outra diferença entre os formulários de cadastro e atualização de computador é o fato de que é possível alterar a lista de softwares do computador, sem que seja preciso chamar um outro formulário. Essa atualização consiste na vinculação ou desvinculação de um software ao computador.

Os trechos de código executados para realizar todas essas atividades encontram-se discriminados em Anexos-A4, páginas 51 e 52.

#### **6º) Formulários de Atualização de Setor, Sintoma, Técnico, Software e Atividade**



Mais uma vez, esses formulários serão comentados em conjunto, pelo fato de apresentarem a mesma estrutura de visualização e codificação. Como não será permitida a atualização dos códigos, pois estes são gerados internamente pelo sistema, o único campo passível de alteração será o campo nome.

Para iniciar a atividade de atualização, o usuário deverá fornecer o código do elemento (setor, ou sintoma, etc) a ser atualizado. Em seguida, deverá clicar no botão de pesquisa, para que o sistema procure pelo código na devida tabela do banco de dados e possa retornar o campo nome. Campo retornado, o usuário poderá digitar o novo valor que deseja aplicar aquele campo. Depois, clicará no botão Salvar para concluir a atividade. Uma mensagem de confirmação mostrará o antigo e o novo valor do campo e esperará pela confirmação para gravar as alterações no banco de dados.

Alguns trechos do código da função de salvar a atualização estão dispostos em Anexos-A4, página 53.

### **7º) Formulário de Exclusão de Cadastro de Computador**

A exclusão poderá ser efetuada, tão logo seja fornecido o código do computador, cujo cadastro deseja-se remover. Existem duas maneiras de fornecer o código ao sistema:

- Digitar o código no local reservado para tal e depois clicar no botão de pesquisa para verificar se o mesmo é válido. Se o código for encontrado na tabela COMPUTADOR, então o computador encontra-se, realmente cadastrado, podendo, assim, ter o seu cadastro removido, conforme desejado, bastando que o usuário clique no botão Excluir para que a operação seja concluída. Se o código fornecido não for válido, uma mensagem de erro será apresentada na tela; ou
- Já que esse formulário dispõe de uma lista contendo todos os cadastros de computador do sistema, o usuário tem a opção de procurar pelo cadastro que deseja excluir. Uma vez encontrando-o, basta clicar no botão Excluir para removê-lo do sistema.

Uma outra lista também pode ser percebida nesse formulário. Ela foi introduzida para que o usuário possa verificar os softwares que se encontram presentes no computador, cujo cadastro deseja-se remover.

Todo esses cuidados foram tomados para que se tenha o máximo de certeza na hora em que se decidir remover um cadastro do sistema, diminuindo, assim, a probabilidade de excluir o cadastro errado.

Uma última observação deve ser feita para o caso em que se deseje remover um cadastro, cujo computador está vinculado a Ordens de Serviço, tanto fechadas, como em aberto. Nesse caso, o sistema deverá avisar o usuário que a exclusão resultará na perda de informação sobre essas ordens e pedir que ele confirme se deseja prosseguir com a exclusão ou não. Caso o usuário deseje prosseguir, todas as Ordens de Serviço que foram registradas para aquele equipamento serão excluídas juntamente com o seu cadastro.

A título de ilustração, partes da função de pesquisa e de exclusão encontram-se dispostas em Anexos-A4, páginas 53 a 56.

#### **8º) Formulários de Exclusão de Setor, Sintoma, Técnico, Software e Atividade**

Assim como no formulário de exclusão de cadastro de computador, o usuário tem a opção de fornecer o código do elemento que deseja excluir, seja ele um setor, um sintoma ou quaisquer um dos outros acima mencionados, ou então consultar a lista de cadastros, em busca daquele que pretende remover.

Uma vez feita a escolha, deve-se clicar no botão Excluir para que a operação seja concluída. Se houver referências, em outras tabelas do banco de dados, ao código do elemento que se deseja excluir, o sistema avisará da eventual perda de outras informações, caso o usuário prossiga com a remoção e lhe dará a oportunidade de desistir dela. Se ele não desistir, a perda daquelas informações precederá a exclusão do elemento do sistema.

Para melhor entender a explanação acima dada, alguns trechos das funções pesquisar e excluir encontram-se disponíveis em Anexos-A4, páginas 56 a 59.



### **9º) Formulário de Consulta de Cadastro de Computador**

A consulta ao cadastro de computadores pode ser feita de duas formas:

- O usuário fornece o código do computador e o sistema fará a verificação de sua validade. Se for válido, o cadastro será apresentado, senão uma mensagem de erro será lançada.
- O usuário pode utilizar o botão de navegação para fazer uma consulta linear em busca do cadastro de seu interesse. O botão lhe permitirá mover-se para o primeiro ou último cadastro, assim como para o próximo da lista.

À medida que o cadastro de um computador é apresentado, os softwares a ele vinculados são listados, como pode ser percebido através dos trechos da função de consulta fornecidos em Anexos-A4, página 59.

### **10º) Formulários de Consulta de Setor, Sintoma, Técnico, Software e Atividade**

A exemplo do formulário de consulta a cadastros de computador, existem duas maneiras de consultar o cadastro de um dos elementos acima. A primeira é fornecendo o código do elemento e a segunda é através do botão de navegação que permite uma busca linear, pela lista de cadastros, daquele desejado (Vide Anexos-A4, página 60).

### **11º) Formulário de Consulta a Ordens de Serviço Pendentes**

Trata-se de um formulário muito simples, onde serão visualizadas duas listas. A primeira lista apresenta todas as Ordens de Serviço que encontram-se pendentes, lembrando que o critério utilizado para essa classificação, é que a data e a hora do fechamento de uma ordem, ainda não tenham sido gravadas pelo sistema, o que só aconteceria, se o usuário já tivesse informado o fechamento daquela, por intermédio do formulário de Fechamento de Ordens de Serviço.

Dentre as informações que constarão nessa primeira lista, estão:

- Código da Ordem de Serviço;
- Código do Computador ao qual pertence a ordem;
- Nome do Setor ao qual o computador encontra-se vinculado;

- Data e Hora em que a ordem foi registrada.

A segunda lista exporá os nomes dos técnicos, a data e a hora das vinculações e desvinculações que foram estabelecidas para a ordem de serviço que se encontra selecionada na primeira lista.

As consultas feitas ao banco de dados para obter as informações, necessárias às duas listas, são fornecidas em Anexos-A4, páginas 60.

### **12º) Formulário de Consulta a Ordens de Serviço Por Setor**

Por se tratar de uma consulta um pouco mais complexa, o formulário merece uma maior atenção, tanto com relação aos componentes nele existentes, como às consultas realizadas ao banco de dados.

Uma consulta por setor requiere que o usuário forneça três tipos de informação ao sistema. São elas:

- Nome do Setor – Através de uma caixa de seleção, onde estão contidos todos os setores cadastrados no sistema, o usuário selecionará aquele de que deseja obter o relatório resultante da consulta.
- Intervalo da Consulta – Clicando no botão Data, uma pequena janela (um outro formulário) aparecerá, onde o usuário poderá fornecer duas datas, que serão utilizadas como o intervalo dentro do qual a data de registro de uma ordem de serviço deverá encontrar-se, para que possa constar no resultado da consulta.
- Estado da Ordem de Serviço – O usuário deverá selecionar em que estado as ordens de serviço deverão se encontrar para serem selecionadas pela consulta. As três opções disponíveis são: em aberto, fechadas e todas. As ordens de serviço em aberto são aquelas cuja data e hora de fechamento ainda não foram registradas pelo sistema, o que consiste na situação oposta a das ordens fechadas. Caso o usuário não tenha selecionado um estado, o sistema assumirá que a consulta deverá ser feita levando em consideração todas as ordens de serviço, independentemente de seu estado.

De posse de todas essas informações, o sistema fornecerá uma caixa de seleção contendo os códigos dos computadores vinculados ao setor. Selecionando-se um computador, o sistema listará todas as ordens de serviço que já foram registradas para ele, assim como também fornecerá duas outras listas de informações a respeito da ordem que tiver sido selecionada. A primeira delas conterá os nomes dos técnicos que foram vinculados a ordem de serviço, juntamente com a data e a hora em que as vinculações e desvinculações ocorreram. A outra trará os sintomas apresentados, quando da abertura da ordem de serviço.

As consultas utilizadas na obtenção dos dados pertinentes a cada uma das listas encontram-se em Anexos-A4, páginas 60 a 63.

### **13º) Formulário de Consulta a Ordens de Serviço Por Técnico**

Esse formulário tem algumas diferenças, com relação ao descrito no item anterior, mas requer, basicamente, os mesmos tipos de informação do usuário.

Para iniciar a consulta, o usuário deverá selecionar o nome do técnico, fornecer o intervalo de datas e selecionar o estado em que as ordens de serviço deverão encontrar-se. Uma vez que o sistema tem acesso a todos esses dados, ele fornecerá quatro listas de informações. São elas:

- Lista das Ordens de Serviço – Aqui estão listadas as ordens às quais o técnico foi vinculado, juntamente com o código do computador ao qual a ordem pertence, sua data e hora de abertura e de fechamento;
- Lista dos Sintomas – Fornecerá os sintomas apresentados pelo computador da ordem de serviço selecionada na lista acima citada;
- Lista de Vinculações/Desvinculações – Conterá a data e a hora das vinculações e desvinculações estabelecidas entre o técnico e a ordem de serviço selecionada, lembrando que é possível que um técnico tenha sido vinculado a uma mesma ordem por mais de uma vez; e
- Lista das Atividades Executadas – Apresentará uma descrição das atividades que foram realizadas pelo técnico quando vinculado à ordem de serviço selecionada. Também fornecerá a data e a hora em que cada uma

dessas atividades foi registrada, assim como alguma complementação que o técnico tenha feito a respeito daquelas.

Para um melhor entendimento, as consultas utilizadas para gerar essas listas encontram-se discriminadas em Anexos-A4, páginas 63 a 65.

#### **14º) Formulário de Solicitação de Serviço**

Para a abertura de uma ordem de serviço, faz-se necessário que o usuário forneça o código do computador e o nome do setor, além de listar quais são os sintomas apresentados por aquele.

Visando facilitar o fornecimento desses dados, o formulário apresenta três caixas de seleção. A primeira contém o código de todos os computadores cadastrados. A segunda contém os nomes de todos os setores e, a terceira, de todos os sintomas. Dessa forma, é mantida uma padronização com respeito aos nomes de setores e sintomas que são cadastrados.

A listagem dos sintomas ocorre individualmente. O usuário seleciona um sintoma e clica no botão Incluir para incluí-lo na lista de sintomas. Caso ele se arrependa ou tenha cometido um erro ao incluí-lo, o usuário deverá selecioná-lo e clicar no botão Excluir para excluí-lo da lista de sintomas apresentados.

Havendo fornecido todas as informações necessárias, basta clicar no botão Salvar, para que os dados sejam gravados no banco de dados. O código que descreve tal atividade é apresentado em Anexos-A4, páginas 65 e 66.

#### **15º) Formulário de Distribuição do Serviço**

A distribuição do serviço é feita via fornecimento do código da ordem de serviço e do código do técnico ao qual se deseja vincular o serviço. De posse dessa informação, o sistema verificará se os códigos fornecidos são válidos. Caso sejam, será, então, verificado, se o técnico está disponível, ou seja, se ele já não se encontra vinculado a uma outra ordem de serviço. A ordem de serviço também não poderá estar vinculada a um outro técnico. Em qualquer um desses casos, a distribuição não poderia ser feita e uma mensagem de erro seria apresentada na tela. Por outro lado, se todos os requisitos, para que a distribuição possa ser feita



foram obedecidos, os dados serão gravados no banco de dados e o sistema registrará a data e a hora em que a vinculação ocorreu.

Para um melhor entendimento, os trechos de código utilizados, encontram-se disponíveis em Anexos-A4, páginas 66 e 67.

#### **16º) Formulário de Dispensa de Técnico**

Um técnico será dispensado de um serviço, mediante o fornecimento dos códigos da ordem de serviço e do técnico. O sistema testará se os códigos são válidos e se existe, realmente, a vinculação entre o técnico e o serviço. Se tudo der certo, a data e a hora da desvinculação serão gravadas e o técnico estará, assim, dispensado. Se alguma das validações, acima mencionadas, falhar, uma mensagem de erro será exibida na tela (Vide Anexos-A4, páginas 67 e 68).

#### **17º) Formulário de Registro de Atividade**

Para efetuar o registro de uma atividade, o usuário deverá fornecer o código da ordem de serviço, o nome do técnico e o nome da atividade. Uma descrição, complementando a atividade, poderá ser fornecida, visando o melhor entendimento de quem a consultar.

O sistema checará se o técnico está vinculado à ordem de serviço e, se estiver, a atividade poderá ser registrada. Caso contrário, será exibida uma mensagem de erro na tela.

Os trechos do código utilizado para validar e gravar as informações no banco de dados encontram-se disponíveis em Anexos-A4, páginas 68 a 70.

#### **18º) Formulário de Fechamento de Ordem de Serviço**

Trata-se do formulário mais simples, porque ele possui, apenas, uma caixa de seleção, contendo os códigos de todas as ordens de serviço que se encontram em aberto. No momento em que um desses códigos é selecionado, o sistema verificará se existe algum técnico vinculado a ele e que ainda não foi desvinculado. Se houver, o sistema gravará a data e a hora da desvinculação,

dispensando o técnico, para em seguida registrar a data e a hora em que a ordem foi fechada (vide Anexos-A4, páginas 70 a 71).

## 6. Conclusões/Sugestões

Algumas das situações de desconforto ocorreram quando auxiliei alguns colegas em prestações de suporte. Por mais de uma vez, precisei abrir uma CPU para instalar e/ou desinstalar uma peça, o que me acarretava certo nervosismo, pois apesar de ter estudado todos os dispositivos e conhecer, na teoria, todos as peças de hardware, bem como seu funcionamento, a prática constitui-se numa experiência totalmente nova, o que transmite uma sensação de insegurança. Entretanto, depois de ser designada para esse mesmo serviço outras vezes, a insegurança cedeu lugar à experiência e a novos conhecimentos.

O tempo que dedicava ao projeto foi o responsável pela situação de maior conforto por dois motivos simples: o primeiro é que eu já havia me acostumado com projetos ao longo do curso e o segundo é porque todas as atividades, nele envolvidas, faziam parte de áreas de meu interesse e, apesar de ter que revisar todas elas, através da leitura de livros, considerei tudo muito prazeroso.

Com relação às sugestões que deveriam ser apresentadas com respeito a conteúdos e práticas de ensino, eu pensei muito e, atualmente, não tenho nenhuma a oferecer, visto que muita coisa ocorreu nos últimos quatro anos, como, por exemplo, a reforma curricular e a construção do laboratório de hardware, que se constituíam nos objetos das minhas sugestões para melhoramento da qualidade do curso. Sendo assim, ao invés de sugerir, eu prefiro parabenizar o curso de Ciência da Computação pelos melhoramentos e deixar meus votos de que o departamento e a coordenação continuem trabalhando juntos para elevar, cada vez mais, o conceito do nosso curso.

# 7. ANEXOS



# **ANEXOS-A1**

# **ANEXOS-A2**

## Declaração

Declaro para os devidos fins que **Emmanuelle Albuquerque dos Santos Coelho**, apresentou o Relatório Final do Estágio e que o mesmo vem contribuir através de um sistema, a ser implantado, com o gerenciamento do atendimento aos usuários e, por consequência, com a melhoria dos processos que envolvem as atividades de suporte e solução de problemas da área de suporte ao usuário do SEBRAE Paraíba.

João Pessoa, 14 de outubro de 2002.

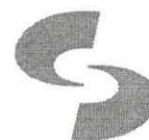


**Valdey Ventura Paulo**  
Gerente de Tecnologia da Informação  
SEBRAE Paraíba

**ANEXOS-A3**

**ANEXOS-A4**

**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE**  
**CENTRO DE CIÊNCIAS E TECNOLOGIA**  
**DEPARTAMENTO DE SISTEMAS E COMPUTAÇÃO**



**UFPA CCT** Departamento de  
Sistemas e  
Computação

PORTARIA DSC/CCT/UFCG Nº 20/2002


**P O R T A R I A**

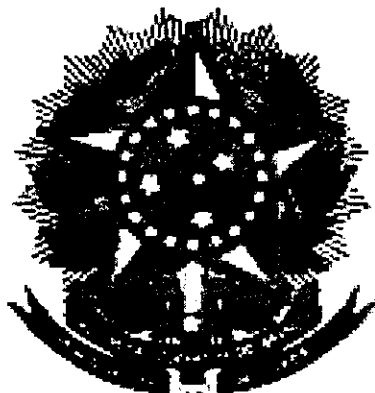
O Chefe do Departamento de Sistemas e Computação do Centro de Ciências e Tecnologia da Universidade Federal de Campina Grande, no uso de suas atribuições,

**R E S O L V E:**

Designar os Professores: HAZENCLEVE FREIRE DE MEDEIROS, matrícula SIAPE Nº 0337791, BRUNO CORREIA DA NÓBREGA QUEIROZ, matrícula SIAPE Nº 0329840 e MARCUS SALERNO DE AQUINO, matrícula SIAPE Nº 0337008, para sob a presidência do primeiro, constituir banca examinadora para julgar a monografia de estágio supervisionado do aluno Luís Benedito Ainsworth Fhaning, matrícula Nº 29621015.

Campina Grande, 10 de outubro de 2002.

  
Profº Bruno C. N. Queiroz  
CHEFE DO DSC  
Mat. 32984-0  
bcnq@dsc.ufpb.br



**RELATÓRIO FINAL DE ESTÁGIO  
DSC/CCT/UFCG**

**Aluno:**

**Adilton Ângelo Seixas Magalhães**

**Orientadora:**

**José Antônio Beltrão Moura, D. Sc.**

**DESENVOLVIMENTO DO SISTEMA:**

**Sistema de Intimação Eletrônica**

<b>1. INDICE.</b>	
2. APRESENTAÇÃO .....	03
3. AMBIENTE DE ESTÁGIO	
3.1. SOFTWARE DE APOIO.....	04
3.2. EQUIPE .....	05
3.3 ASPECTOS POSITIVOS .....	05
3.4 ASPECTOS NEGATIVOS.....	05
4. DESCRIÇÃO DO PROBLEMA .....	06
5. PROPOSTA DA SOLUÇÃO .....	07
6. ATIVIDADES DESENVOLVIDAS .....	08
6.1. CRONOGRAMA.....	08
6.2. ANÁLISE .....	09
6.2.1.REQUISITOS NÃO FUNCIONAIS .....	09
6.2.2. REQUISITOS FUNCIONAIS .....	10
6.2.3. ARQUITETURA.....	10
6.2.4. PLANEJAMENTO DE RELEASE.....	10
6.2.5. PLANEJAMENTO DE ITERAÇÃO.....	11
7. CONCLUSÃO .....	15
8. REFERÊNCIA BIBLIOGRÁFICA .....	16
9. ANEXOS	
ANEXO I. DIAGRAMAS DE CLASSES .....	17
ANEXO II. TELAS DOS SISTEMAS .....	18
ANEXO II. BIGCHART .....	27
ANEXO IV. PLANO DE ESTÁGIO .....	28
ANEXO V. DECLARAÇÃO DE APROVAÇÃO DO ESTÁGIO.....	31



## 2. APRESENTAÇÃO

A política da qualidade adotada pela Subseção Judiciária de Campina Grande foi desenvolvida em consonância com a missão, visão e valores estabelecidos no Planejamento Estratégico da Justiça Federal da 5ª Região, e consiste em:

***"Promover a Justiça por meio da melhoria contínua da prestação jurisdicional, com celeridade, efetividade e transparência".***

Esta política tem os seguintes objetivos:

- Melhorar a qualidade do atendimento prestado aos jurisdicionados;
- Desenvolver uma liderança participativa (Gestão à Vista);
- Melhorar continuamente os processos de forma sistêmica;
- Detectar e combater re-trabalhos;
- Acelerar o processamento dos feitos;
- Melhorar a capacitação dos servidores;
- Melhorar a credibilidade das informações prestadas (virtual e física);

Com o objetivo de atender a esta política de qualidade, visto a necessidade de melhorias no campo da informática, firmou-se um convênio com a UFCG, especialmente com o DSC onde há pessoas capacitadas a atender as necessidades requeridas pelo setor de informática da Justiça Federal.

Desta forma, após firmado o convênio com a universidade, requisitou-se um estagiário de informática para promover a melhoria dos serviços prestados pela Justiça Federal em Campina Grande. Incentivou-se o desenvolvimento de um Sistema que automatizasse as intimações judiciais, sendo analisada a melhor conduta para o desenvolvimento deste sistema que pudesse evoluir de forma ágil e em sintonia com as exigências requeridas pelo método tradicional.

No aspecto acadêmico, o projeto Sistema de Intimações Eletrônicas foi considerado ideal para o aprimoramento dos conhecimentos adquiridos na graduação em Ciência da Computação, pois engloba conhecimentos das disciplinas Sistemas de Informações I e II, Métodos Avançados de Programação, Banco de Dados, Engenharia de Software. Além disso, foi importante para a aquisição de novos conhecimentos não abordados na universidade, tal como enfrentar um problema real em um ambiente Corporativo.

Os demais itens deste documento relatam, com detalhes, o desenvolvimento do Sistema de Intimação Eletrônica, cujo alvo do estágio é o desenvolvimento de um sistema Web para consulta de intimações. O capítulo 3 descreve o ambiente de estágio. O capítulo 4 descreve o problema do estágio. O capítulo 5 apresenta a solução do problema. O capítulo 6 mostra a evolução do projeto discriminando todas as atividades do projeto. Este capítulo foi dividido em quatro seções, nas quais está descrito o cronograma do projeto. O capítulo 7 apresenta a conclusão do estágio. O capítulo 8 lista as referências bibliográficas. O capítulo 9 contém os anexos do relatório. O Anexo I contém os diagramas de classes. O Anexo II contém as telas dos sistemas. O Anexo III mostra a evolução do projeto através de um Bigchart. O Anexo IV Plano de estágio e o Anexo V a aprovação do estágio.

Platipau

### 3. AMBIENTE DE ESTÁGIO.

O estágio foi desenvolvido nas instalações do setor de informática da Justiça Federal de Campina Grande, localizado no térreo do prédio da Justiça Federal, no endereço abaixo:

Justiça Federal da Paraíba,  
Subseção Campina Grande  
Setor de Informática - térreo,  
Rua Edgard Vilarim Meira, s/nº  
Liberdade CEP: 58105-000  
Campina Grande / PB - Brasil  
PABX: (83)310-9100 FAX: (83)310-9104

Atualmente, este setor é composto de quatro máquinas, das quais apenas 2 foram utilizadas para o desenvolvimento do projeto, pois as outras duas são utilizadas como servidores.

1. Máquina de Desenvolvimento 1: Destinada para o uso do desenvolvimento e teste do sistema, entre outras atividades realizadas pelo setor de Informática, também era utilizada como servidor web, mantém o SGBD usado no projeto (MySQL). Configuração: Pentium 4, 2.8 Ghz, 256 MB de memória RAM, 40 GB de Hard Drive, placa de rede 3Com 10/100 Mbps, placa mãe SIS. Sistema Operacional: Windows Server 2003.

2. Máquina de Desenvolvimento 2: Destinada para o uso do desenvolvimento e teste do sistema, entre outras atividades realizadas pelo setor de Informática, também era utilizada como servidor web, mantém o SGBD usado no projeto (MySQL). Configuração: Pentium 4, 2.4 Ghz, 256 MB de memória RAM, 40 GB de Hard Drive, placa de rede 3Com 10/100 Mbps, placa mãe SIS. Sistema Operacional: Windows 2000.

#### 3.1. SOFTWARE DE APOIO

Os softwares utilizados como suporte no desenvolvimento deste projeto foram os seguintes:

- **Jakarta Tomcat 5.0**: Web Container da Apache dá suporte ao uso de tecnologias Java para desenvolvimento web [TOMCAT].
- **Jakarta ANT**: Ferramenta de integração e montagem de código [ANT].
- **MySQL 4.0**: SGBD utilizado para persistência dos dados [MYSQL, 2004].
- **Poseidon**: Ferramenta de modelagem UML [POSEIDON, 2004].
- **Eclipse**: Ambiente de programação em Java [ECLIPSE, 2004].
- **ER-WIN 4.0**: Ferramenta utilizada para visualizar a base de dados da Justiça.

### 3.2 EQUIPE

O projeto foi desenvolvido pelo estagiário Adilton Ângelo e coordenado por Francisco Hertz, com o intuito de cumprir as exigências estabelecidas pela disciplina Estágio Integrado e atender a necessidade do Foro da Justiça Federal.

Sendo assim, o estagiário participou de todos os papéis no desenvolvimento do projeto.

**Analista de Sistema** (Responsável pela geração do modelo conceitual e análise de requisitos):

- Adilton Ângelo

**Gerente do projeto** (Responsável pela interface entre a equipe de desenvolvimento e os analistas de negócio, controla o andamento do projeto, checa cronogramas):

- Francisco Hertz
- Adilton Ângelo

**Gerente de Configuração** (Mantém os sistemas de suporte ao desenvolvimento)

- Francisco Hertz (Servidor de aplicação, manutenção da rede),

**Engenheiro de Software** (Analisa o problema, provê uma solução, implementa, documenta, testa e publica).

- Francisco Hertz
- Adilton Ângelo

**Web Designer** (Responsável pelo layout do sistema)

- Adilton Ângelo

### 3.3. ASPECTOS POSITIVOS

Houve vários aspectos que contribuíram para o bom andamento das atividades: ambiente de trabalho bastante adequado; equipamento de boa qualidade facilitando e agilizando o desenvolvimento do projeto (computador de última geração); privilégio de trabalhar com um supervisor técnico mais experiente na área de computação e a oportunidade de desenvolver, gerenciar, ser engenheiro de software e web Designer. Além de utilizar um processo de desenvolvimento de software, XP1 no projeto.

1. XP1: Processo de Desenvolvimento desenvolvido no DSC, baseado nas práticas de Extreme Programming (XP), mas com algumas mudanças e simplificações. Este processo é particularmente indicado quando os requisitos funcionais do software a ser desenvolvido não estão muito bem definidos, pois permite fazer mudanças em ciclos pequenos e com baixo custo de mudança.

### 3.4. ASPECTOS NEGATIVOS

Apesar do sucesso deste projeto, destacamos alguns aspectos negativos: excesso de trabalho sobre uma única pessoa, impossibilitando comentários e revisão por outra pessoa, como por exemplo "par programming"; interrupção do projeto para atender outras necessidades no setor de informática; tempo insuficiente para efetuar um teste de carga, ou mesmo disponibilizar o sistema para uso interno.

## 4. DESCRIÇÃO DO PROBLEMA.

Em decorrência do princípio da ampla defesa, assegurado pela Constituição Federal, é imprescindível que sejam os acusados cientificados da existência do processo e de todo o seu desenvolvimento. Ninguém pode ser processado ou condenado sem que tenha ciência da acusação que se lhe faz, das alegações da parte acusadora, das provas produzidas e das decisões exaradas nos autos. Essa ciência é feita através da citação, em que se cientifica o acusado da imputação, e das intimações e notificações, em que se lhe comunicam os atos do processo passado e futuro. Desta forma, intimação é o ato pelo qual se dá ciência a alguém dos atos e termos do processo, para que ele faça ou deixe de fazer alguma coisa, normalmente esse alguém é um advogado que representa um cliente.

Assim, para obter segurança na remessa e no recebimento das informações, entre a Justiça e os advogados, no quesito intimação, reduzir as despesas com hora-trabalho/servidor e gastos resultantes das intimações por telefone/fax e/ou carta, requisitou-se a implementação de um sistema que permitisse encontrar os advogados no tempo hábil em qualquer lugar do país ou do mundo, para que o mesmo tivesse acesso à (às) intimação (ões) a ele destinadas.



## 5. PROPOSTA DA SOLUÇÃO.

Com o objetivo de atender às metas estabelecidas pela Justiça Federal, foi proposto pelo setor de informática desta subseção um Sistema de Intimações Eletrônicas, que será um serviço oferecido pela Justiça Federal de Campina Grande aos advogados que desejam agilizar o recebimento de intimações oriundas das Varas Federais.

Através dele o advogado receberá intimações pela Internet, bastando conectar-se ao sistema (link ENTRAR). Para utilizá-lo, o advogado deverá efetuar seu cadastramento junto a qualquer Vara da Justiça Federal que esteja oferecendo o serviço de Intimação Eletrônica, oportunidade na qual assinará um Termo de Compromisso.

As intimações eletrônicas são consideradas como realizadas a partir da data em que o advogado efetua a consulta, iniciando-se a contagem do prazo no primeiro dia útil seguinte. Em não havendo consultas na semana, o sistema efetuará a intimação automaticamente às 6<sup>as</sup> feiras - 18:00 horas, ou no último dia útil da semana, no mesmo horário.

Os requisitos funcionais do sistema serão descritos de forma detalhada no capítulo 5.

## 6. ATIVIDADES DESENVOLVIDAS.

Este capítulo contém a descrição das atividades desenvolvidas durante o estágio. Neste será detalhado o desenvolvimento das soluções propostas no capítulo anterior.

Todo o projeto foi gerenciado pelo processo de Desenvolvimento XP1. Portanto as atividades serão especificadas através do planejamento de Release<sup>1</sup> e planejamento de Iteração<sup>2</sup>.

### 6.1. CRONOGRAMA DE ATIVIDADES.

O estágio teve dedicação de 20 horas semanais, no Foro da Justiça Federal, nos seguintes horários programados: Segunda: 14:00 às 18:00; Terça: 14:00 às 18:00; Quarta: 14:00 às 18:00; Quinta: 14:00 às 18:00; Sexta: 08:00 às 12:00.

O trabalho foi dividido em três etapas: a primeira etapa correspondente à análise, que demandou quase um terço do tempo utilizado no desenvolvimento do projeto, devido à ausência de conhecimento sobre o assunto (Intimações); a segunda etapa que compreendeu o período de desenvolvimento, por sua vez dividido em duas partes: o desenvolvimento propriamente dito e o web designer do sistema. Segue abaixo um cronograma macro, onde temos tempo estimado e tempo gasto, respectivamente.

	Novembro 2003	Dezembro 2003	Janeiro 2004	Fevereiro 2004	Março 2004	Abril 2004
<b>Análise</b>						
Estimado						
Gasto						
<b>Desenvolvimento</b>						
Estimado						
Gasto						
<b>Web Designer</b>						
Estimado						
Gasto						

Para cada etapa, correspondem as datas de início e fim.

Análise: 05/11/2003 a 02/12/2003.

Desenvolvimento: 02/12/2003 a 12/03/2003.

Web Designer: 12/03/2003 a 30/04/2004.

1. Release: Um release é a liberação e instalação do software em ambiente externo à equipe de desenvolvimento.

2. Iteração: Em XP1, a iteração é a unidade de planejamento detalhado. Uma iteração dura exatamente duas semanas, sempre. O tempo é fixo, o escopo do que será feito na iteração é variável.[XP1, 2004].

## 6.2. Análise

Nesta seção estão descritas as atividades referentes à análise do projeto, Sistema de Intimação Eletrônica.

### 6.2.1. Requisitos Não Funcionais

- Requisito de Interface: Considerando que o usuário final é um usuário leigo, um advogado ou um servidor da Justiça, há a necessidade de utilizar interface intuitiva e simples, interface Web.
- Requisitos de Hardware / Software: O sistema deve rodar em qualquer máquina que tenha um navegador web, internet Explorer, Netscape, etc.
- Requisito de Linguagem: A linguagem adotada para o desenvolvimento do software é a linguagem Java, para possibilitar a portabilidade do sistema;
- Desempenho: O sistema não deve demorar muito, mas como não foram realizadas as estatísticas, já que se tratava de um sistema web, o desempenho dependerá da velocidade da conexão.
- Documentação Necessária: O sistema deve apresentar documentação da API através do utilitário *Javadoc*.
- Requisito de Facilidade de Uso Necessário: Requer-se que o sistema seja fácil de usar, com o mínimo de intervenção da equipe de desenvolvimento.
- Documentação: O sistema deve apresentar documentação das classes Java através do *Javadoc*. É necessária ainda uma documentação complementar descrita em um manual do usuário para utilização do sistema, já que serão usuários leigos que utilizarão o sistema.
- Tolerância à Falhas: Usaremos a tolerância à falhas implementada pelo SGBD (Mysql);
- Segurança: Só têm acesso ao Sistema de Intimação Eletrônica os usuários cadastrados no Sistema. Este acesso é restrito através de login e senha.



### 6.2.2. Requisitos Funcionais

**Busca:** O sistema deve retornar um conjunto de intimações, cujo responsável é o advogado que utiliza o sistema no devido momento. A busca poderá ser feita por período de datas.

**Tipos de Usuários:** O sistema deve apresentar dois tipos de usuários: advogados, e administradores, que são os oficiais de justiça. Cada usuário deverá atuar apenas em sua área correspondente.

**Cadastro de Áreas de Intimações:** O administrador é o único usuário que tem permissão para cadastrar as intimações.

**Cadastro de Usuários:** O advogado pode cadastrar-se, mas é necessário assinar um termo de responsabilidade na Justiça Federal.

**Visualização de Documentos:** Os advogados portados de seu login e senha podem visualizar as intimações dirigidas a eles.

### 6.2.3. Arquitetura

Como o sistema foi desenvolvido com a tecnologia Java/JSP, a Arquitetura utilizada é a arquitetura três camadas. Veja logo abaixo na Figura 1 a com será a arquitetura. SIE (Sistema de Intimação Eletrônica).

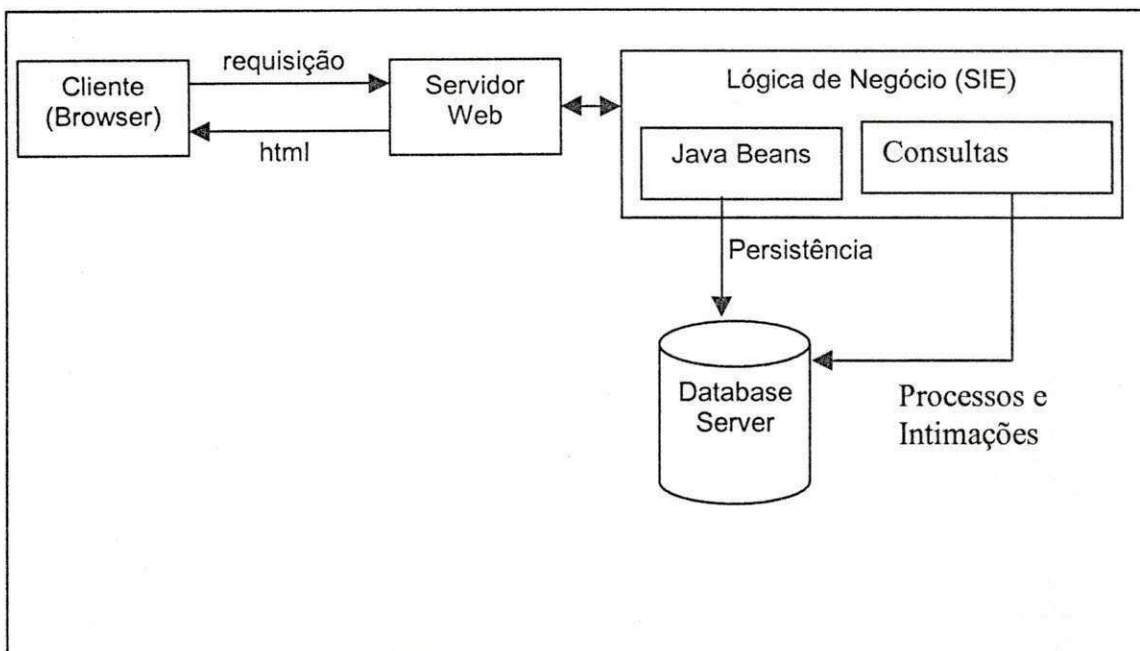


Figura 2. Projeto Arquitetural SIE

### 6.2.4. Planejamento de Release

Nesta seção, está descrito o planejamento de "Release" do Sistema Intimação Eletrônica.

Release 1: Definir o esquema lógico do Banco de Dados, implementá-lo e povoá-lo com dados para testes.

Início: 02/12/2003 Fim: 30/12/2003

Release 2: Entregar o sistema, permitindo cadastrar usuários, lançar intimações e consultá-las.

Início: 30/02/2004 Fim: 16/02/2004

Release 3: Entregar o projeto, permitindo consultas avançadas, alteração de cadastro de usuários e intimações automáticas.

Início: 16/02/2004 Fim: 29/03/2004

Release 4: Entregar o produto com um manual, um help online e uma interface simples.

Início: 29/03/2004 Fim: 30/04/2004

### 6.2.5. Planejamento de Iteração

Nesta seção está descrito o planejamento de Iteração das quatro release.

1ª Release: Foi dividida em duas iterações, que são detalhadas a seguir.

Iteração 1: Esta iteração teve início no dia 02/10/2004 com duração duas semanas. Esta iteração foi caracterizada pela criação do esquema lógico da camada de persistência.

User Story	Estim.	Gasto
US. 01: Definir o esquema lógico do Banco de Dados	10	12

Iteração 2: Essa iteração teve início no dia 16/12/2003. Foi utilizado o SBGB Mysql por questões de limitação de um SGBD de maior porte, e também por ele ser um SGBD free. Também teve duração de duas semanas.

User Story <sup>1</sup>	Estim.	Gasto
US. 02: Implementar o Banco de Dados	6	7
US. 03: Povoar o Banco de Dados	3	4

1. User Story: Cenário descrito pelo usuário capaz de realizar o levantamento de requisitos. [XP1, 2003].

**SUMÁRIO**

O total de horas estimadas para esta release foi de: 19

O total de horas gasta: 23

Erro de estimava: 21% a mais do tempo planejado.

2ª Release: Está descrito o planejamento das Iterações 3,4 e 5, correspondente à segunda release.

Iteração 3: Esta iteração teve início no dia 30/12/2004 com duração duas semanas. Nesta iteração foi desenvolvido um sistema de login para os usuários.

User Story	Estim.	Gasto
US. 04: Implementar um sistema de login.	15.0	17.0

Iteração 4: Essa iteração teve início no dia 19/01/2004 com duração de duas semanas, com a implementação do sistema de cadastro de usuários e cadastro de intimações.

User Story	Estim.	Gasto
US. 05: Implementar o cadastro usuários	18	19
US. 06: Efetuar o lançamento de intimações.	15	14

Iteração 5: Essa iteração teve início no dia 02/02/2004 com duração de duas semanas. Nesta iteração foi desenvolvido o módulo de consulta a intimações, consulta estas feitas por usuários comuns. Algumas mudanças foram necessárias no esquema lógico para permitir a apresentação de uma intimação.

User Story	Estim.	Gasto
US. 07: Consultar as Intimações	20.0	30.0



**SUMÁRIO**

O total de horas estimadas para esta release foi de: 68

O total de horas gasta: 80

Erro de Estimativa: 17.7% a mais do tempo planejado.

3ª Release: Está descrito o planejamento das Iterações 6,7 e 8, correspondente à terceira release.

Iteração 6: Essa iteração teve início no dia 16/02/2004 com duração de duas semanas. Implementou-se a alteração de cadastro de usuários e a intimação automática de usuários, no caso advogados.

User Story	Estim.	Gasto
US. 09: Implementar a alteração de dados de usuários.	7	8
US. 10: Construção do módulo de intimações automáticas.	12	14

Iteração 7: Esta iteração teve início no dia 01/03/2004 com duração de duas semanas. Desenvolveu-se uma busca mais avançada a intimações, permitindo desta forma que usuários avançados possam visualizar quais intimações foram lidas e em que data elas foram excluídas.

User Story	Estim.	Gasto
US. 08: Permitir uma pesquisa avançada a intimações.	8.0	10.0

Iteração 8: Essa iteração teve início no dia 12/03/2004 com duração de duas semanas. Nesta iteração foi desenvolvido o módulo de consulta a usuários.

User Story	Estim.	Gasto
US. 11: Implementar a consulta a usuários cadastrados.	9	10

**SUMÁRIO**

O total de horas estimadas para esta release foi de: 36

O total de horas gasta: 42

Erro de Estimativa: 16.7% a mais do tempo planejado.

4ª Release: Está descrito o planejamento das Iterações 9,10 e 11.

Iteração 9: Esta iteração teve início no dia 29/03/2004 com duração duas semanas. Nesta iteração foi construído um módulo que permite emitir relatórios sobre as intimações vistas.

User Story	Estim.	Gasto
US. 12: Permitir a visualização de relatórios tipo folha.	12	11

Iteração 10: Essa iteração teve início no dia 12/04/2004 com duração de três semanas. Esta última iteração foi voltada ao Designer da página e documentação, inclusive a criação de um manual de usuário.

User Story	Estim.	Gasto
US. 13: Formatar todas as páginas JSP	18	20
US. 14: Fazer um manual para o sistema.	12	15

## SUMÁRIO

O total de horas estimadas para esta release foi de: 42

O total de horas gastas: 45

Erro de Estimativa: 7% a mais do tempo planejado.

## 7. CONCLUSÃO.

Analisando este projeto de estágio, percebe-se que houve aproveitamento tanto da parte da empresa quanto, principalmente, da parte do estagiário. Houve algumas dificuldades, que se tornaram parte do aprendizado, porém, em geral, foi obtido êxito no desenvolvimento da solução e em seu uso.

Ilustrando os pontos benéficos para a empresa e o estagiário, seguem-se:

Para a empresa:

- A experiência com um sistema novo e moderno, um sistema Web, ainda não experimentado pelos servidores da Justiça, e pelos jurisdicionados, neste quesito intimação.
- Possibilidade de redução de custos com telefone/fax e deslocamento de oficiais.

Para o estagiário:

- A experiência adquirida com o desenvolvimento de um projeto para uma grande empresa enriqueceu o conhecimento deste.
- Colocar em prática os conhecimentos adquiridos nas disciplinas de programação e engenharia de software, como nunca havia feito antes.
- Aprendizagem sobre como fazer um bom design para uma página web
- Experiência profissional (no tocante a programação) junto à equipe mais experiente.

Apesar de todos estes benefícios, muitas foram as dificuldades encontradas pelo estagiário no desenvolvimento deste projeto, tais como: havia pouco tempo para dedicação ao projeto, devido à solicitação do estagiário para sancionar problemas diversos na área de informática, durante o expediente; falta de experiência no desenvolvimento de web sites que usam recursos mais avançados do que o usuário estava acostumado.

Gostaria de sugerir que a coordenação, juntamente com os professores, procurasse firmar mais convênios com empresas, para que mais alunos possam ingressar em um estágio. E para os alunos, que eles se interessem mais por um estágio, pois é de grande valia a aprendizagem adquirida no mesmo. E para os estagiários, que possam buscar explorar o máximo possível no seu estágio, pois são grandes as oportunidades surgidas durante o mesmo.

**9. REFERÊNCIA BIBLIOGRÁFICA.**

[TOMCAT, 2004] - [www.jakarta.org.br](http://www.jakarta.org.br)

[ANT, 2003] - [www.jakarta.org.br](http://www.jakarta.org.br)

[MYSQL, 2004] - [www.mysql.org](http://www.mysql.org)

[ECLIPSE, 2004] - [www.eclipse.org](http://www.eclipse.org)

[ER-WIN 4.0 ] - [www.microsoft.com](http://www.microsoft.com)

[POSEIDON 2.3.1] - [www.gentleware.com](http://www.gentleware.com)



## 10. ANEXOS.

### 10.1. ANEXO I. DIAGRAMA DE CLASSES

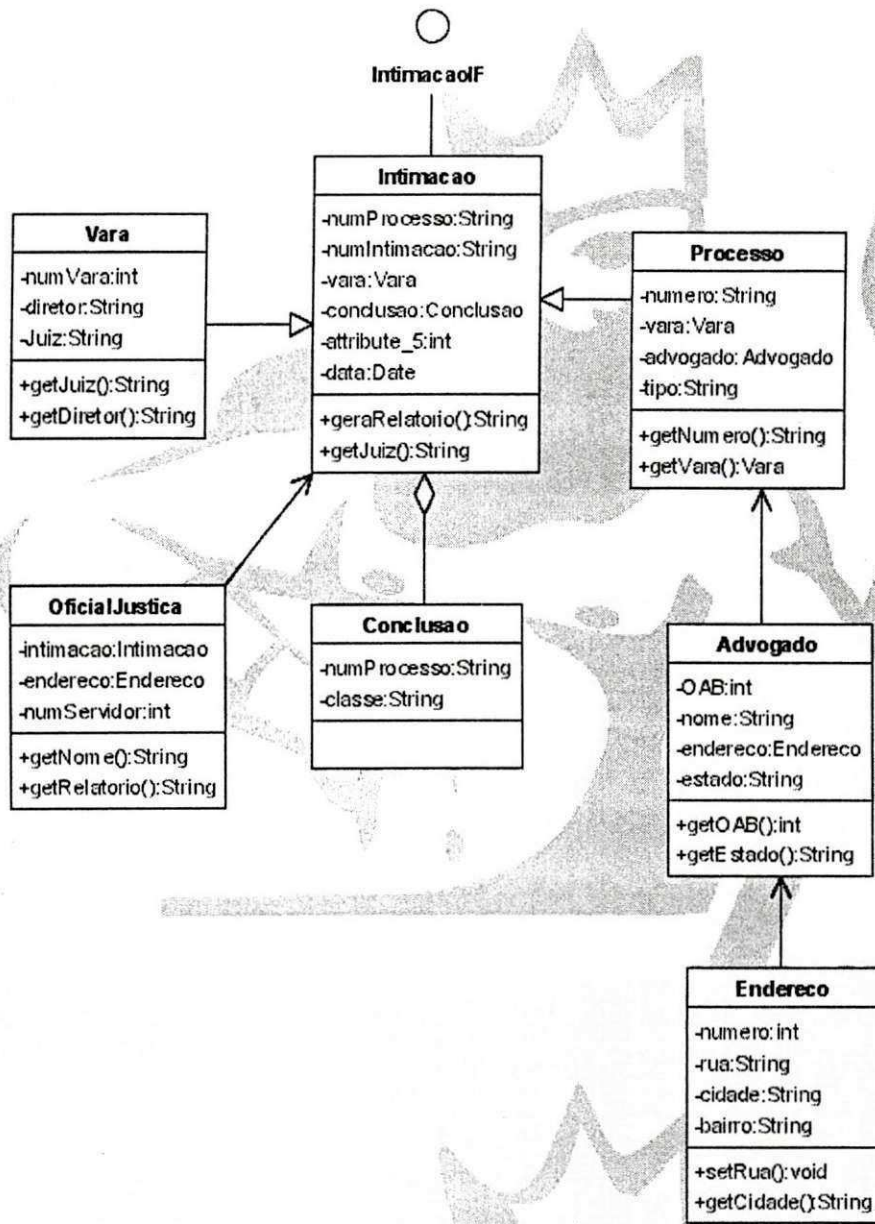


Diagrama de Classes

## 10.2. ANEXO II. TELAS DOS SISTEMAS

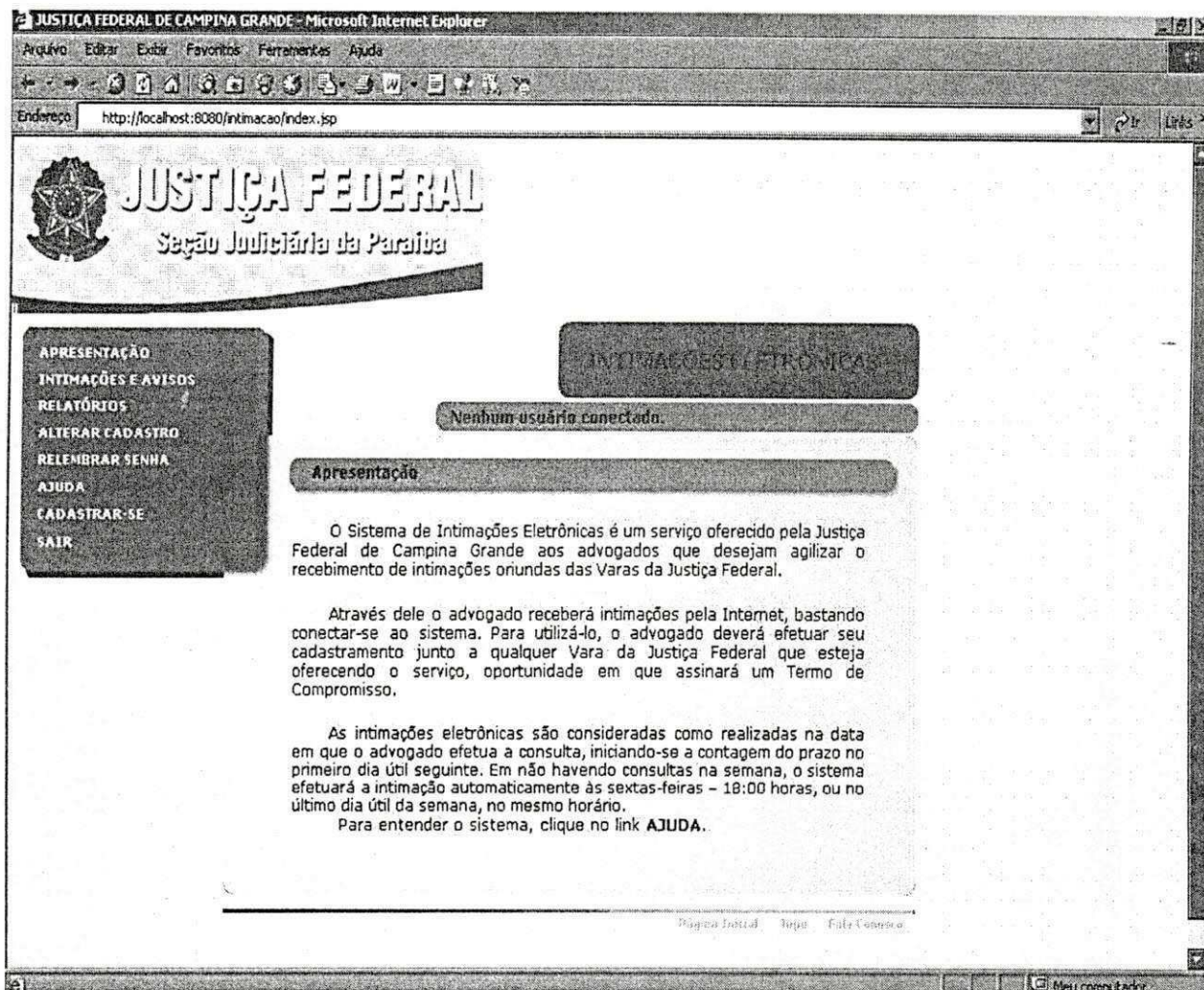


Figura 1: Tela Inicial ( Apresentação )



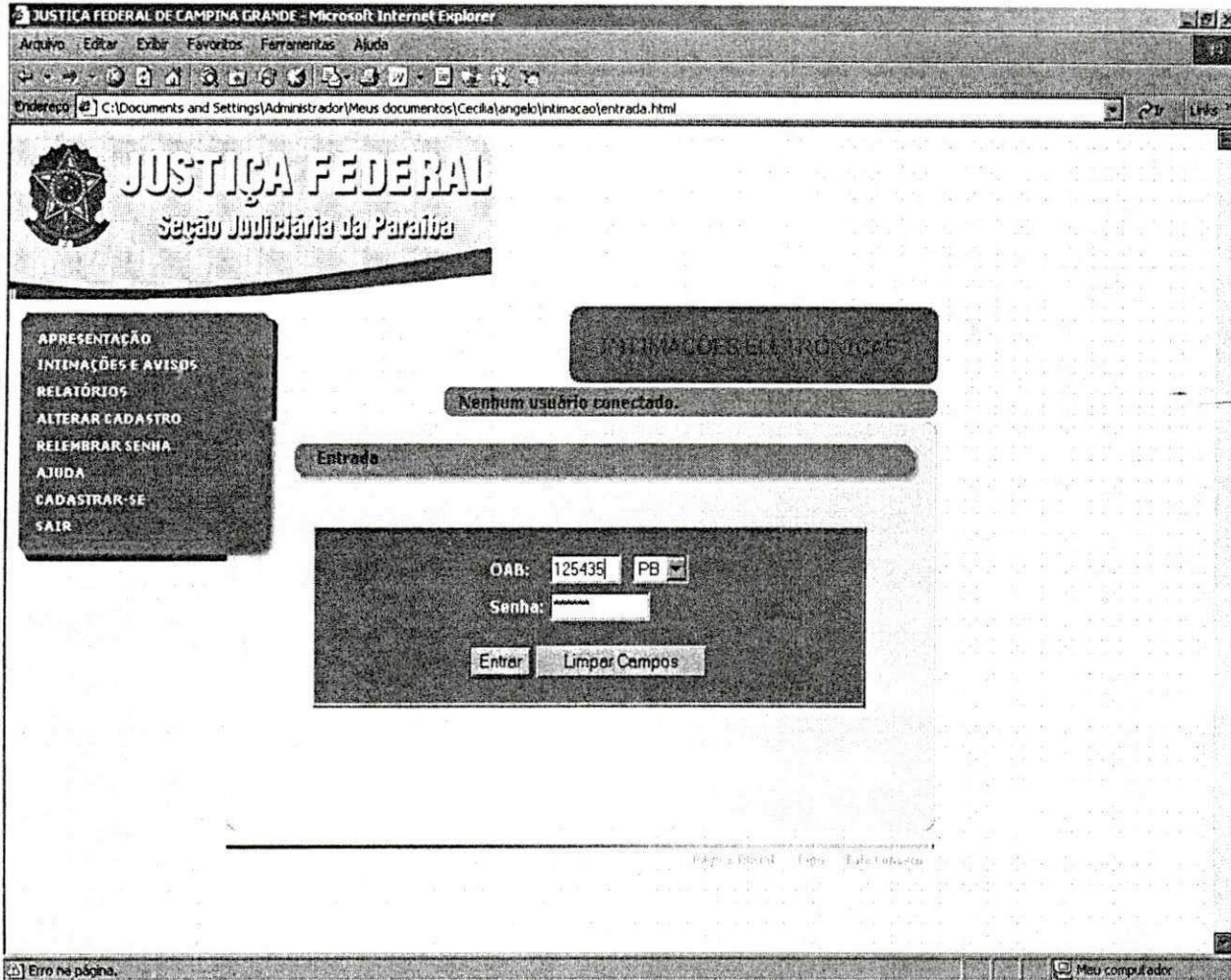


Figura 2: Entrada

JUSTIÇA FEDERAL DE CAMPINA GRANDE - Microsoft Internet Explorer

Arquivo Editar Exibir Favoritos Ferramentas Ajuda

Endereço http://localhost:8080/intimacao/intimacoes.jsp

**JUSTIÇA FEDERAL**  
Seção Judiciária da Paraíba

APRESENTAÇÃO  
INTIMAÇÕES E AVISOS  
RELATÓRIOS  
ALTERAR CADASTRO  
RELEMBRAR SENHA  
AJUDA  
CADASTRAR-SE  
SAIR

INTIMAÇÕES E AVISOS

Usuário: teste

Intimações

**Nova Intimações:**  
(Considere-se intimado neste instante: 04/05/2004, às 14h e 25 min)

**Processo: 2004.75.00.075488-7 Campina Grande**

Autor: MARIA DE LOURDES DA SILVA  
Réu: INSTITUTO NACIONAL DO SEGURO SOCIAL -  
INSS Processo  
2004.82.03.056995-4 6ª Vara Federal de Campina Grande,  
PB setença. Maria de Lourdes da Silva ajuizou ação contra  
o Instituto Nacional do Seguro Social - INSS objetivando  
a concessão de benefícios de amparo social para  
deficientes, desde o requerimento administrativo. 1.1  
O INSS, em contestação, destacou que a renda mensal  
obtida pela companheira da mesma, ultrapassava o valor

Imprimir

Para mostrar outras intimações, informa a data ou o período desejado:

a

Figura 3: Intimações e avisos.



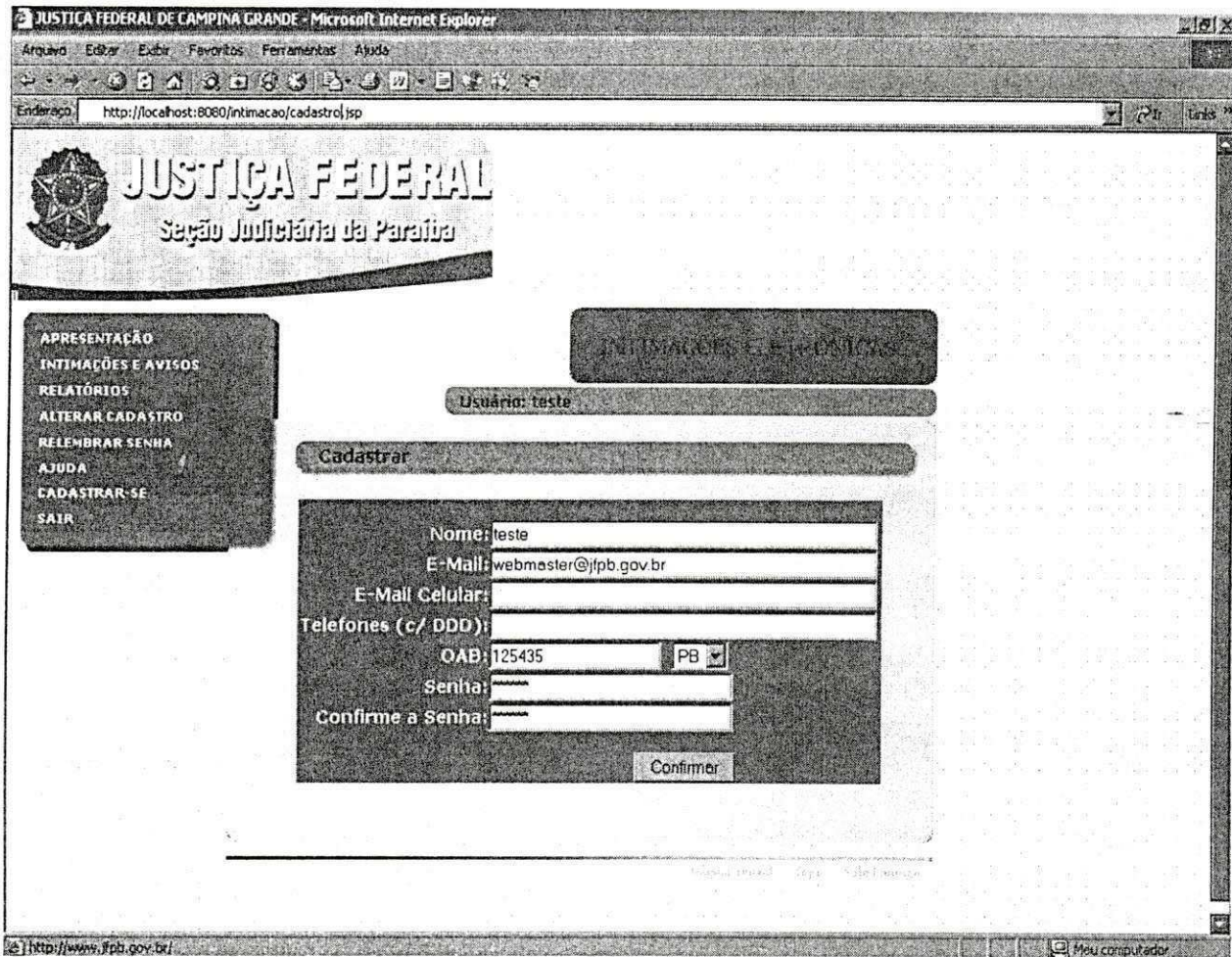


Figura 4: Cadastro

JUSTIÇA FEDERAL DE CAMPINA GRANDE - Microsoft Internet Explorer

Arquivo Editar Exibir Favoritos Ferramentas Ajuda

Endereço: http://localhost:8080/intimacao/alteracadastro.jsp

**JUSTIÇA FEDERAL**  
Seção Judiciária da Paraíba

APRESENTAÇÃO  
INTIMAÇÕES E AVISOS  
RELATÓRIOS  
ALTERAR CADASTRO  
RELEMBRAR SENHA  
AJUDA  
CADASTRAR-SE  
SAIR

INTIMAÇÕES E AVISOS

Usuário: teste

Alterar Cadastro

Nome: teste  
E-Mail: webmaster@jfpb.gov.br  
E-Mail Celular:  
Telefones (c/ DDD): 123456  
OAB: 125435 PB  
Senha:  
Confirme a Senha:

Alterar Limpar Campos

Meu computador

Figura 5: Alterar Cadastro



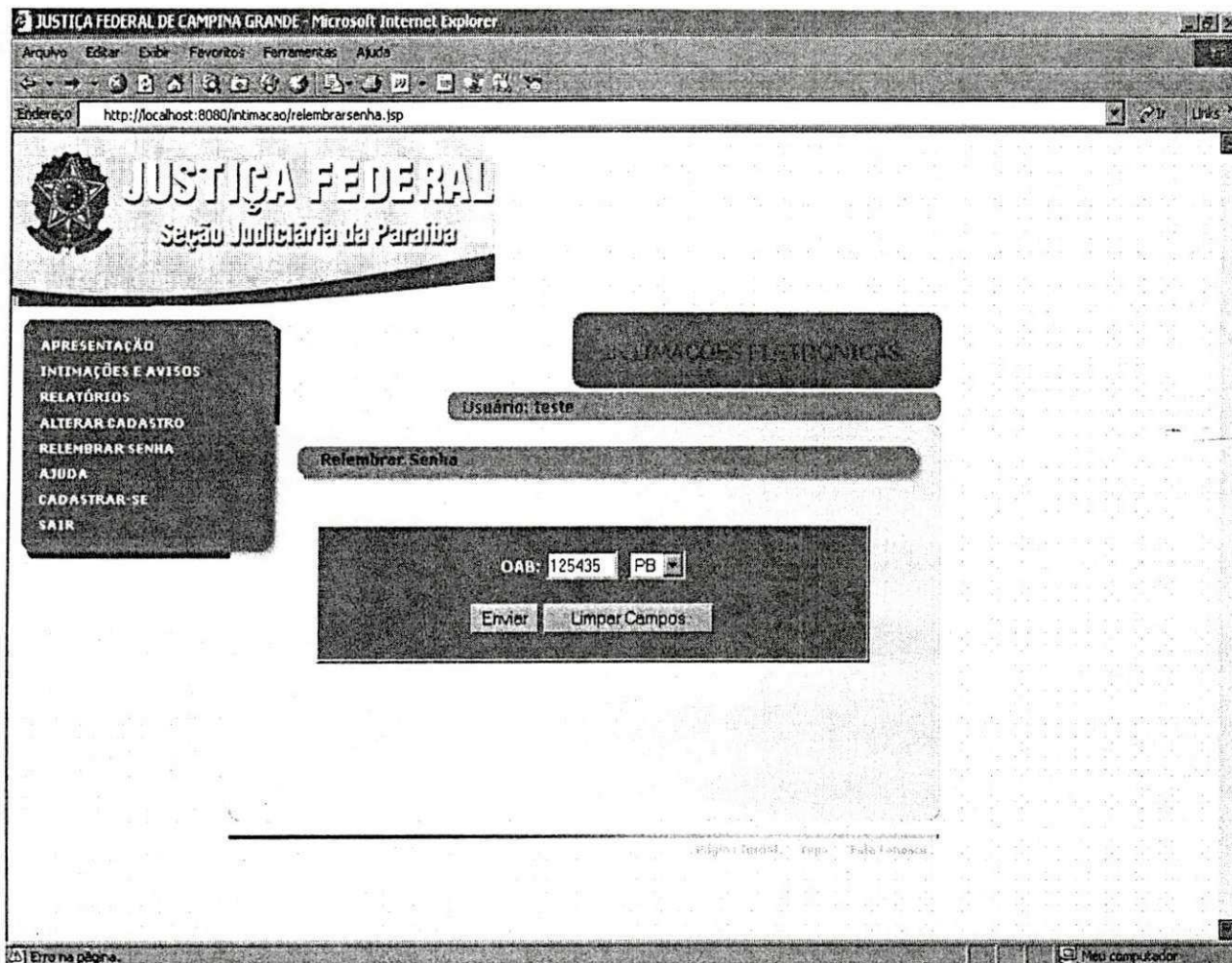


Figura 6: Relembra Senha



Untitled Document - Microsoft Internet Explorer

Arquivo Editar Exibir Favoritos Ferramentas Ajuda

Endereço http://localhost:8080/intimacao/relintimacoes.jsp

### Sistema de Intimações Eletrônicas - Relatório

[Voltar](#)

EXIBIR:	<input checked="" type="checkbox"/> Lidas			
	<input type="checkbox"/> Ainda Não Lidas			
	<input type="checkbox"/> Excluídas Antes da Leitura			
ADVOGADO:	TODOS			
INÍCIO EM:		até		(dd/mm/aaaa)
INCLUIDAS EM:	22/04/2004	a	24/04/2004	(dd/mm/aaaa)
PROCESSO:				
TIPO DE RELATÓRIO:	<input type="radio"/> Folhas	<input type="radio"/> Etiquetas (De		
	<input type="button" value="Executar"/>			

Figura 7: Relatório das Intimações

Microsoft Internet Explorer  
Arquivo Editar Exibir Favoritos Ferramentas Ajuda  
Endereço: http://localhost:8080/intimacao/relatorio.jsp

Voltar Imprimir

**Sistema de Intimações Eletrônicas - Relatório**  
6ª Vara Federal de Campina Grande - PB  
Lidas no dia 24/04/2004

Inclusão	Leitura	Processo Advogado
22/04/2004	24/04/2004	<a href="#">2004.82.03.056995-4 teste1</a> - 125435/PB
22/04/2004	24/04/2004	<a href="#">2004.82.03.056904-6 teste2</a> - 545423/PB
22/04/2004	24/04/2004	<a href="#">2004.82.03.056933-7 teste3</a> - 247541/PB
23/04/2004	24/04/2004	<a href="#">2004.82.03.056985-2 teste4</a> - 352361/PB
23/04/2004	24/04/2004	<a href="#">2004.82.03.056932-1 teste5</a> - 269145/PB
23/04/2004	24/04/2004	<a href="#">2004.82.03.056927-9 teste6</a> - 896253/PB
23/04/2004	24/04/2004	<a href="#">2004.82.03.056977-3 teste7</a> - 132569/PB
24/04/2004	24/04/2004	<a href="#">2004.82.03.056944-7 teste8</a> - 856747/PB

Mesa computador

Figura 8: Relatório

JUSTIÇA FEDERAL  
Seção Judiciária da Paraíba

APRESENTAÇÃO  
INTIMAÇÕES E AVISOS  
RELATÓRIOS  
ALTERAR CADASTRO  
RELEMBRAR SENHA  
AJUDA  
CADASTRAR-SE  
SAIR

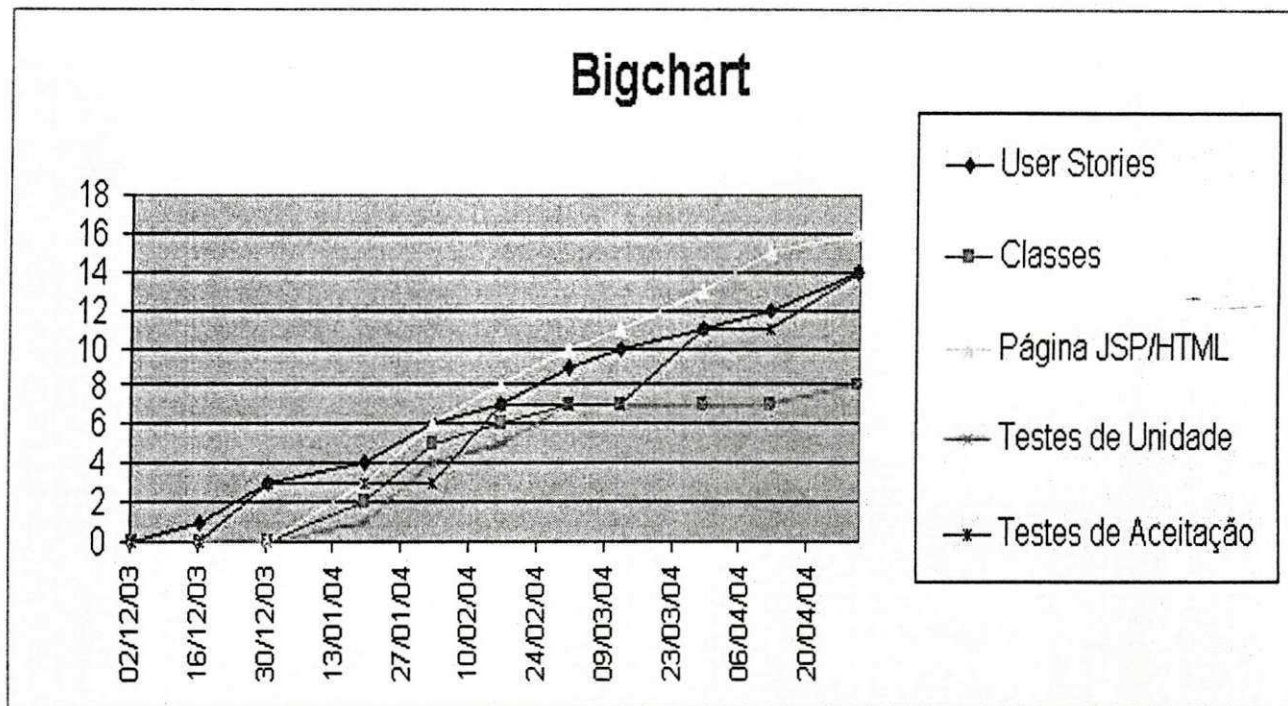
Nenhum usuário conectado.

Ajuda

Item	Função
Intimações e Avisos	Mostrar as intimações normais, as urgentes e os avisos para o advogado conectado.
Relatórios	Listar intimações já efetuadas.
Alterar Cadastro	Alterar os dados cadastrais do usuário. Para isso é necessário entrar no Sistema.
Relembrar Senha	Enviar email com a senha do advogado.
Cadastrar-se	Para se cadastrar e começar a receber intimações via Internet, é necessário dirigir-se à Justiça Federal para assinar o termo de compromisso junto a alguma Vara da Justiça Federal.
Sair	Desconectar-se do Sistema (operação importante após a utilização do serviço).

Figura: Ajuda

10.3. ANEXO III. BIGCHART





#### **10.4. ANEXO IV. PLANO DE ESTÁGIO.**

##### **Ambiente do Estágio**

O projeto será desenvolvido na Justiça Federal Fórum Campina Grande – PB. Na Justiça consta um quadro de 95 computadores, sendo dois servidores, todos os computadores conectados em rede interna, com IP privado. O local (sala) onde foi realizado o projeto localiza-se no primeiro térreo do prédio da Justiça, próximo a área administrativa, constando apenas de 4 computadores, dois servidores e dois computadores. Situada no endereço abaixo.

Justiça Federal da Paraíba,  
Subseção Campina Grande  
Setor de Informática – térreo,  
Rua Edgard Vilarim Meira, s/nº  
Liberdade CEP: 58105-000  
Campina Grande / PB – Brasil  
PABX: (83)310-9100 FAX: (83)310-9104

## Supervisão

### Supervisor Técnico:

Nome: Francisco Hertz A. Farias.

Endereço: Fórum Juiz Federal Nereu Pereira dos Santos Filho

Rua Edgard Vilarim Meira, S/N – Liberdade

CEP 58105-000 – Campina Grande – PB –

Fone: (83) 310-9106

E-mail: [hertz@jfpb.gov.br](mailto:hertz@jfpb.gov.br)

### Supervisor Acadêmico:

Nome: José Antônio Beltrão Moura

Endereço: Universidade Federal de Campina Grande

Centro de Ciências e Tecnologia -

Departamento de Sistemas e Computação.

Rua Aprígio Veloso, s/n Bodocongó.

58109-970 Campina Grande, PB – Brasil.

Telefone: (83) 3101019

Avenida Aprígio Veloso, 882

Caixa Postal 10.106

58.109-970 Campina Grande – Pb

E-mail: [antao@dsc.ufcg.edu.br](mailto:antao@dsc.ufcg.edu.br)

## Resumo do Problema

Com o objetivo de obter a desejada segurança na remessa e no recebimento das informações, entre a Justiça e os advogados, no quesito intimação, e no intuito de otimiza-la, com relação a reduzir as despesas com hora-trabalho/servidor e despesas telefônicas resultantes das intimações por telefone/fax e por carta, que foi requisitado a implementação de um sistema que permitisse encontrar os advogados no tempo hábil em qualquer lugar do país ou do mundo, para que o mesmo tivesse acesso a (as) intimações a ele destinadas.

## Proposta de Solução

O Sistema de Intimações Eletrônicas será um serviço oferecido pela Justiça Federal de Campina Grande aos advogados que desejam agilizar o recebimento de intimações oriundas das Varas Federais.

Através dele o advogado receberá intimações pela Internet, bastando conectar-se ao sistema (link ENTRAR). Para utilizá-lo, o advogado deverá efetuar seu cadastramento junto a qualquer Vara da Justiça Federal que está oferecendo o serviço de Intimação Eletrônica, oportunidade em que assinará um Termo de Compromisso.

As intimações eletrônicas são consideradas como realizadas na data em que o advogado efetua a consulta, iniciando-se a contagem do prazo no primeiro dia útil seguinte. Em não havendo consultas na semana, o sistema efetuará a intimação automaticamente às 6<sup>as</sup> feiras – 18:00 horas, ou no último dia útil da semana, no mesmo horário.

## Atividades a serem Desenvolvidas

### 1.0 Construção do Sistema de Intimações Eletrônicas

1.1. Conhecimento sobre o assunto	(15 horas)
1.2. Levantamento de Requisitos	(25 horas)
1.3. Análise	(30 horas)
1.4 Implementação	(175 horas)
1.5 Testes	(40 horas)
1.6 Documentação	(15 horas)

Total de horas: 300



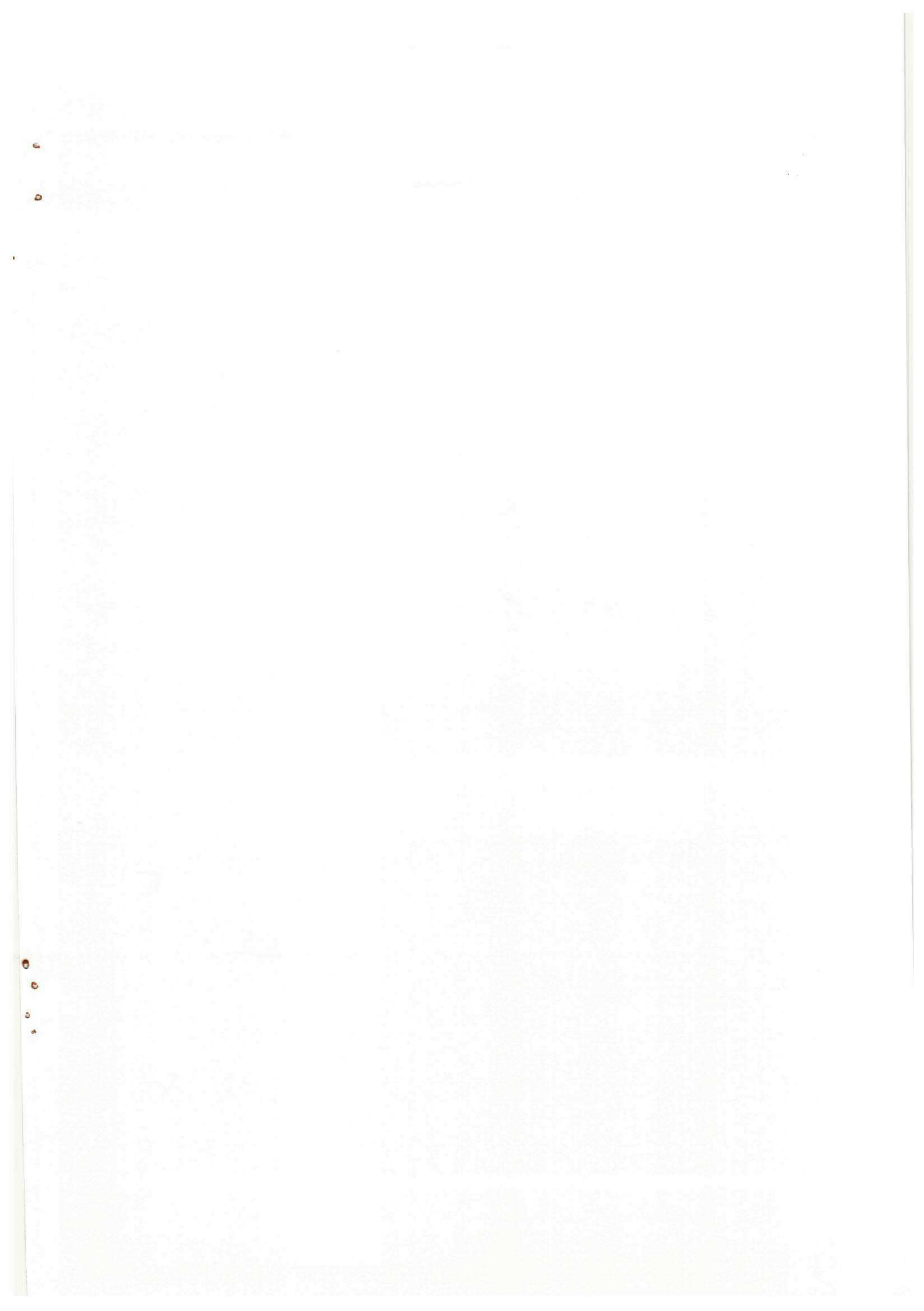
**10.5. ANEXO V. DECLARAÇÃO DE APROVAÇÃO DO ESTÁGIO**

Declaro para os devidos fins que **Adilton Ângelo Seixas Magalhães**, cumpriu corretamente suas atividades para com o **Sistema de Intimação Eletrônica**, o qual foi alvo de seu estágio na Justiça Federal de Campina Grande tendo sido realizado no período de 01 de novembro de 2003 a 30 de abril de 2004. Seus esforços foram valorizados sem nenhuma crítica desta instituição.

Campina Grande, 04 de Maio de 2004.



Francisco Hertz A. Farias  
Supervisor Técnico



# 1

## ATIVIDADES ESSENCIAIS

### Atividades Fundamentais

- Solicitar Serviço
- Distribuir Serviço
- Dispensar Técnico
- Registrar Parecer
- Fechar OS
- Consultar OS\_Pendentes
- Consultar p/ Setor
- Consultar p/ Técnico
- Consultar p/ Computador

### Atividades Custodiais

- Manter COMPUTADOR
- Manter SETOR
- Manter SINTOMA
- Manter TECNICO
- Manter SOFTWARE
- Manter COMP\_SOFT
- Manter ATIVIDADE
- Manter OS
- Manter OS\_SINTOMA
- Desvincular OS\_TEC
- Manter ATIVIDADE\_OS

# 2

## DICIONÁRIO DE DADOS PARA OS DEPÓSITOS DE DADOS

### Dicionário de Dados COMPUTADOR

CAMPO	DESCRIÇÃO	DOMÍNIO	TAMANHO
@COD_COMPUTADOR	Identificação única para cada computador.	String	10
TAM_MEM_RAM	Tamanho da memória RAM.	Integer	----
TAM_HD	Capacidade do HD.	Integer	----
DISKETT	Informa se o computador possui dispositivo de DISKETT. Caso possua, assume valor 'S', senão assume valor 'N'.	Char	1
CD_ROM	Informa se o computador possui dispositivo de CD_ROM. Caso possua, assume valor 'S', senão assume valor 'N'.	Char	1
ATALHOS	Descrição dos atalhos não padrão encontrados na área de trabalho.	String	200
SISTEMA_OPERACIONAL	Sistema operacional que roda no computador.	String	20
#CODIGO_SETOR	Código do setor ao qual o computador pertence.	Integer	----

### Dicionário de Dados SETOR

CAMPO	DESCRIÇÃO	DOMÍNIO	TAMANHO
@COD_SETOR	Identificação única para cada setor.	Integer	----
NOME_SETOR	Nome do setor.	String	30



### Dicionário de Dados OS

CAMPO	DESCRIÇÃO	DOMÍNIO	TAMANHO
@COD_OS	Identificação única para cada ordem de serviço.	Integer	----
#CODIGO_SETOR	Código do setor que solicitou o serviço.	Integer	----
#CODIGO_COMPUTADOR	Código do computador que solicitou o serviço.	String	10
DATA_HORA_REGISTRO	Data e hora do registro da ordem de serviço.	DateTime	----
DATA_HORA_FECHAMENTO	Data e hora do fechamento da ordem de serviço.	DateTime	----

### Dicionário de Dados SINTOMA

CAMPO	DESCRIÇÃO	DOMÍNIO	TAMANHO
@COD_SINTOMA	Identificação única para cada sintoma.	Integer	----
NOME_SINTOMA	Nome do sintoma.	String	100

### Dicionário de Dados OS\_SINTOMA

CAMPO	DESCRIÇÃO	DOMÍNIO	TAMANHO
@CODIGO_OS	Código da ordem de serviço.	Integer	----
@CODIGO_SINTOMA	Código do sintoma.	Integer	----

### Dicionário de Dados TECNICO

CAMPO	DESCRIÇÃO	DOMÍNIO	TAMANHO
@COD_TECNICO	Identificação única para cada técnico.	Integer	----
NOME_TECNICO	Nome do técnico.	String	40



**Dicionário de Dados OS\_TEC**

CAMPO	DESCRIÇÃO	DOMÍNIO	TAMANHO
@CODIGO_OS	Identificação única para cada ordem de serviço.	Integer	----
@CODIGO_TECNICO	Código do técnico que será vinculado à ordem de serviço.	Integer	----
@DATA_HORA_VINCULACAO	Data e hora da vinculação do técnico à ordem de serviço.	DateTime	----
DATA_HORA_DESVINCULACAO	Data e hora da desvinculação do técnico à ordem de serviço.	DateTime	----

**Dicionário de Dados SOFTWARE**

CAMPO	DESCRIÇÃO	DOMÍNIO	TAMANHO
@COD_SOFTWARE	Identificação única para cada software.	Integer	----
NOME_SOFTWARE	Nome do software.	String	40

**Dicionário de Dados COMP\_SOFT**

CAMPO	DESCRIÇÃO	DOMÍNIO	TAMANHO
@CODIGO_COMPUTADOR	Código do computador.	String	10
@CODIGO_SOFTWARE	Código do software.	Integer	----

**Dicionário de Dados ATIVIDADE**

CAMPO	DESCRIÇÃO	DOMÍNIO	TAMANHO
@COD_ATIVIDADE	Identificação única para cada atividade.	Integer	----
NOME_ATIVIDADE	Nome do atividade.	String	40

### Dicionário de Dados ATIVIDADE\_OS

CAMPO	DESCRIÇÃO	DOMÍNIO	TAMANHO
@CODIGO_OS	Código da ordem de serviço.	Integer	----
@CODIGO_TECNICO	Código do técnico.	Integer	----
@CODIGO_ATIVIDADE	Código da atividade a ser registrada.	Integer	----
@DATA_HORA	Data e hora do registro da atividade.	DateTime	----
COMPLEMENTO	Descrição complementar da atividade registrada.	String	100

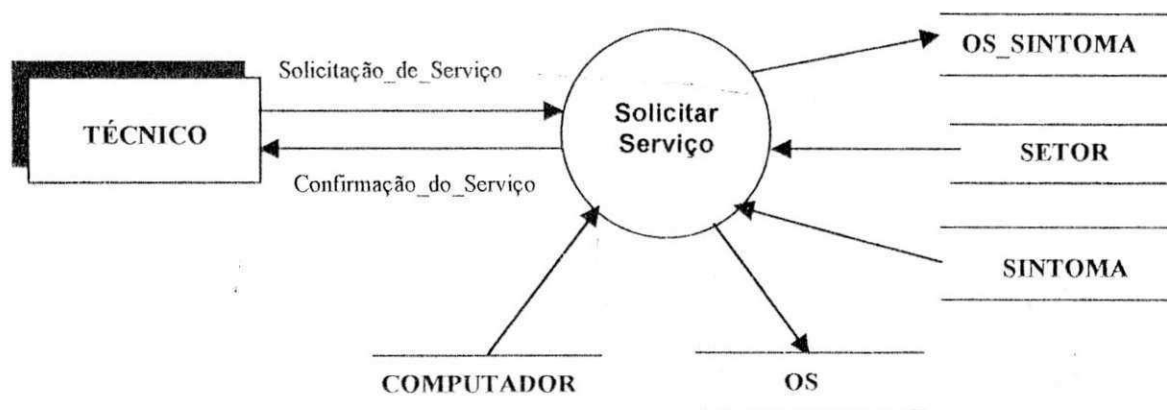


# 3

## DIAGRAMA DE FLUXO DE DADOS E MINIESPECIFICAÇÕES

### 1ª Atividade: Solicitar Serviço

Diagrama de Fluxo de Dados



Dicionário de Dados de Solicitar Serviço	
Solicitação_de_Serviço	CODIGO_COMPUTADOR + CODIGO_SETOR + SINTOMAS_COMPUTADOR
Confirmação_do_Serviço	Mensagem de confirmação do serviço que aparecerá na tela do computador.
CODIGO_COMPUTADOR	Código do computador.
CODIGO_SETOR	Código do setor ao qual o computador pertence.
SINTOMAS_COMPUTADOR	Lista dos sintomas apresentados.

## Miniespecificação de Solicitar Serviço

### INÍCIO PROCEDIMENTO SOLICITAR\_SERVICO

Encontre COMPUTADOR com CODIGO\_COMPUTADOR

Se encontrado

Encontre SETOR com CODIGO\_SETOR

Se encontrado

Para cada CODIGO\_SINTOMA em SINTOMAS\_COMPUTADOR faça:

Encontre SINTOMA com CODIGO\_SINTOMA

Se **não** encontrado

Mensagem "CODIGO\_SINTOMA inválido"

Exclua CODIGO\_SINTOMA de SINTOMAS\_COMPUTADOR

Fim\_Se

Fim\_Para

Se SINTOMAS\_COMPUTADOR não está vazio

Crie uma nova entrada em OS com OS.COD\_OS único

Grave Registro\_OS

Para cada CODIGO\_SINTOMA em SINTOMAS\_COMPUTADOR faça:

Grave Registro OS\_SINTOMA

Fim\_Para

Fim\_Se

Senão

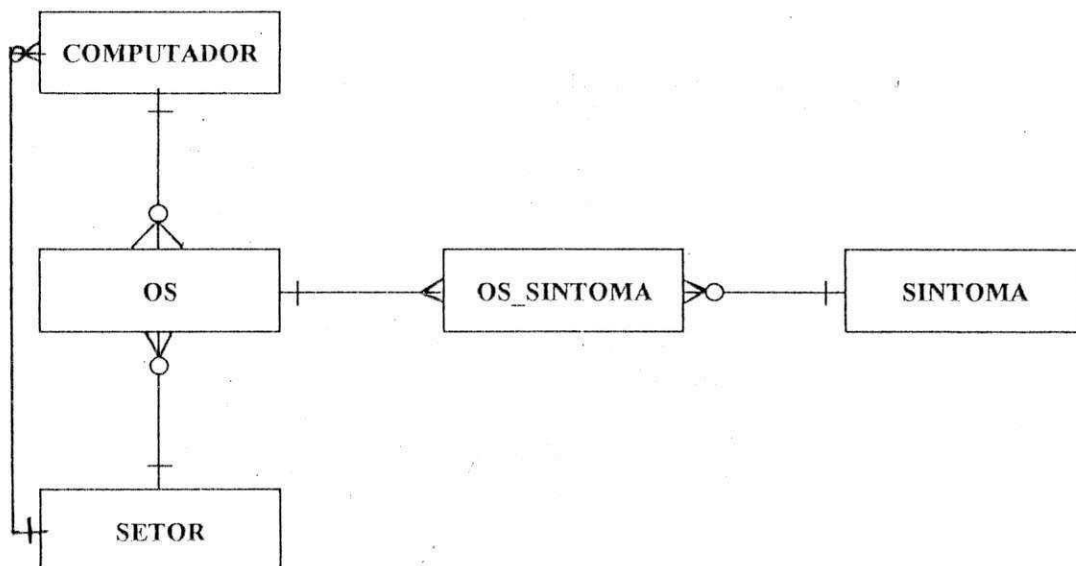
Mensagem "CODIGO\_SETOR inválido"

Senão

Mensagem "CODIGO\_COMPUTADOR inválido"

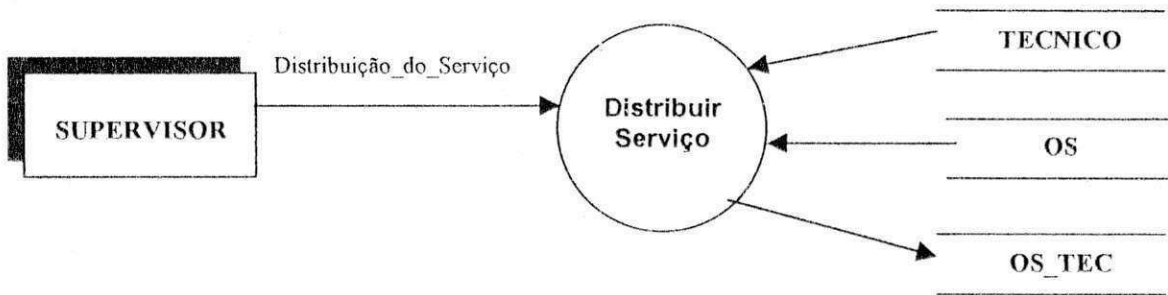
FIM PROCEDIMENTO SOLICITAR\_SERVICO

### DER



## 2ª Atividade: Distribuir Serviço

### Diagrama de Fluxo de Dados



Dicionário de Dados de Distribuir Serviço	
Distribuição_do_Serviço	CODIGO_OS + CODIGO_TECNICO
CODIGO_OS	Código da ordem de serviço.
CODIGO_TECNICO	Código do Técnico.



### Miniespecificação de Distribuir Serviço

#### INÍCIO PROCEDIMENTO DISTRIBUIR\_SERVICO

Encontre OS com CODIGO\_OS

Se encontrado

Encontre TECNICO com CODIGO\_TECNICO

Se encontrado

Encontre OS\_TEC com CODIGO\_OS e DATA\_HORA\_DESVINCULACAO =  
"NULL"

Se encontrado

Mensagem "OS já encontra-se vinculada a um técnico"

Senão

Encontre OS\_TEC com CODIGO\_TECNICO e  
DATA\_HORA\_DESVINCULACAO = "NULL"

Se encontrado

Mensagem "Técnico não disponível"

Senão

Grave Registro\_OS\_TEC

Senão

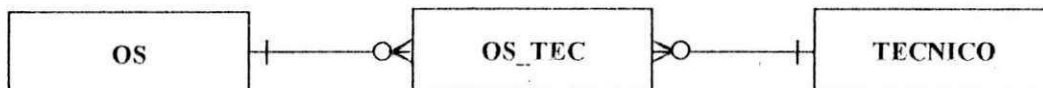
Mensagem "CODIGO\_TECNICO inválido"

Senão

Mensagem "CODIGO\_OS inválido"

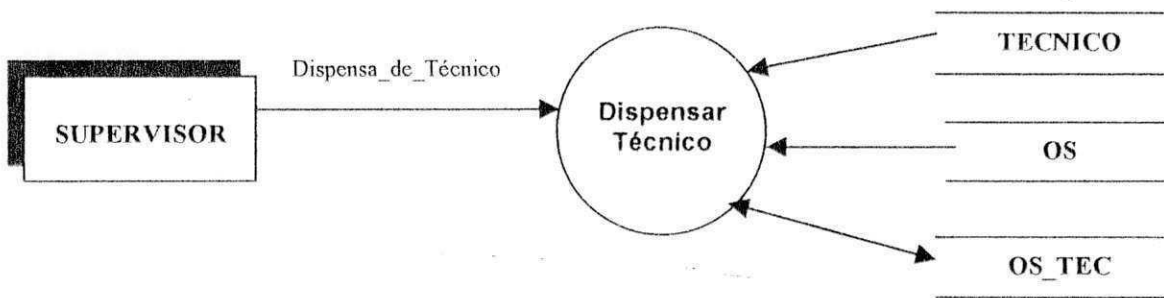
FIM PROCEDIMENTO DISTRIBUIR\_SERVICO

### DER



### 3ª Atividade: Dispensar Técnico

#### Diagrama de Fluxo de Dados



Dicionário de Dados de Dispensar Técnico	
Dispensa_de_Técnico	CODIGO_OS + CODIGO_TECNICO
CODIGO_OS	Código da ordem de serviço.
CODIGO_TECNICO	Código do Técnico.

## Miniespecificação de Dispensar Técnico

### **INÍCIO PROCEDIMENTO DISPENSAR\_TECNICO**

Encontre OS com CODIGO\_OS

Se encontrado

Encontre OS\_TEC com CODIGO\_OS

Se encontrado

Encontre TECNICO com CODIGO\_TECNICO

Se encontrado

Encontre OS\_TEC com CODIGO\_OS e CODIGO\_TECNICO

Se encontrado

Encontre OS\_TEC com CODIGO\_OS, CODIGO\_TECNICO e  
DATA\_HORA\_DESVINCULACAO = "NULL"

Se encontrado

Grave a data e hora atuais em DATA\_HORA\_DESVINCULACAO em  
OS\_TEC

Senão

Mensagem "Técnico já dispensado"

Senão

Mensagem "Técnico não vinculado à Ordem de Serviço"

Senão

Mensagem "CODIGO\_TECNICO inválido"

Senão

Mensagem "Ordem de Serviço ainda não vinculada a um Técnico"

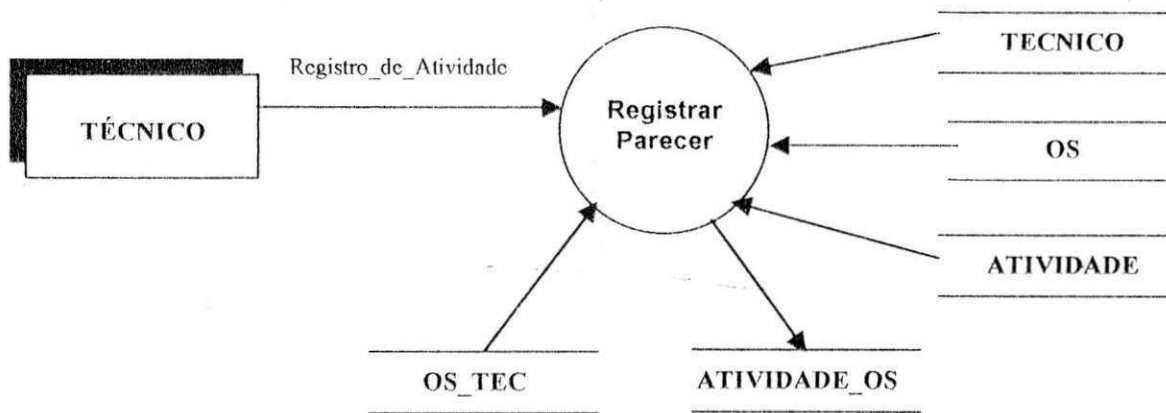
Senão

Mensagem "CODIGO\_OS inválido"

**FIM PROCEDIMENTO DISPENSAR\_TECNICO**

#### 4ª Atividade: Registrar Atividade

Diagrama de Fluxo de Dados



Dicionário de Dados de Registrar Atividade	
Registro_de_Atividade	CODIGO_OS + CODIGO_TECNICO + CODIGO_ATIVIDADE + COMPLEMENTO
CODIGO_OS	Código da ordem de serviço.
CODIGO_TECNICO	Código do Técnico.
CODIGO_ATIVIDADE	Código da Atividade.
COMPLEMENTO	Descrição complementar da atividade.

## Miniespecificação de Registrar Atividade

### INICIO PROCEDIMENTO REGISTRAR\_ATIVIDADE

Encontre OS com CODIGO\_OS

Se encontrado

Encontre OS\_TEC com CODIGO\_OS

Se encontrado

Encontre TECNICO com CODIGO\_TECNICO

Se encontrado

Encontre OS\_TEC com CODIGO\_OS e CODIGO\_TECNICO

Se encontrado

Encontre ATIVIDADE com CODIGO\_ATIVIDADE

Se encontrado

Grave Registro\_ATIVIDADE\_OS

Senão

Mensagem "CODIGO\_ATIVIDADE inválido"

Senão

Mensagem "Técnico não vinculado à Ordem de Serviço"

Senão

Mensagem "CODIGO\_TECNICO inválido"

Senão

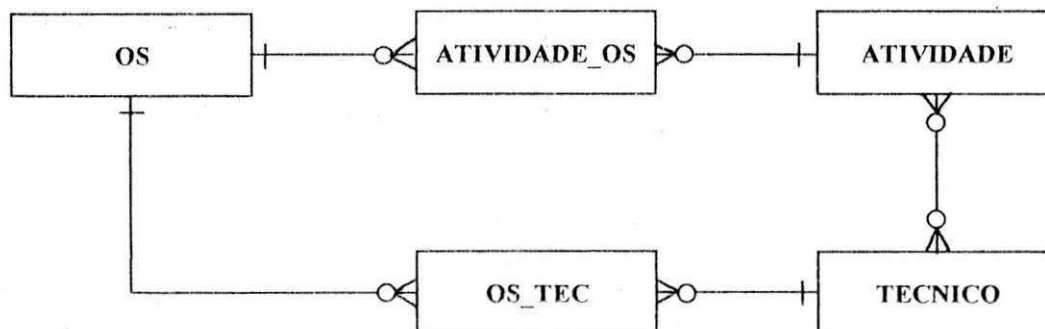
Mensagem "Ordem de Serviço ainda não vinculada a um Técnico"

Senão

Mensagem "CODIGO\_OS inválido"

FIM PROCEDIMENTO REGISTRAR\_ATIVIDADE

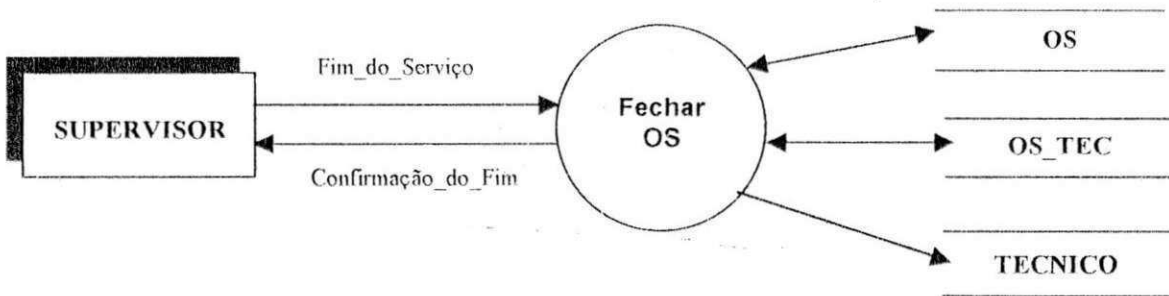
### DER





## 5ª Atividade: Fechar OS

### Diagrama de Fluxo de Dados



Dicionário de Dados de Fechar OS	
Fim_do_Serviço	CODIGO_OS
Confirmação_do_Fim	Mensagem de confirmação que aparecerá na tela.
CODIGO_OS	Código da ordem de serviço.

## Miniespecificação de Fechar OS

### **INICIO PROCEDIMENTO FECHAR\_OS**

Encontre OS com CODIGO\_OS

Se encontrado

Encontre OS\_TEC com CODIGO\_OS

Se encontrado

Encontre OS com CODIGO\_OS e DATA\_HORA\_FECHAMENTO = "NULL"

Se encontrado

Encontre OS\_TEC com CODIGO\_OS e DATA\_DESVINCULACAO = "NULL"

Se encontrado

Grave a data e a hora atuais em DATA\_HORA\_DESVINCULACAO em  
OS\_TEC

Senão

Mensagem "Técnico já dispensado"

Grave a hora e a data atuais em DATA\_HORA\_FECHAMENTO em OS

Senão

Mensagem "Ordem de Serviço encontra-se fechada"

Senão

Mensagem "Ordem de Serviço ainda não foi vinculada a um Técnico"

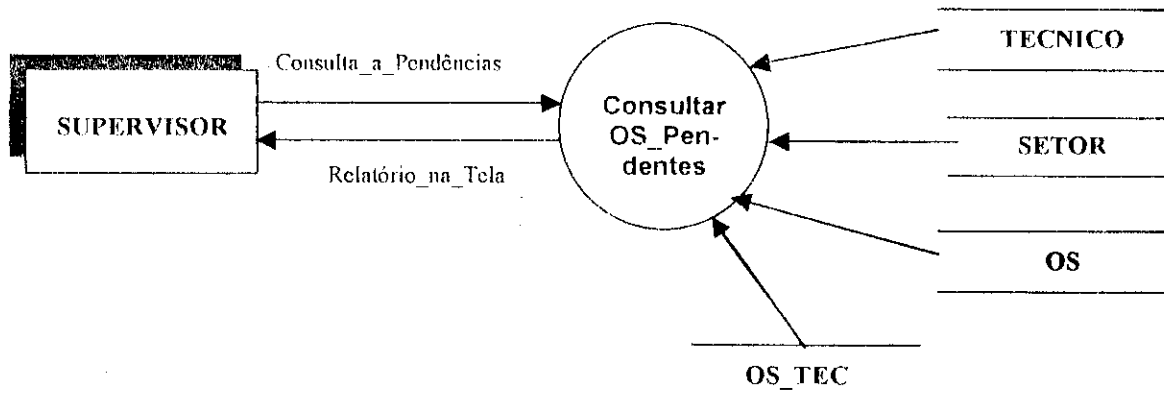
Senão

Mensagem "CODIGO\_OS inválido"

**FIM PROCEDIMENTO FECHAR\_OS**

## 6ª Atividade: Consultar OS Pendentes

### Diagrama de Fluxo de Dados



### Dicionário de Dados de Consultar OS Pendentes

Consulta_a_Pendências	O estagiário requiere ao sistema uma lista com as ordens de serviço pendentes.
Relatório_na_Tela	Lista_OS_Pendentes*
lista_OS_Pendentes	COD_OS + CODIGO_COMPUTADOR + NOME_SETOR + DATA_HORA_REGISTRO + lista_técnicos*
lista_técnicos	NOME_TECNICO + DATA_HORA_VINCULACAO + DATA_HORA_DESVINCULACAO

### Miniespecificação de Consultar OS Pendentes

#### INÍCIO PROCEDIMENTO CONSULTAR\_OS\_PENDENTES

Para cada OS com DATA\_HORA\_FECHAMENTO = "NULL" faça:

    Mostre COD\_OS, CODIGO\_COMPUTADOR, NOME\_SETOR associado com OS.CODIGO\_SETOR em SETOR e DATA\_HORA\_REGISTRO

Para cada OS\_TEC com OS.COD\_OS faça:

    Mostre NOME\_TECNICO associado com OS\_TEC.CODIGO\_TECNICO em TECNICO, DATA\_HORA\_VINCULACAO e DATA\_HORA\_DESVINCULACAO

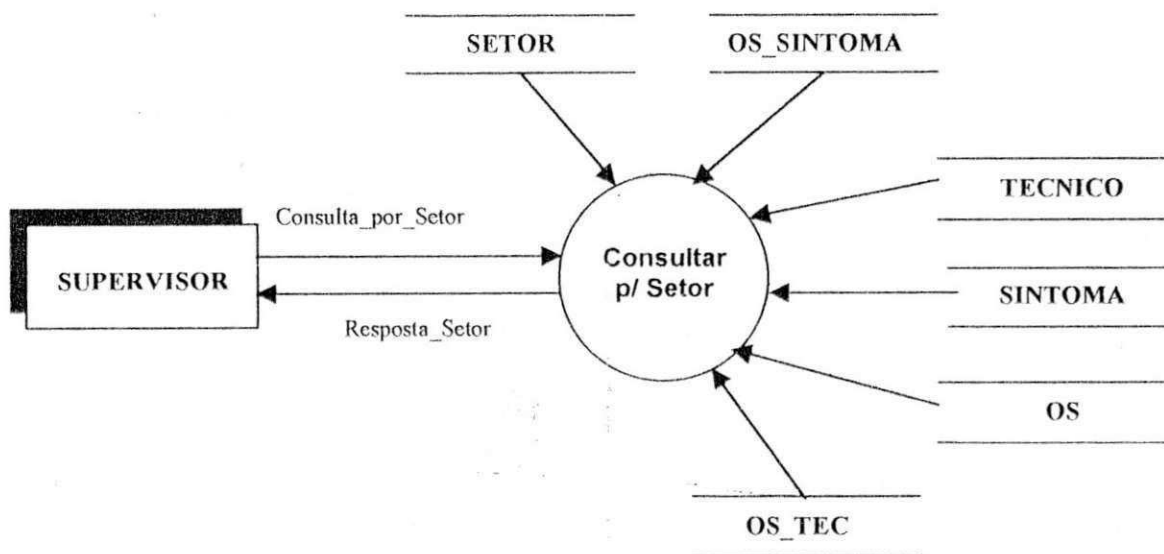
    Fim\_Para

Fim\_Para

#### FIM PROCEDIMENTO CONSULTAR\_OS\_PENDENTES

## 7ª Atividade: Consultar p/ Setor

### Diagrama de Fluxo de Dados



Dicionário de Dados de Consultar p/ Setor	
Consulta_por_Setor	CODIGO_SETOR + ESTADO + DATA1 + DATA2
Relatório_Setor	NOME_SETOR + DATA1 + DATA2 + lista_computadores*
lista_computadores	CODIGO_COMPUTADOR + lista_OS *
lista_OS	COD_OS + DATA_HORA_REGISTRO + DATA_HORA_FECHAMENTO + lista_sintomas * + lista_OS_TEC *
lista_sintomas	NOME_SINTOMA
Lista_OS_TEC	NOME_TECNICO + DATA_HORA_VINCULACAO + DATA_HORA_DESVINCULACAO + lista_atividades *
CODIGO_SETOR	Código do Setor
ESTADO	Indica quais OSs serão selecionadas. Possíveis valores assumidos: 'A' -> OSs em aberto 'F' -> OSs fechadas 'T' -> Todas
DATA1	Limite inferior para a data de registro das Ordens de Serviço na consulta.
DATA2	Limite superior para a data de registro das Ordens de Serviço na consulta.

## Miniespecificação de Consultar p/ Setor

### PROCEDIMENTO CONSULTAR\_POR\_SETOR

#### INÍCIO

Encontre SETOR com CODIGO\_SETOR

Se encontrado

Mostre NOME\_SETOR

Se ESTADO = 'A'

Para cada OS com CODIGO\_SETOR, DATA\_HORA\_FECHAMENTO = "NULL"  
e DATA\_HORA\_REGISTRO entre DATA1 e DATA2 faça:

Salve cada OS.COD\_COMPUTADOR distinto em COMP\_FILE

Fim\_Para

Para cada COD\_COMPUTADOR em COMP\_FILE faça:

Mostre COD\_COMPUTADOR

Para cada OS com CODIGO\_SETOR, DATA\_HORA\_FECHAMENTO =  
"NULL" e DATA\_HORA\_REGISTRO entre DATA1 e DATA2 faça:

Mostre COD\_OS, DATA\_HORA\_REGISTRO e  
DATA\_HORA\_FECHAMENTO

Chamar LISTAR\_SINTOMA\_TECNICO(COD\_OS)

Fim\_Para

Fim\_Para

Senão

Se ESTADO = 'F'

Para cada OS com CODIGO\_SETOR, DATA\_HORA\_FECHAMENTO não =  
"NULL" e DATA\_HORA\_REGISTRO entre DATA1 e DATA2 faça:

Salve cada OS.COD\_COMPUTADOR distinto em COMP\_FILE

Fim\_Para

Para cada COD\_COMPUTADOR em COMP\_FILE faça:

Mostre COD\_COMPUTADOR

Para cada OS com CODIGO\_SETOR, DATA\_HORA\_FECHAMENTO não  
"NULL" e DATA\_HORA\_REGISTRO entre DATA1 e DATA2 faça:

Mostre COD\_OS, DATA\_HORA\_REGISTRO e  
DATA\_HORA\_FECHAMENTO

Chamar LISTAR\_SINTOMA\_TECNICO(COD\_OS)

Fim\_Para

Fim\_Para

Senão

Se ESTADO = 'T'

Para cada OS com CODIGO\_SETOR e DATA\_HORA\_REGISTRO entre  
DATA1 e DATA2 faça:

Salve cada OS.COD\_COMPUTADOR distinto em COMP\_FILE

Fim\_Para

Para cada COD\_COMPUTADOR em COMP\_FILE faça:

Mostre COD\_COMPUTADOR

Para cada OS com CODIGO\_SETOR e DATA\_HORA\_REGISTRO  
entre DATA1 e DATA2 faça:

Mostre COD\_OS, DATA\_HORA\_REGISTRO e



DATA\_HORA\_FECHAMENTO

Chamar LISTAR\_SINTOMA\_TECNICO(COD\_OS)

Fim\_Para

Fim\_Para

Fim\_Se

Senão

Mensagem "CODIGO\_SETOR inválido"

**FIM PROCEDIMENTO CONSULTAR\_POR\_SETOR**

**PROCEDIMENTO LISTAR\_SINTOMA\_TECNICO(ID\_OS)**  
**INÍCIO**

Para cada OS\_SINTOMA com ID\_OS faça:

Mostre NOME\_SINTOMA associado com OS\_SINTOMA.CODIGO\_SINTOMA em  
SINTOMA

Fim\_Para

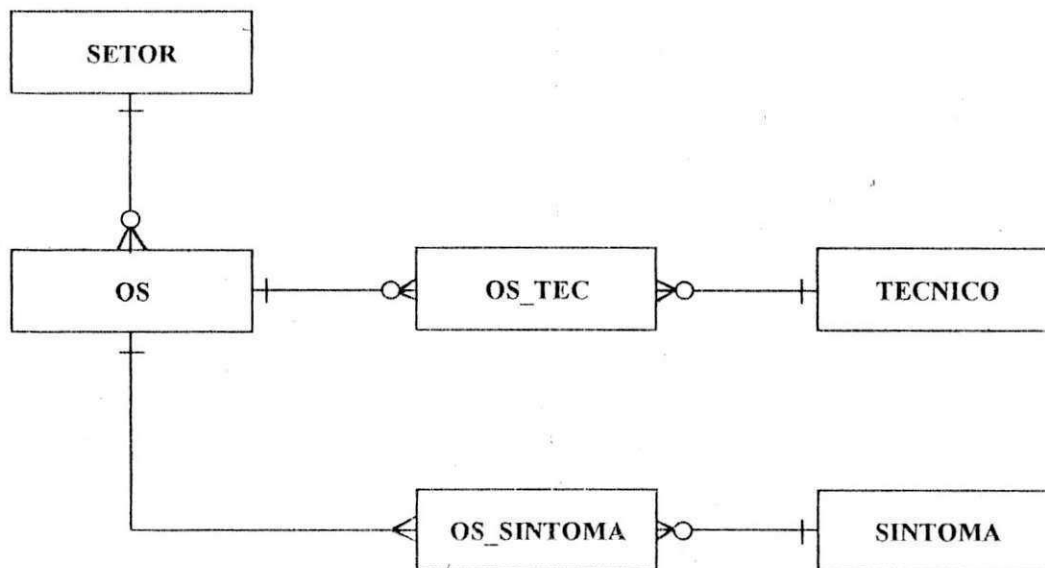
Para cada OS\_TEC com ID\_OS faça:

Mostre NOME\_TECNICO associado com OS\_TEC.CODIGO\_TECNICO em  
TECNICO, DATA\_HORA\_VINCULACAO e DATA\_HORA\_DESVINCULACAO

Fim\_Para

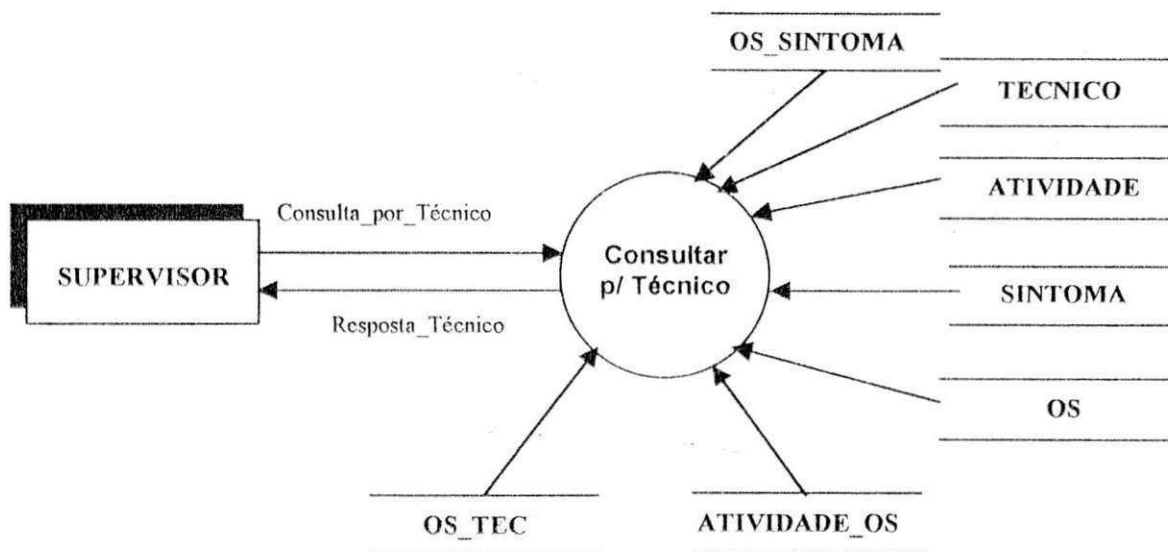
**FIM PROCEDIMENTO LISTAR\_SINTOMA\_TECNICO**

DER



## 8ª Atividade: Consultar p/ Técnico

### Diagrama de Fluxo de Dados



#### Dicionário de Dados de Consultar p/ Técnico

Consulta_por_Técnico	CODIGO_TECNICO + ESTADO + DATA1 + DATA2
Relatório_Técnico	NOME_TECNICO + DATA1 + DATA2 + lista_serviços *
lista_serviços	COD_OS + DATA_HORA_REGISTRO + DATA_HORA_FECHAMENTO+ DATA_HORA_VINCULACAO + DATA_HORA_DESVINCULACAO + CODIGO_COMPUTADOR + lista_sintomas * + lista_atividades *
lista_sintomas	NOME_SINTOMA
lista_atividades	NOME_ATIVIDADE + DATA_HORA + COMPLEMENTO
ESTADO	Indica quais OSs serão selecionadas. Possíveis valores assumidos: 'A' -> OSs em aberto 'F' -> OSs fechadas 'T' -> Todas
DATA1	Limite inferior para a data de registro das Ordens de Serviço na consulta.
DATA2	Limite superior para a data de registro das Ordens de Serviço na consulta.

### Miniespecificação de Consultar p/ Técnico

#### PROCEDIMENTO CONSULTAR\_POR\_TECNICO

##### INÍCIO

Encontre TECNICO com CODIGO\_TECNICO

Se encontrado

Mostre NOME\_TECNICO

Se ESTADO = 'A'

Para cada OS\_TEC com CODIGO\_TECNICO, DATA\_HORA\_FECHAMENTO =  
"NULL" e DATA\_HORA\_REGISTRO entre DATA1 e DATA2 faça:

Chame LISTAR\_RESULTADO(OS\_TEC.CODIGO\_OS, CODIGO\_TECNICO)

Fim\_Para

Senão

Se ESTADO = 'F'

Para cada OS\_TEC com CODIGO\_TECNICO, DATA\_HORA\_FECHAMENTO  
não = "NULL" e DATA\_HORA\_REGISTRO entre DATA1 e DATA2 faça:

Salve cada OS\_TEC.CODIGO\_OS distinto em OS\_FILE

Fim\_Para

Para cada CODIGO\_OS em OS\_FILE faça:

Chame LISTAR\_RESULTADO(CODIGO\_OS, CODIGO\_TECNICO)

Fim\_Para

Senão

Se ESTADO = 'T'

Para cada OS\_TEC com CODIGO\_TECNICO e  
DATA\_HORA\_REGISTRO entre DATA1 e DATA2 faça:

Salve cada OS\_TEC.CODIGO\_OS distinto em OS\_FILE

Fim\_Para

Para cada CODIGO\_OS em OS\_FILE faça:

Chame LISTAR\_RESULTADO(CODIGO\_OS, CODIGO\_TECNICO)

Fim\_Para

Fim\_Se

Senão

Mensagem "CODIGO\_TECNICO inválido"

**FIM PROCEDIMENTO CONSULTAR\_POR\_TECNICO**

#### PROCEDIMENTO LISTAR\_RESULTADO(ID\_OS, ID\_TECNICO)

##### INÍCIO

Encontre OS com IS\_OS

Mostre OS.COD\_OS, OS.DATA\_HORA\_REGISTRO, OS.DATA\_HORA\_FECHAMENTO  
e OS.COD\_COMPUTADOR

Para cada OS\_SINTOMA com ID\_OS faça:

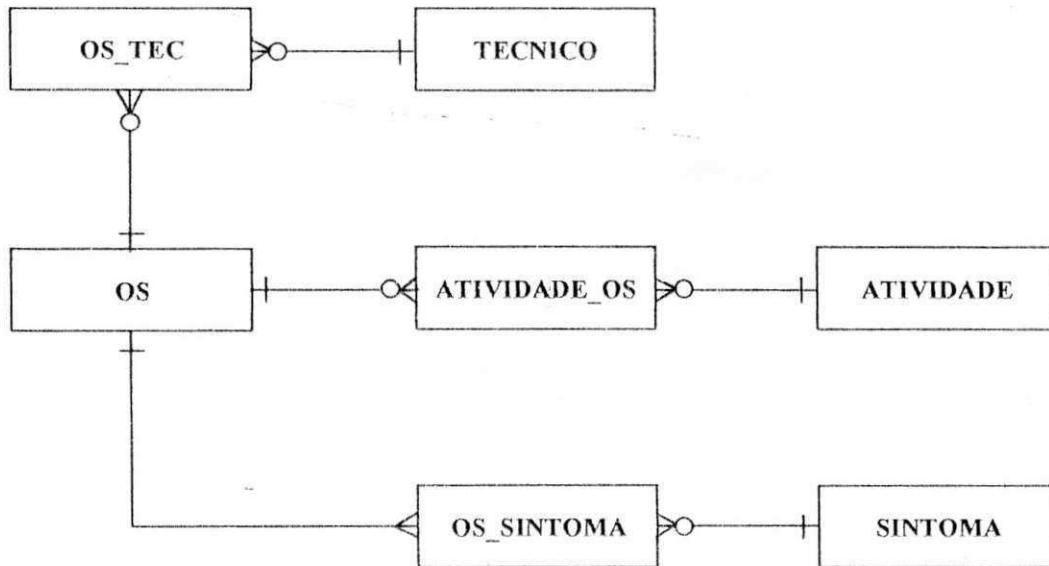
Mostre NOME\_SINTOMA associado com OS\_SINTOMA.CODIGO\_SINTOMA em  
SINTOMA

Fim\_Para

Para cada OS\_TEC com ID\_OS e OD\_TECNICO faça:

Mostre DATA\_HORA\_VINCULACAO e DATA\_HORA\_DESVINCULACAO  
Fim\_Para  
Para cada ATIVIDADE\_OS com COD\_OS faça:  
Mostre NOME\_ATIVIDADE associado com ATIVIDADE\_OS.CODIGO\_ATIVIDADE  
em ATIVIDADE, DATA\_HORA e COMPLEMENTO  
Fim\_Para  
**FIM PROCEDIMENTO LISTAR\_RESULTADO**

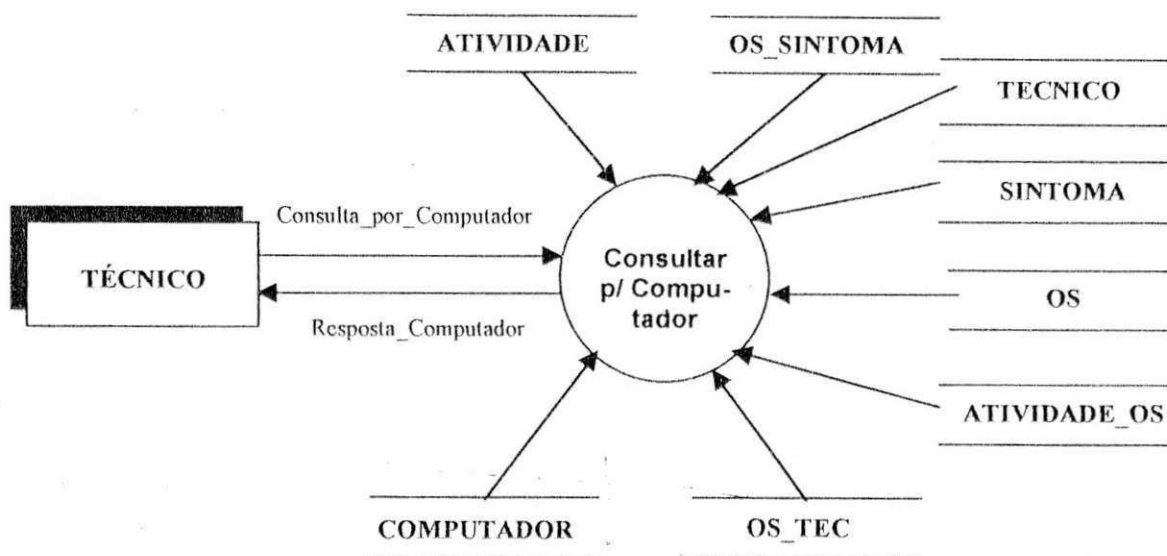
DER





## 9ª Atividade: Consultar p/ Computador

### Diagrama de Fluxo de Dados



Dicionário de Dados de Consultar p/ Computador	
Consulta_por_Computador	CODIGO_COMPUTADOR + ESTADO + DATA1 + DATA2
Relatório_Computador	CODIGO_COMPUTADOR + DATA1 + DATA2 + lista_OS *
lista_OS	COD_OS + DATA_HORA_REGISTRO + DATA_HORA_FECHAMENTO + lista_sintomas * + lista_OS_TEC *
lista_sintomas	NOME_SINTOMA
lista_OS_TEC	NOME_TECNICO + DATA_HORA_VINCULACAO + DATA_HORA_DESVINCULACAO + lista_atividades *
lista_atividades	NOME_ATIVIDADE + DATA_HORA + COMPLEMENTO
ESTADO	Indica quais OSs serão selecionadas. Possíveis valores assumidos: 'A' -> OSs em aberto 'F' -> OSs fechadas 'T' -> Todas
DATA1	Limite inferior para a data de registro das Ordens de Serviço na consulta.
DATA2	Limite superior para a data de registro das Ordens de Serviço na consulta.

## Miniespecificação de Consultar p/ Computador

### PROCEDIMENTO CONSULTAR\_POR\_COMPUTADOR

#### INÍCIO

Encontre COMPUTADOR com CODIGO\_COMPUTADOR

Se encontrado

Retorne CODIGO\_COMPUTADOR

Se ESTADO = 'A'

Para cada OS com CODIGO\_COMPUTADOR e DATA\_HORA\_FECHAMENTO =  
"NULL" e DATA\_HORA\_REGISTRO entre DATA1 e DATA2 faça:

Chame LISTAR\_DADOS(OS.COD\_OS)

Fim\_Para

Senão

Se ESTADO = 'F'

Para cada OS com CODIGO\_COMPUTADOR,  
DATA\_HORA\_FECHAMENTO não = "NULL" e  
DATA\_HORA\_REGISTRO entre DATA1 e DATA2 faça:

Chame LISTAR\_DADOS(OS.COD\_OS)

Fim\_Para

Senão

Se ESTADO = 'T'

Para cada OS com CODIGO\_COMPUTADOR e  
DATA\_HORA\_REGISTRO entre DATA1 e DATA2 faça:

Chame LISTAR\_DADOS(OS.COD\_OS)

Fim\_Para

Fim\_Se

Senão

Mensagem "COD\_COMPUTADOR inválido"

### FIM PROCEDIMENTO CONSULTAR\_POR\_COMPUTADOR

### PROCEDIMENTO LISTAR\_DADOS(ID\_OS)

#### INÍCIO

Encontre OS com ID\_OS

Mostre OS.COD\_OS, OS.DATA\_HORA\_REGISTRO, OS.DATA\_HORA\_FECHAMENTO

Para cada OS\_SINTOMA com ID\_OS faça:

Mostre NOME\_SINTOMA associado com OS\_SINTOMA.CODIGO\_SINTOMA em SINTOMA

Fim\_Para

Para cada OS\_TEC com ID\_OS faça:

Mostre NOME\_TECNICO associado com OS\_TEC.CODIGO\_TECNICO em  
TECNICO, DATA\_HORA\_VINCULACAO e DATA\_HORA\_DESVINCULACAO

Para cada ATIVIDADE\_OS com ID\_OS e OS\_TEC.CODIGO\_TECNICO faça:

Mostre NOME\_ATIVIDADE associado com  
ATIVIDADE\_OS.CODIGO\_ATIVIDADE em ATIVIDADE,  
DATA\_HORA e COMPLEMENTO

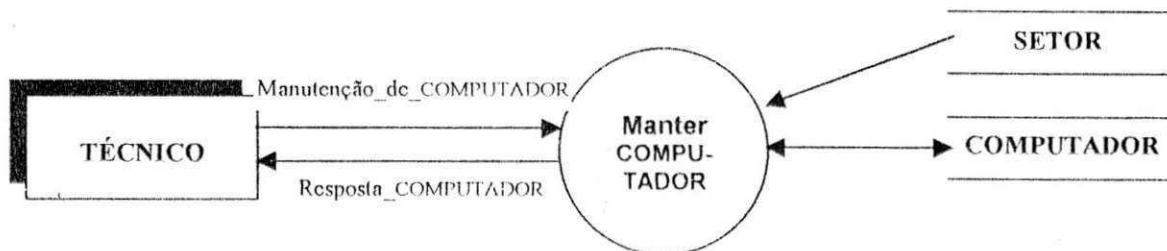
Fim\_Para

Fim\_Para

### FIM PROCEDIMENTO LISTAR\_RESULTADO

## 10ª Atividade: Manter COMPUTADOR

### Diagrama de Fluxo de Dados



Dicionário de Dados de Manter COMPUTADOR	
Manutenção_de_COMPUTADOR	OP_COMPUTADOR + PARAMETROS_COMPUTADOR
Resposta_COMPUTADOR	Se OP_COMPUTADOR = 1, 2 ou 3, a resposta será uma mensagem de confirmação na tela. Se OP_COMPUTADOR = 4, a resposta será o retorno de um registro.
OP_COMPUTADOR	Indica o tipo da operação de manutenção. Possíveis valores assumidos: 1 -> Cadastrar 2 -> Atualizar 3 -> Excluir 4 -> Consultar
PARAMETROS_COMPUTADOR	CODIGO_COMPUTADOR + MEMORIA_RAM + TAMANHO_HD + DISP_DISKETT + DISP_CDROM + SHORTCUTS + SO + CODIGO_SETOR
MEMORIA_RAM	Tamanho da memória RAM do computador.
TAMANHO_HD	Capacidade do HD.
DISP_DISKETT	Informa se o computador possui dispositivo de DISKETT. Possíveis valores assumidos: 'S' -> Sim 'N' -> Não
DISP_CDROM	Informa se o computador possui dispositivo de CD_ROM. Possíveis valores assumidos: 'S' -> Sim 'N' -> Não
SHORTCUTS	Descrição de atalhos não-padrão encontrados na área de trabalho.
SO	Sistema operacional que roda no computador.
CODIGO_SETOR	Código do setor ao qual o computador pertence.



## Miniespecificação de Manter COMPUTADOR

### PROCEDIMENTO MANTER\_COMPUTADOR

#### INÍCIO

Se OP\_COMPUTADOR = 1

Encontre COMPUTADOR com CODIGO\_COMPUTADOR

Se encontrado

Mensagem "Erro: Computador já cadastrado"

Senão

Encontre SETOR com CODIGO\_SETOR

Se encontrado

Crie uma nova entrada em COMPUTADOR com COD.COMPUTADOR único

Grave Registro\_COMPUTADOR a partir de PARAMETROS\_COMPUTADOR

Senão

Mensagem "CODIGO\_SETOR inválido"

Senão

Se OP\_COMPUTADOR = 2

Encontre COMPUTADOR com CODIGO\_COMPUTADOR

Se encontrado

Valide PARAMETROS\_COMPUTADOR.

Se PARAMETROS\_COMPUTADOR válidos

Atualize Registro\_COMPUTADOR

Senão

Mensagem "CODIGO\_COMPUTADOR inválido"

Senão

Se OP\_COMPUTADOR = 3

Encontre COMPUTADOR com CODIGO\_COMPUTADOR

Se encontrado

Encontre OS com CODIGO\_COMPUTADOR faça:

Se encontrado

Mensagem "ERRO: Existem Ordens de Serviço vinculadas ao Computador"

Senão

Para cada COMP\_SOFT com CODIGO\_COMPUTADOR faça:

Exclua Registro\_COMP\_SOFT

Fim\_Para

Exclua Registro\_COMPUTADOR

Senão

Mensagem "CODIGO\_COMPUTADOR inválido"

Senão

Se OP\_COMPUTADOR = 4

Encontre COMPUTADOR com CODIGO\_COMPUTADOR

Se encontrado

Retorne Registro\_COMPUTADOR

Senão

Mensagem "CODIGO\_COMPUTADOR inválido"

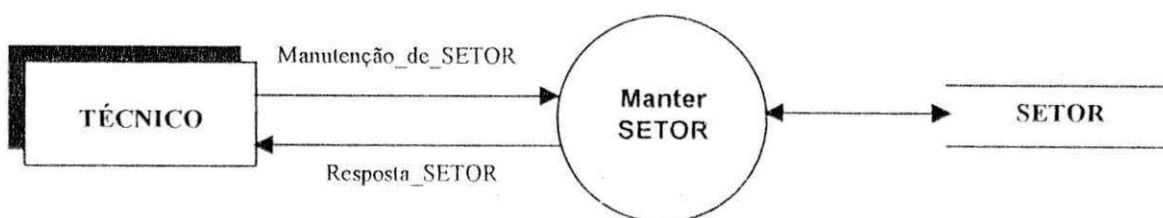
Fim\_se

#### FIM PROCEDIMENTO MANTER\_COMPUTADOR



## 11ª Atividade: Manter SETOR

### Diagrama de Fluxo de Dados



Dicionário de Dados de Manter SETOR	
Manutenção_de_SETOR	OP_SETOR + PARAMETROS_SETOR
Resposta_SETOR	Se OP_SETOR = 1, 2 ou 3, a resposta será uma mensagem de confirmação na tela. Se OP_SETOR = 4, a resposta será o retorno de um registro.
OP_SETOR	Indica o tipo da operação de manutenção. Possíveis valores assumidos: 1 -> Cadastrar 2 -> Atualizar 3 -> Excluir 4 -> Consultar
PARAMETROS_SETOR	CODIGO_SETOR + NOME_DO_SETOR
CODIGO_SETOR	Código do setor.
NOME_DO_SETOR	Nome do setor.

## Miniespecificação de Manter SETOR

### PROCEDIMENTO MANTER\_SINTOMA

#### INÍCIO

Se OP\_SETOR = 1

Crie uma nova entrada em SETOR com COD\_SETOR único  
Grave Registro\_SETOR a partir de PARAMETROS\_SETOR

Senão

Se OP\_SETOR = 2

Encontre SETOR com CODIGO\_SETOR

Se encontrado

Se NOME\_DO\_SETOR em PARAMETROS\_SETOR não = "NULL"  
Atualize Registro\_SETOR com NOME\_DO\_SETOR

Fim\_Se

Senão

Mensagem "CODIGO\_SETOR inválido"

Senão

Se OP\_SETOR = 3

Encontre SETOR com CODIGO\_SETOR

Se encontrado

Encontre OS com CODIGO\_SETOR

Se encontrado

Mensagem "Erro: Existem Ordens de Serviço vinculadas ao Setor"

Senão

Exclua Registro\_SETOR

Senão

Mensagem "CODIGO\_SETOR inválido"

Senão

Se OP\_SETOR = 4

Encontre SETOR com CODIGO\_SETOR

Se encontrado

Retorne Registro\_SETOR

Senão

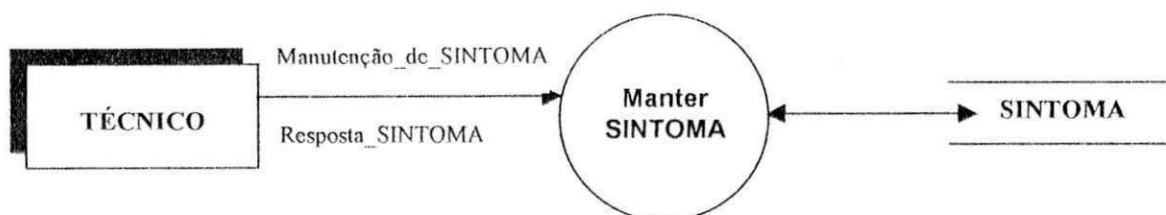
Mensagem "CODIGO\_SETOR inválido"

Fim\_Se

**FIM PROCEDIMENTO MANTER\_SETOR**

## 12ª Atividade: Manter SINTOMA

### Diagrama de Fluxo de Dados



Dicionário de Dados de Manter SINTOMA	
Manutenção_de_SINTOMA	OP_SINTOMA + PARAMETROS_SINTOMA
Resposta_SINTOMA	Se OP_SINTOMA = 1, 2 ou 3, a resposta será uma mensagem de confirmação na tela. Se OP_SINTOMA = 4, a resposta será o retorno de um registro.
OP_SINTOMA	Indica o tipo da operação de manutenção. Possíveis valores assumidos: 1 -> Cadastrar 2 -> Atualizar 3 -> Excluir 4 -> Consultar
PARAMETROS_SINTOMA	CODIGO_SINTOMA + NOME_DO_SINTOMA
CODIGO_SINTOMA	Código do sintoma.
NOME_DO_SINTOMA	Nome do sintoma.

## Miniespecificação de Manter SINTOMA

### PROCEDIMENTO MANTER\_SINTOMA

#### INÍCIO

Se OP\_SINTOMA = 1

Crie uma nova entrada em SINTOMA com COD\_SINTOMA único  
Grave Registro\_SINTOMA a partir de PARAMETROS\_SINTOMA

Senão

Se OP\_SINTOMA = 2

Encontre SINTOMA com CODIGO\_SINTOMA

Se encontrado

Se NOME\_DO\_SINTOMA em PARAMETROS\_SINTOMA não = "NULL"

Atualize Registro\_SINTOMA com NOME\_D0\_SINTOMA

Fim\_Se

Senão

Mensagem "CODIGO\_SINTOMA inválido"

Senão

Se OP\_SINTOMA = 3

Encontre SINTOMA com CODIGO\_SINTOMA

Se encontrado

Encontre OS\_SINTOMA com CODIGO\_SINTOMA

Se encontrado

Mensagem "Erro: Existem Ordens de Serviço vinculadas ao  
Sintoma"

Senão

Exclua Registro\_SINTOMA

Senão

Mensagem "CODIGO\_SINTOMA inválido"

Senão

Se OP\_SINTOMA = 4

Encontre SINTOMA com CODIGO\_SINTOMA

Se encontrado

Retorne Registro\_SINTOMA

Senão

Mensagem "CODIGO\_SINTOMA inválido"

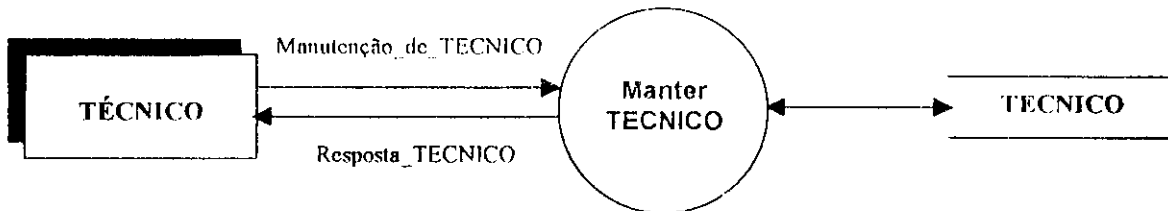
Fim\_Se

#### FIM PROCEDIMENTO MANTER\_SINTOMA



## 13ª Atividade: Manter TECNICO

### Diagrama de Fluxo de Dados



Dicionário de Dados de Manter TECNICO	
Manutenção_de_TECNICO	OP_TECNICO + PARAMETROS_TECNICO
Resposta_TECNICO	Se OP_TECNICO = 1, 2 ou 3, a resposta será uma mensagem de confirmação na tela. Se OP_TECNICO = 4, a resposta será o retorno de um registro.
OP_TECNICO	Indica o tipo da operação de manutenção. Possíveis valores assumidos: 1 -> Cadastrar 2 -> Atualizar 3 -> Excluir 4 -> Consultar
PARAMETROS_TECNICO	CODIGO_TECNICO + NOME_DO_TECNICO
CODIGO_TECNICO	Código do técnico.
NOME_DO_TECNICO	Nome do técnico.

## Miniespecificação de Manter TECNICO

### PROCEDIMENTO MANTER\_TECNICO

#### INÍCIO

Se OP\_TECNICO = 1

Encontre TECNICO com NOME\_DO\_TECNICO

Se encontrado

Mensagem "Técnico já cadastrado"

Senão

Crie uma nova entrada em SINTOMA com COD\_TECNICO único

Grave Registro\_TECNICO a partir de PARAMETROS\_TECNICO

Senão

Se OP\_TECNICO = 2

Encontre TECNICO com CODIGO\_TECNICO

Se encontrado

Se NOME\_DO\_TECNICO em PARAMETROS\_TECNICO não = "NULL"

Atualize Registro\_TECNICO com NOME\_DO\_TECNICO

Fim\_Se

Senão

Mensagem "CODIGO\_TECNICO inválido"

Senão

Se OP\_TECNICO = 3

Encontre TECNICO com CODIGO\_TECNICO

Se encontrado

Encontre OS\_TEC com CODIGO\_TECNICO

Se encontrado

Mensagem "Erro: Existem Ordens de Serviço vinculadas ao Técnico"

Senão

Exclua Registro\_TECNICO

Senão

Mensagem "CODIGO\_TECNICO inválido"

Senão

Se OP\_TECNICO = 4

Encontre TECNICO com CODIGO\_TECNICO

Se encontrado

Retorne Registro\_TECNICO

Senão

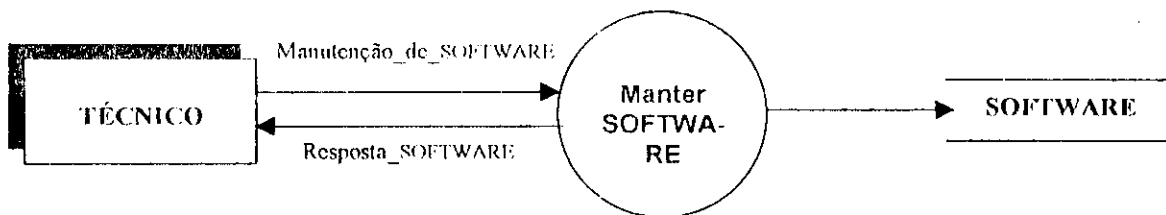
Mensagem "CODIGO\_TECNICO inválido"

Fim\_Se

**FIM PROCEDIMENTO MANTER\_TECNICO**

## 14ª Atividade: Manter SOFTWARE

### Diagrama de Fluxo de Dados



Dicionário de Dados de Cadastrar SOFTWARE	
Manutenção_de SOFTWARE	OP_SOFTWARE + PARAMETROS_SOFTWARE
Resposta SOFTWARE	Se OP_SOFTWARE = 1, 2 ou 3, a resposta será uma mensagem de confirmação na tela. Se OP_SOFTWARE = 4, a resposta será o retorno de um registro.
OP_SOFTWARE	Indica o tipo da operação de manutenção. Possíveis valores assumidos: 1 -> Cadastrar 2 -> Atualizar 3 -> Excluir 4 -> Consultar
PARAMETROS_SOFTWARE	CODIGO_SOFTWARE + NOME_DO_SOFTWARE
CODIGO_SOFTWARE	Código do software.
NOME_DO_SOFTWARE	Nome do software.

## Miniespecificação de Manter SOFTWARE

### PROCEDIMENTO MANTER\_SOFTWARE

#### INÍCIO

Se OP\_SOFTWARE = 1

Encontre SOFTWARE com NOME\_DO\_SOFTWARE

Se encontrado

Mensagem "Software já cadastrado"

Senão

Crie uma nova entrada em SOFTWARE com COD\_SOFTWARE único

Grave Registro\_SOFTWARE a partir de PARAMETROS\_SOFTWARE

Senão

Se OP\_SOFTWARE = 2

Encontre SOFTWARE com CODIGO\_SOFTWARE

Se encontrado

Se NOME\_DO\_SOFTWARE em PARAMETROS\_SOFTWARE não =  
"NULL"

Atualize Registro\_SOFTWARE com NOME\_DO\_SOFTWARE

Fim\_Se

Senão

Mensagem "CODIGO\_SOFTWARE inválido"

Senão

Se OP\_SOFTWARE = 3

Encontre SOFTWARE com CODIGO\_SOFTWARE

Se encontrado

Encontre COMP\_SOFT com CODIGO\_SOFTWARE

Se encontrado

Mensagem "Erro: Software presente em cadastros de  
Computadores"

Senão

Exclua Registro\_SOFTWARE

Senão

Mensagem "CODIGO\_SOFTWARE inválido"

Senão

Se OP\_SOFTWARE = 4

Encontre SOFTWARE com CODIGO\_SOFTWARE

Se encontrado

Retorne Registro\_SOFTWARE

Senão

Mensagem "CODIGO\_SOFTWARE inválido"

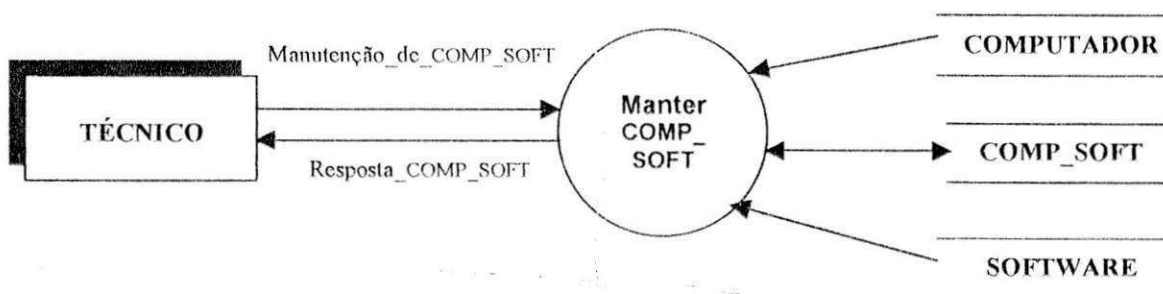
Fim\_Se

#### FIM PROCEDIMENTO MANTER\_SOFTWARE



## 15ª Atividade: Manter COMP SOFT

### Diagrama de Fluxo de Dados



Dicionário de Dados de Manter COMP_SOFT	
Manutenção_de_COMP_SOFT	OP_COMP_SOFT + PARAMETROS_COMP_SOFT
Resposta_COMP_SOFT	A resposta será uma mensagem de confirmação na tela.
OP_COMP_SOFT	Indica o tipo da operação de manutenção. Possíveis valores assumidos: 1 -> Vincular 2 -> Desvincular
PARAMETROS_COMP_SOFT	CODIGO_COMPUTADOR + CODIGO_SOFTWARE
CODIGO_COMPUTADOR	Código do computador.
CODIGO_SOFTWARE	Código do software.

## Miniespecificação de Manter COMP\_SOFT

### PROCEDIMENTO MANTER\_COMP\_SOFT

#### INÍCIO

Se OP\_COMP\_SOFT = 1

Encontre COMPUTADOR com CODIGO\_COMPUTADOR

Se encontrado

Encontre SOFTWARE com CODIGO\_SOFTWARE

Se encontrado

Encontre COMP\_SOFT com CODIGO\_COMPUTADOR e  
CODIGO\_SOFTWARE

Se encontrado

Mensagem "Vinculação já existe"

Senão

Grave Registro\_COMP\_SOFT

Senão

Mensagem "CODIGO\_SOFTWARE inválido"

Senão

Mensagem "CODIGO\_COMPUTADOR inválido"

Senão

Se OP\_COMP\_SOFT = 2

Se CODIGO\_COMPUTADOR não = "NULL"

Encontre COMPUTADOR com CODIGO\_COMPUTADOR

Se encontrado

Para cada COMP\_SOFT com CODIGO\_COMPUTADOR faça:

Exclua Registro\_COMP\_SOFT

Fim\_Para

Senão

Mensagem "CODIGO\_COMPUTADOR inválido"

Senão

Se CODIGO\_SOFTWARE não = "NULL"

Encontre SOFTWARE com CODIGO\_SOFTWARE

Se encontrado

Para cada COMP\_SOFT com CODIGO\_SOFTWARE faça:

Exclua Registro\_COMP\_SOFT

Fim\_para

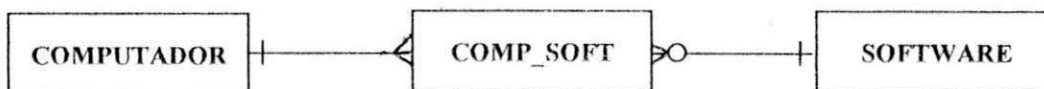
Senão

Mensagem "CODIGO\_SOFTWARE inválido"

Fim\_Se

FIM PROCEDIMENTO MANTER\_COMP\_SOFT

### DER



## 16ª Atividade: Manter ATIVIDADE

### Diagrama de Fluxo de Dados



Dicionário de Dados de Manter ATIVIDADE	
Manutenção_de_ATIVIDADE	OP_ATIVIDADE + PARAMETROS_ATIVIDADE
Resposta_ATIVIDADE	Se OP_ATIVIDADE = 1, 2 ou 3, a resposta será uma mensagem de confirmação na tela. Se OP_ATIVIDADE = 4, a resposta será o retorno de um registro.
OP_ATIVIDADE	Indica o tipo da operação de manutenção. Possíveis valores assumidos: 1 -> Cadastrar 2 -> Atualizar 3 -> Excluir 4 -> Consultar
PARAMETROS_ATIVIDADE	CODIGO_ATIVIDADE + NOME_DA_ATIVIDADE
CODIGO_ATIVIDADE	Código da atividade.
NOME_DA_ATIVIDADE	Nome da Atividade.

## Miniespecificação de Manter ATIVIDADE

### PROCEDIMENTO MANTER\_ATIVIDADE

#### INÍCIO

Se OP\_ATIVIDADE = 1

Crie uma nova entrada em ATIVIDADE com COD\_ATIVIDADE único  
Grave Registro\_ATIVIDADE a partir de PARAMETROS\_ATIVIDADE

Senão

Se OP\_ATIVIDADE = 2

Encontre ATIVIDADE com CODIGO\_ATIVIDADE

Se encontrado

Se NOME\_DO\_ATIVIDADE em PARAMETROS\_ATIVIDADE não =  
"NULL"

Atualize Registro\_ATIVIDADE com NOME\_DA\_ATIVIDADE

Fim\_Se

Senão

Mensagem "CODIGO\_ATIVIDADE inválido"

Senão

Se OP\_ATIVIDADE = 3

Encontre ATIVIDADE com CODIGO\_ATIVIDADE

Se encontrado

Encontre ATIVIDADE\_OS com CODIGO\_ATIVIDADE

Se encontrado

Mensagem "Erro: Existem Ordens de Serviço e Técnicos  
associados à Atividade"

Senão

Exclua Registro\_ATIVIDADE

Senão

Mensagem "CODIGO\_ATIVIDADE inválido"

Senão

Se OP\_ATIVIDADE = 4

Encontre ATIVIDADE com CODIGO\_ATIVIDADE

Se encontrado

Retorne Registro\_ATIVIDADE

Senão

Mensagem "CODIGO\_ATIVIDADE inválido"

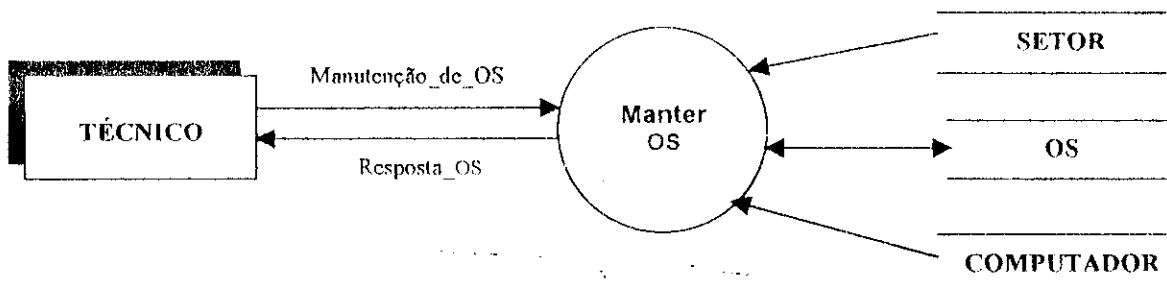
Fim\_Se

**FIM PROCEDIMENTO MANTER\_ATIVIDADE**



## 17ª Atividade: Manter OS

### Diagrama de Fluxo de Dados



Dicionário de Dados de Manter OS	
Manutenção_de_OS	OP_OS + PARAMETROS_OS
Resposta_OS	Mensagem de confirmação na tela.
OP_OS	Indica o tipo da operação de manutenção. Possíveis valores assumidos: 1 -> Atualizar 2 -> Excluir
PARAMETROS_OS	CODIGO_OS + CODIGO_SETOR + CODIGO_COMPUTADOR
CODIGO_OS	Código do ordem de serviço.
CODIGO_SETOR	Código do setor.
CODIGO_COMPUTADOR	Código do computador.

## Miniespecificação de Manter OS

### PROCEDIMENTO MANTER\_OS

#### INÍCIO

Se OP\_OS = 1

Encontre OS com CODIGO\_OS

Se encontrado

Se CODIGO\_SETOR em PARAMETROS\_OS não = "NULL"

Encontre SETOR com CODIGO\_SETOR

Se encontrado

Atualize Registro\_OS com CODIGO\_SETOR

Senão

Mensagem "CODIGO\_SETOR inválido"

Fim\_Se

Se CODIGO\_COMPUTADOR em PARAMETROS\_OS não = "NULL"

Encontre COMPUTADOR com CODIGO\_COMPUTADOR

Se encontrado

Atualize Registro\_OS com CODIGO\_COMPUTADOR

Senão

Mensagem "CODIGO\_COMPUTADOR inválido"

Fim\_Se

Senão

Mensagem "CODIGO\_OS inválido"

Senão

Se OP\_OS = 2

Encontre OS com CODIGO\_OS

Se encontrado

Exclua Registro\_OS

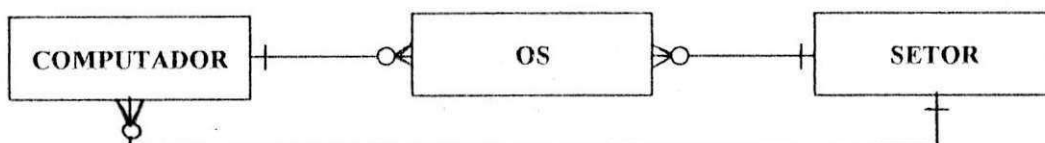
Senão

Mensagem "CODIGO\_OS inválido"

Fim\_Se

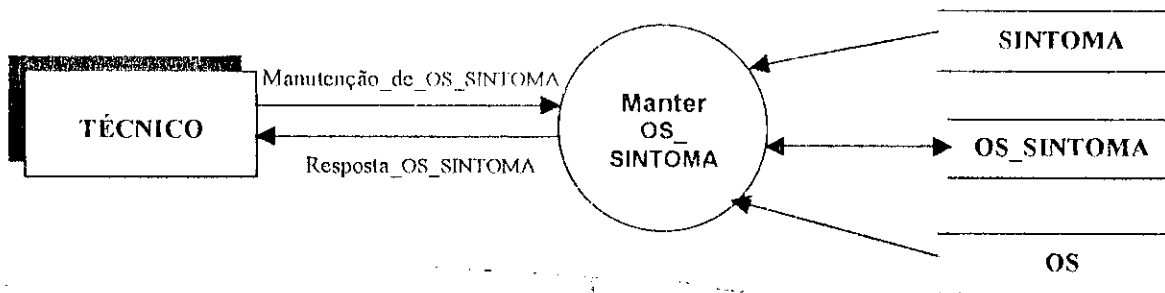
FIM PROCEDIMENTO MANTER\_OS

### DER



## 18ª Atividade: Manter OS SINTOMA

### Diagrama de Fluxo de Dados



Dicionário de Dados de Manter OS_SINTOMA	
Manutenção_de_OS_SINTOMA	OP_OS_SINTOMA + PARAMETROS_OS_SINTOMA
Resposta_OS_SINTOMA	Mensagem de confirmação na tela.
OP_OS_SINTOMA	Indica o tipo da operação de manutenção. Possíveis valores assumidos: 1 -> Vincular 2 -> Desvincular
PARAMETROS_OS_SITNOMA	CODIGO_OS + CODIGO_SINTOMA
CODIGO_OS	Código do ordem de serviço.
CODIGO_SINTOMA	Código do sintoma.

## Miniespecificação de Manter OS\_SINTOMA

### PROCEDIMENTO MANTER\_OS\_SINTOMA

#### INÍCIO

Se OP\_OS\_SINTOMA = 1

Encontre OS com CODIGO\_OS

Se encontrado

Encontre SINTOMA com CODIGO\_SINTOMA

Se encontrado

Encontre OS\_SINTOMA com CODIGO\_OS e CODIGO\_SINTOMA

Se encontrado

Mensagem "Erro: Vinculação já existe"

Senão

Grave Registro\_OS\_SINTOMA

Senão

Mensagem "CODIGO\_SINTOMA inválido"

Senão

Mensagem "CODIGO\_OS inválido"

Senão

Se OP\_OS\_SINTOMA = 2

Se CODIGO\_OS não = "NULL"

Encontre OS com CODIGO\_OS

Se encontrado

Para cada OS\_SINTOMA com CODIGO\_OS faça:

Exclua Registro\_OS\_SINTOMA

Fim\_Para

Senão

Mensagem "CODIGO\_OS inválido"

Senão

Se CODIGO\_SINTOMA não = "NULL"

Encontre SINTOMA com CODIGO\_SINTOMA

Se encontrado

Para cada OS\_SINTOMA com CODIGO\_SINTOMA faça:

Exclua Registro\_OS\_SINTOMA

Fim\_Para

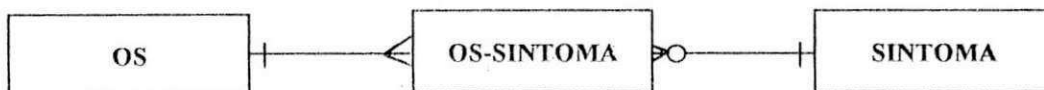
Senão

Mensagem "CODIGO\_SINTOMA inválido"

Fim\_Se

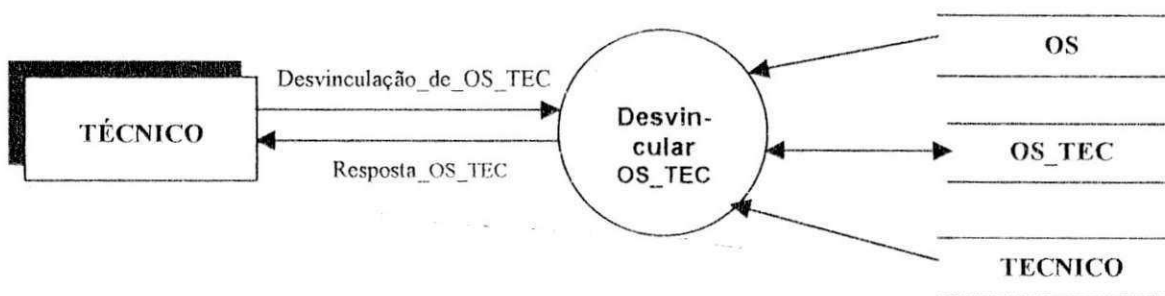
FIM PROCEDIMENTO MANTER\_OS\_SINTOMA

#### DER



## 19ª Atividade: Desvincular OS TEC

### Diagrama de Fluxo de Dados



Dicionário de Dados de Desvincular OS_TEC	
Manutenção_de_OS_TEC	PARAMETROS_OS_TEC
Resposta_OS_TEC	Mensagem de confirmação na tela.
PARAMETROS_OS_TEC	CODIGO_OS + CODIGO_TECNICO
CODIGO_OS	Código do ordem de serviço.
CODIGO_TECNICO	Código do técnico.



## Miniespecificação de Desvincular OS TEC

### PROCEDIMENTO DESVINCULAR\_OS\_TEC

#### INÍCIO

Se CODIGO\_OS não = "NULL"

Encontre OS com CODIGO\_OS

Se encontrado

Para cada OS\_TEC com CODIGO\_OS faça:

Exclua Registro\_OS\_TEC

Fim\_Para

Senão

Mensagem "CODIGO\_OS inválido"

Senão

Se CODIGO\_TECNICO não = "NULL"

Encontre TECNICO com CODIGO\_TECNICO

Se encontrado

Para cada OS\_TEC com CODIGO\_TECNICO faça:

Exclua Registro\_OS\_TEC

Fim\_Para

Senão

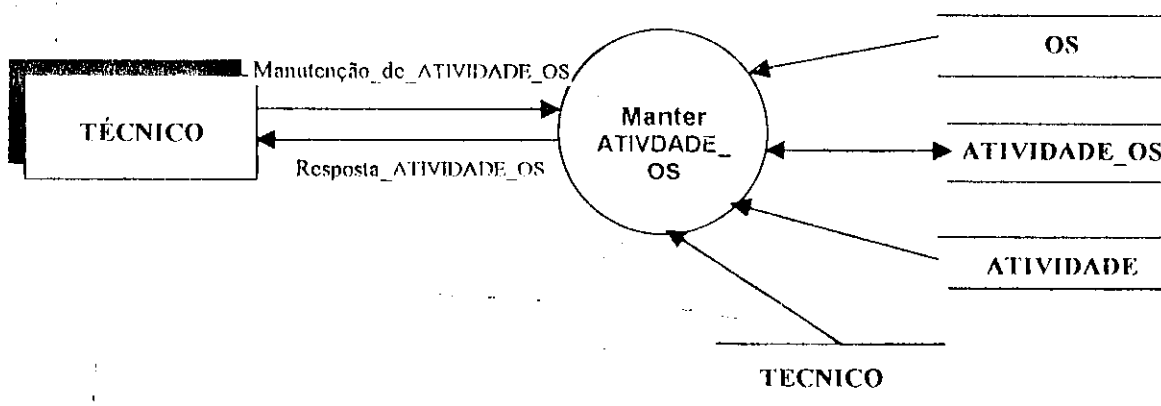
Mensagem "CODIGO\_TECNICO inválido"

Fim\_Se

**FIM PROCEDIMENTO DESVINCULAR\_OS\_TEC**

## 20ª Atividade: Manter ATIVIDADE OS

### Diagrama de Fluxo de Dados



<b>Dicionário de Dados de Manter ATIVIDADE_OS</b>	
Manutenção_de_ATIVIDADE_OS	OP_ATIVIDADE_OS + PARAMETROS_ATIVIDADE_OS
Resposta_ATIVIDADE_OS	Mensagem de confirmação na tela.
OP_ATIVIDADE_OS	Indica o tipo da operação de manutenção. Possíveis valores assumidos: 1 -> Atualizar 2 -> Desvincular
PARAMETROS_ATIVIDADE_OS	CODIGO_OS + CODIGO_TECNICO1 + CODIGO_ATIVIDADE1 + COMPLEMENTACAO + CODIGO_TECNICO2 + CODIGO_ATIVIDADE2
CODIGO_OS	Código do ordem de serviço.
CODIGO_TECNICO1, CODIGO_TECNICO2	Códigos de técnicos.
CODIGO_ATIVIDADE1, CODIGO_ATIVIDADE2	Códigos de atividades.
COMPLEMENTAÇÃO	Descrição mais detalhada da atividade.

## Miniespecificação de Manter ATIVIDADE\_OS

### PROCEDIMENTO MANTER\_ATIVIDADE\_OS

#### INÍCIO

Se OP\_ATIVIDADE\_OS = 1

Se CODIGO\_OS não = "NULL" e CODIGO\_TECNICO1 não = "NULL" e  
CODIGO\_ATIVIDADE1 não = "NULL"

Valide PARAMETROS\_ATIVIDADE\_OS

Se PARAMETROS\_ATIVIDADE\_OS válidos

Atualize Registro\_ATIVIDADE\_OS

Senão

Mensagem "Erro: Parâmetro(s) inválido(s) fornecido(s)"

Senão

Mensagem "Erro: Parâmetros insuficientes"

Senão

Se CODIGO\_OS não = "NULL"

Encontre OS com CODIGO\_OS

Se encontrado

Para cada ATIVIDADE\_OS com CODIGO\_OS faça:

Exclua Registro\_ATIVIDADE\_OS

Fim\_Para

Senão

Mensagem "CODIGO\_OS inválido"

Senão

Se CODIGO\_TECNICO1 não = "NULL"

Encontre TECNICO com CODIGO\_TECNICO

Se encontrado

Para cada ATIVIDADE\_OS com CODIGO\_TECNICO faça:

Exclua Registro\_ATIVIDADE\_OS

Fim\_Para

Senão

Mensagem "CODIGO\_TECNICO inválido"

Senão

Se CODIGO\_TECNICO1 não = "NULL"

Encontre TECNICO com CODIGO\_TECNICO

Se encontrado

Para cada ATIVIDADE\_OS com CODIGO\_TECNICO faça:

Exclua Registro\_ATIVIDADE\_OS

Fim\_Para

Senão

Mensagem "CODIGO\_TECNICO inválido"

Fim\_Se

**FIM PROCEDIMENTO MANTER\_ATIVIDADE\_OS**

# 4

## BANCO DE DADOS

```
create database Informatica
```

```
create table Computador (  
  CodComputador char(10) not null primary key,  
  TamMemRAM int,  
  TamHD int,  
  Diskett char(1),  
  CDRom char(1),  
  Atalhos varchar(200),  
  SistemaOperacional varchar(20),  
  CodigoSetor int references Setor(CodSetor) )
```

```
create table Setor(  
  CodSetor int not null identity(1,1) primary key,  
  NomeSetor varchar(30) not null)
```

```
create table OS(  
  CodOS int not null identity(1,1) primary key,  
  CodigoSetor int not null references Setor(CodSetor),  
  CodigoComputador char(10) not null references Computador(CodComputador),  
  DataHoraRegistro datetime default getdate(),  
  DataHoraFechamento datetime )
```

```
create table Sintoma (  
  CodSintoma int not null identity(1,1) primary key,  
  NomeSintoma varchar(100) not null )
```

```
create table OSSintoma (  
  CodigoOS int not null references OS(CodOS),  
  CodigoSintoma int not null references Sintoma(CodSintoma),  
  primary key (CodigoOS, CodigoSintoma) )
```

```
create table Tecnico (  
  CodTecnico int not null identity(1,1) primary key,  
  NomeTecnico varchar(40) not null )
```

```
create table OSTec (  
  CodigoOS int not null references OS(CodOS),  
  CodigoTecnico int not null references Tecnico(CodTecnico),  
  DataHoraVinculacao datetime default getdate(),  
  DataHoraDesvinculacao datetime,  
  primary key(CodigoOS, CodigoTecnico, DataHoraVinculacao) )
```

```
create table Software (  
  CodSoftware int not null identity(1,1) primary key,  
  NomeSoftware varchar(40) not null )
```

```
create table CompSoft (  
  CodigoComputador char(10) not null references Computador(CodComputador),  
  CodigoSoftware int not null references Software(CodSoftware),  
  primary key(CodigoComputador, CodigoSoftware) )
```

```
create table Atividade (  
  CodAtividade int not null identity(1,1) primary key,  
  NomeAtividade varchar(40) not null )
```

```
create table AtividadeOS (  
  CodigoOS int not null references OS(CodOS),  
  CodigoTecnico int not null references Tecnico(CodTecnico),  
  CodigoAtividade int not null references Atividade(CodAtividade),  
  DataHora datetime default getdate(),  
  Complemento varchar(100),  
  primary key(CodigoOS, CodigoTecnico, CodigoAtividade, DataHora) )
```



# 5

## Implementação do Sistema

### CADASTRO DE COMPUTADOR

```
//Verificar se o computador já está cadastrado
Computador.SetKey;
Computador.FieldName('CodComputador').AsString := Edit1.Text;
Computador.KeyFieldCount := 1;
//Se estiver, envie uma mensagem de erro
if Computador.GotoKey then
begin
  MessageDlg('Computador já cadastrado', mtError, [mbOK], 0);
  ...
end
else
//Se não estiver cadastrado e a quantidade de memória RAM não tiver sido fornecida
if ComboBox1.Text = "" then
begin
  MessageDlg('Selecionar quantidade de RAM', mtWarning, [mbOK], 0);
  ...
end
else
//Se tiver sido fornecida, mas a capacidade do HD não foi fornecida
if ComboBox2.Text = "" then
begin
  MessageDlg('Selecionar capacidade do HD', mtWarning, [mbOk], 0);
  ...
end
else
//Se todos os parâmetros foram fornecidos corretamente, grave o registro
begin
  Setor.Locate('NomeSetor', ComboBox3.Text, []);
  Computador.Insert;
  Computador.FieldName('CodComputador').AsString := Edit1.Text;
  Computador.FieldName('TamMemRAM').AsInteger := StrToInt(ComboBox1.Text);
  Computador.FieldName('TamHD').AsInteger := StrToInt(ComboBox2.Text);
  if CheckBox1.Checked then
    Computador.FieldName('Diskett').AsString := 'S'
  else
    Computador.FieldName('Diskett').AsString := 'N';
  if CheckBox2.Checked then
    Computador.FieldName('CDRom').AsString := 'S'
  else
    Computador.FieldName('CDRom').AsString := 'N';
  Computador.FieldName('Atalhos').AsString := Memo1.Text;
  Computador.FieldName('SistemaOperacional').AsString := Edit4.Text;
  Computador.FieldName('CodigoSetor').AsInteger := Setor.FieldName('CodSetor').AsInteger;
```

```

Computador.Post;
...
MsgDlg('Cadastro gravado com sucesso', mtInformation, [mbOK], 0);
end;

```

### CADASTRO DE SETOR

```

//Se o nome do setor não tiver sido fornecido, envie uma mensagem de aviso
if Edit1.Text = "" then
begin
  MsgDlg('Digite o nome do setor', mtWarning, [mbOK], 0);
  .....
end
//Senão, insira o registro na tabela Setor
else
begin
  Setor.Insert;
  Setor.FieldName('NomeSetor').AsString := Edit1.Text;
  Setor.Post;
  ...
end;

```

### CADASTRO DE SOFTWARES PARA UM COMPUTADOR

```

Achou := False;
//Se nenhum software foi selecionado, envie uma mensagem de aviso
if ComboBox1.Text = "" then
  MsgDlg('Selecione um software', mtWarning, [mbOK], 0)
else
begin
  //Se um software foi selecionado, mova para o primeiro registro da tabela CompSoft
  CompSoft.First;
  //Enquanto não chegar ao fim da tabela faça:
  while not CompSoft.EOF do
  //Se já existir uma vinculação entre o software e o computador em CompSoft,
  //envie uma mensagem de aviso e saia do laço
  if (CompSoft.FieldName('CodigoSoftware').AsInteger =
    Software.FieldName('CodSoftware').AsInteger) and
    (CompSoft.FieldName('CodigoComputador').AsString =
    CadCompForm.Edit1.Text) then
  begin
    MsgDlg('Software já listado no cadastro', mtWarning, [mbOK], 0);
    Achou := True;
    Break;
  end
  else
  //Senão mova para o próximo registro
  CompSoft.Next;
  //Se ainda não existe uma vinculação entre o software e o computador, grave o registro
  if Achou = False then

```

```

begin
  CompSoft.Insert;
  CompSoft.FieldName('CodigoComputador').AsString :=
    CadCompForm.Edit1.Text;
  CompSoft.FieldName('CodigoSoftware').AsInteger :=
    Software.FieldName('CodSoftware').AsInteger;
  CompSoft.Post;
  .....
end;
end;

```

## ATUALIZAÇÃO DE CADASTRO DE COMPUTADOR

### A. Pesquisando o código fornecido para retornar o seu cadastro

```

//Coloque a tabela Computador em estado de procura pelo código do computador fornecido
Computador.SetKey;
Computador.FieldName('CodComputador').AsString := Edit1.Text;
Computador.KeyFieldCount := 1;
//Se não encontrá-lo, envie uma mensagem de erro
if not Computador.GotoKey then
  begin
    MessageDlg('Codigo inválido', mtError, [mbOK], 0);
    ...
  end
//Se encontrá-lo, retorne o cadastro do computador e a lista de softwares
else
  begin
    ...
    ComboBox1.Text := IntToStr(Computador.FieldName('TamMemRAM').AsInteger);
    ComboBox2.Text := IntToStr(Computador.FieldName('TamHD').AsInteger);
    Setor.SetKey;
    Setor.FieldName('CodSetor').AsInteger := Computador.FieldName('CodigoSetor').AsInteger;
    Setor.KeyFieldCount := 1;
    Setor.GotoKey;
    ComboBox4.Text := Setor.FieldName('NomeSetor').AsString;
    Edit2.Text := Computador.FieldName('SistemaOperacional').AsString;
    if Computador.FieldName('Diskett').AsString = 'S' then
      CheckBox1.Checked := True
    else
      CheckBox1.Checked := False;
    if Computador.FieldName('CDRom').AsString = 'S' then
      CheckBox2.Checked := True
    else
      CheckBox2.Checked := False;
    Memo1.Text := Computador.FieldName('Atalhos').AsString;
    //Consulta para retornar a lista dos softwares do computador
    with Query1 do
      begin
        Close;
        SQL.Clear;
        SQL.Add('Select NomeSoftware from Software, CompSoft');
        SQL.Add('Where CodSoftware = CodigoSoftware and CodigoComputador = :vComputador');

```

```

    Params[0].AsString := Edit1.Text;
    ExecSQL;
    Open;
end;
end;

```

## B. Salvando as alterações feitas no cadastro de computador

```

Computador.Edit;
Computador.FieldName('TamMemRAM').AsInteger := StrToInt(ComboBox1.Text);
Computador.FieldName('TamHD').AsInteger := StrToInt(ComboBox2.Text);
if CheckBox1.Checked then
    Computador.FieldName('Diskett').AsString := 'S'
else
    Computador.FieldName('Diskett').AsString := 'N';
if CheckBox2.Checked then
    Computador.FieldName('CDRom').AsString := 'S'
else
    Computador.FieldName('CDRom').AsString := 'N';
Computador.FieldName('Atalhos').AsString := Memo1.Text;
Computador.FieldName('SistemaOperacional').AsString := Edit2.Text;
Computador.FieldName('CodigoSetor').AsInteger := Setor.FieldName('CodSetor').AsInteger;
Computador.Post;
Computador.Refresh;
...
MsgDlg('Alterações salvas com sucesso', mtInformation, [mbOK], 0);

```

## C. Incluindo e Excluindo softwares da lista de softwares do computador

```

//Se o software ainda não consta na lista, faça a inclusão
if Achou = False then
begin
    CompSoft.Insert;
    CompSoft.FieldName('CodigoComputador').AsString := Edit1.Text;
    CompSoft.FieldName('CodigoSoftware').AsInteger :=
        Software.FieldName('CodSoftware').AsInteger;
    CompSoft.Post;
end;
//Excluindo o software da lista
with Query2 do
begin
    Close;
    SQL.Clear;
    SQL.Add('Delete from CompSoft');
    SQL.Add('Where CodigoSoftware = :vSoftware and CodigoComputador = :vCodigo');
    Params[0].AsInteger := Software.FieldName('CodSoftware').AsInteger;
    Params[1].AsString := Edit1.Text;
    ExecSQL;
end;

```

## ATUALIZAÇÃO DE CADASTRO DE SETOR

```
// Se nenhum código foi fornecido, envie uma mensagem de aviso
if Edit1.Text = " then
  begin
    MessageDlg('Digite o código do setor a ser atualizado', mtWarning, [mbOK], 0);
    .....
  end
else
  .....
else
  //Envie uma mensagem informando o antigo e o novo nome do setor e espere a confirmação para atualizar
  case MessageDlg('Código do Setor: ' + Edit2.Text + ',#13 +
    'Antigo Nome do Setor: ' +
    Setor.FieldName('NomeSetor').AsString + ',#13 +
    'Novo Nome do Setor: ' + Edit3.Text + ',#13#13 +
    'Deseja prosseguir com a alteração?', mtConfirmation, [mbYes,
    mbNo], 0) of
  //Se o usuário confirmar, grave as alterações no banco de dados
  mrYes:
    begin
      Setor.Edit;
      Setor.FieldName('NomeSetor').AsString := Edit3.Text;
      MessageDlg('Alteração realizada com êxito', mtInformation, [mbOK], 0);
    end;
  end;
```

## EXCLUSÃO DE CADASTRO DE COMPUTADOR

### A. Função Pesquisar

```
//Se o código não tiver sido fornecido pelo usuário, envie uma mensagem de erro
if Edit1.Text = " then
  MessageDlg('Digite o código do computador a ser excluído', mtError, [mbOK], 0)
else
  begin
  //Senão, procure pelo código na tabela Computador
  Computador.SetKey;
  Computador.FieldName('CodComputador').AsString := Edit1.Text;
  Computador.KeyFieldCount := 1;
  //Se não encontrá-lo, envie uma mensagem de erro
  if not Computador.GotoKey then
    begin
      MessageDlg('Computador não encontrado', mtError, [mbOK], 0);
      ...
    end
  else
    begin
      ...
    end
  //Se encontrá-lo, forneça a lista de softwares presentes no computador
  with Query1 do
```



```

begin
  Close;
  SQL.Clear;
  SQL.Add('Select NomeSoftware from Software, CompSoft');
  SQL.Add('Where CodSoftware = CodigoSoftware and CodigoComputador = :vComputador');
  Params[0].AsString := Computador.FieldName('CodComputador').AsString;
  Open;
end;
end;
end;

```

## B. Função Excluir

```

//Listar todas as Ordens de Serviço que têm vínculo com o computador
with Query3 do
  begin
    Close;
    SQL.Clear;
    SQL.Add('Select CodOS from OS');
    SQL.Add('Where CodigoComputador = :vComp');
    Params[0].AsString := Computador.FieldName('CodComputador').AsString;
    Open;
  end;
//Se nao houver nenhuma
if Query3.IsEmpty then
  begin
//Remove todas as vinculações do computador com softwares na tabela CompSoft
with Query2 do
  begin
    Close;
    SQL.Clear;
    SQL.Add('Delete from CompSoft');
    SQL.Add('Where CodigoComputador = :vCodigo');
    Params[0].AsString := Computador.FieldName('CodComputador').AsString;
    ExecSQL;
  end;
  CompSoft.Refresh;
//Depois, remova o cadastro do computador do sistema
with CompQuery do
  begin
    Close;
    SQL.Clear;
    SQL.Add('Delete from Computador');
    SQL.Add('Where CodComputador = :vCodigo');
    Params[0].AsString := Edit1.Text;
    ExecSQL;
  end;
  ...
  Computador.Refresh;
  ...
end
//Se existem vinculações de Ordens de Serviço ao computador, perguntar ao usuário se ele deseja exclui-las
else
  case MessageDlg('Existem Ordens de Serviço vinculadas a esse Computador.#13
    'Deseja excluir o cadastro do Computador e ,assim, excluir tais

```

```

Ordens de Serviço?', mtConfirmation, [mbYes, mbNo], 0) of
//Caso a resposta seja sim,
mrYes:
begin
Query3.First;
while not Query3.EOF do
begin
ShowMessage('Query3: '+Query3.FieldName('CodOS').AsString);
// Excluir todas as ocorrências da OS em Query4 em OSSintoma
with Query2 do
begin
Close;
SQL.Clear;
SQL.Add('Delete from OSSintoma');
SQL.Add('Where CodigoOS = :vOS');
Params[0].AsInteger := Query3.FieldName('CodOS').AsInteger;
ExecSQL;
end;
// Excluir todas as ocorrências da OS em OSTec
with Query2 do
begin
Close;
SQL.Clear;
SQL.Add('Delete from OSTec');
SQL.Add('Where CodigoOS = :vOS');
Params[0].AsInteger := Query3.FieldName('CodOS').AsInteger;
ExecSQL;
end;
// Excluir todas as ocorrências da OS em AtividadeOS
with Query2 do
begin
Close;
SQL.Clear;
SQL.Add('Delete from AtividadeOS');
SQL.Add('Where CodigoOS = :vOS');
Params[0].AsInteger := Query3.FieldName('CodOS').AsInteger;
ExecSQL;
end;
Query3.Next;
end;
// Excluir todas as OS com o Computador a ser excluído
with Query2 do
begin
Close;
SQL.Clear;
SQL.Add('Delete from OS');
SQL.Add('Where CodigoComputador = :vComp');
Params[0].AsString := Computador.FieldName('CodComputador').AsString;
ExecSQL;
end;
// Excluir os Softwares do cadastro do Computador
with Query2 do
begin
Close;
SQL.Clear;
SQL.Add('Delete from CompSoft');

```

```

        SQL.Add('Where CodigoComputador = :vCodigo');
        Params[0].AsString := Computador.FieldName('CodComputador').AsString;
        ExecSQL;
    end;
    CompSoft.Refresh;
// Excluir o cadastro do Computador
    with CompQuery do
        begin
            Close;
            SQL.Clear;
            SQL.Add('Delete from Computador');
            SQL.Add('Where CodComputador = :vCodigo');
            Params[0].AsString := Edit1.Text;
            ExecSQL;
        end;
        ExcExcluirComp.Enabled := False;
        Computador.Refresh;
        ...
    end;
end;
end;

```

## EXCLUSÃO DE CADASTRO DE SETOR

### A. Função Pesquisar

```

//Se o código não tiver sido fornecido, envie uma mensagm de aviso
if Edit1.Text = "" then
    MessageDlg('Digite o código do setor a ser excluído', mtWarning, [mbOk], 0)
else
    begin
//Se o código tiver sido fornecido, procure-o na tabela Setor
        Setor.SetKey;
        Setor.FieldName('CodSetor').AsInteger := StrToInt(Edit1.Text);
        Setor.KeyFieldCount := 1;
//Se não encontrá-lo, envie uma mensagem de erro
        if not Setor.GotoKey then
            begin
                ...
                MessageDlg('Setor não encontrado', mtError, [mbOk], 0);
                ...
            end
        ...
    end;

```

### B. Função Excluir

```

//Verifique se existe algum computador vinculado ao setor
with Query1 do
    begin
        Close;
        SQL.Clear;

```

```

SQL.Add('Select CodComputador from Computador');
SQL.Add('Where CodigoSetor = :vSetor');
Params[0].AsInteger := Setor.FieldName('CodSetor').AsInteger;
Open;
end;
//Se não existir, remove o cadastro do setor
if Query1.IsEmpty then
begin
with SetorQuery do
begin
Close;
SQL.Clear;
SQL.Add('Delete from Setor');
SQL.Add('Where CodSetor = :vSetor');
Params[0].AsInteger := StrToInt(Edit1.Text);
ExecSQL;
end;
...
end
//Se existir, informe ao usuário e pergunte se ele deseja prosseguir com a exclusão
else
begin
case MessageDlg('Existem cadastros de computadores vinculados a esse Setor. #13'Deseja prosseguir com
a exclusão e, assim, #13'perder informações sobre esses cadastros e as Ordens de
Serviço a ele vinculadas?', mtConfirmation, [mbYes, mbNo], 0) of
//Caso a resposta seja sim
mrYes:
begin
//Liste todas as ordens de serviço que foram registradas para o setor
with Query2 do
begin
Close;
SQL.Clear;
SQL.Add('Select CodOS from OS');
SQL.Add('Where CodigoSetor =:vSetor');
Params[0].AsInteger := StrToInt(Edit1.Text);
Open;
end;
Query2.First;
//Para cada uma das ordens de serviço registradas para o setor:
while not Query2.EOF do
begin
// Excluir todas as ocorrências da ordem de serviço em OSSintoma
with SetorQuery do
begin
Close;
SQL.Clear;
SQL.Add('Delete from OSSintoma');
SQL.Add('Where CodigoOS = :vOS');
Params[0].AsInteger := Query2.FieldName('CodOS').AsInteger;
ExecSQL;
end;
// Excluir todas as ocorrências da ordem de serviço em OSTec
with SetorQuery do
begin
Close;

```

```

        SQL.Clear;
        SQL.Add('Delete from OSTec');
        SQL.Add('Where CodigoOS = :vOS');
        Params[0].AsInteger := Query2.FieldName('CodOS').AsInteger;
        ExecSQL;
    end;
// Excluir todas as ocorrências da ordem de serviço em AtividadeOS
with SetorQuery do
    begin
        Close;
        SQL.Clear;
        SQL.Add('Delete from AtividadeOS');
        SQL.Add('Where CodigoOS = :vOS');
        Params[0].AsInteger := Query2.FieldName('CodOS').AsInteger;
        ExecSQL;
    end;
    Query2.Next;
end;
// Excluir todas as ordens de serviço registradas para o setor a ser excluído
with SetorQuery do
    begin
        Close;
        SQL.Clear;
        SQL.Add('Delete from OS');
        SQL.Add('Where CodigoSetor = :vSetor');
        Params[0].AsInteger := Setor.FieldName('CodSetor').AsInteger;
        ExecSQL;
    end;
    Query1.First;
//Para cada um dos computadores vinculados ao setor:
while not Query1.EOF do
    begin
// Excluir a lista de softwares presentes no computador
with SetorQuery do
    begin
        Close;
        SQL.Clear;
        SQL.Add('Delete from CompSoft');
        SQL.Add('Where CodigoComputador = :vComp');
        Params[0].AsString := Query1.FieldName('CodComputador').AsString;
        ExecSQL;
    end;
    Query1.Next;
end;
// Excluir todos os cadastros de computadores vinculados ao setor a ser excluído
with SetorQuery do
    begin
        Close;
        SQL.Clear;
        SQL.Add('Delete from Computador');
        SQL.Add('Where CodigoSetor = :vSetor');
        Params[0].AsInteger := Setor.FieldName('CodSetor').AsInteger;
        ExecSQL;
    end;
// Excluir o Setor
with SetorQuery do

```



```

begin
  Close;
  SQL.Clear;
  SQL.Add('Delete from Setor');
  SQL.Add('Where CodSetor = :vSetor');
  Params[0].AsInteger := StrToInt(Edit1.Text);
  ExecSQL;
end;
...
end;
end;
end;

```

### CONSULTA A CADASTRO DE COMPUTADOR

```

//Se o código não tiver sido fornecido, envie uma mensagem de aviso
if Edit1.Text = "" then
  MessageDlg('Entre com o Código do Computador', mtWarning, [mbOK], 0)
//Senão
else
  begin
    ...
  //Procure pelo código
  ...
  Computador.SetKey;
  Computador.FieldName('CodComputador').AsString := Edit1.Text;
  Computador.KeyFieldCount := 1;
  //Se não encontra-lo, envie uma mensagem de erro
  if not Computador.GotoKey then
    begin
      MessageDlg('Computador não cadastrado', mtError, [mbOK], 0);
      ...
    end
  //Se encontrá-lo, retorne todos os campos do cadastro
  else
    begin
      ...
    //Liste todos os softwares presentes no computador
    with Query1 do
      begin
        Close;
        SQL.Clear;
        SQL.Add('Select NomeSoftware from Software, CompSoft');
        SQL.Add('Where CodSoftware = CodigoSoftware and CodigoComputador = :vComputador');
        Params[0].AsString := DBEdit1.Text;
        Open;
      end;
    end;
  end;
end;
end;

```

## CONSULTA A CADASTRO DE SETOR

```
//Se o código do setor não foi fornecido, envie uma mensagem de aviso
if Edit1.Text = " then
  MessageDlg('Entre com o Código do Setor', mtWarning, [mbOK], 0)
//Senão
else
  begin
    ...
//Procure pelo código na tabela Setor
    Setor.SetKey;
    Setor.FieldName('CodSetor').AsInteger := StrToInt(Edit1.Text);
    Setor.KeyFieldCount := 1;
//Se não encontrá-lo, envie uma mensagem de erro
    if not Setor.GotoKey then
      begin
        MessageDlg('Setor não encontrado', mtError, [mbOK], 0);
        ...
      end;
    end;
end;
```

## CONSULTA A ORDENS DE SERVIÇO PENDENTES

```
//Listando todas as Ordens de Serviço pendentes
with Query1 do
  begin
    Close;
    SQL.Clear;
    SQL.Add('Select CodOS, CodigoComputador, NomeSetor, DataHoraRegistro from OS, Setor');
    SQL.Add('Where CodigoSetor = CodSetor and DataHoraFechamento = NULL');
    Open;
  end;

//Listando técnicos, datas e horas de vinculações e desvinculações para a Ordem de Serviço selecionada na
primeira lista
with Query2 do
  begin
    Close;
    SQL.Clear;
    SQL.Add('Select NomeTecnico, DataHoraVinculacao, DataHoraDesvinculacao from OSTec, Tecnico');
    SQL.Add('Where CodigoTecnico = CodTecnico and CodigoOS = :vOS');
    Params[0].AsInteger := Query1.FieldName('CodOS').AsInteger;
    Open;
  end;
```

## CONSULTA A ORDENS DE SERVIÇO POR SETOR

### **A. Consulta realizada para ordens de serviço em aberto**

```

//Liste os computadores que possuem ordens de serviço nesse estado e que estão dentro do intervalo de
//consulta
with Query1 do
begin
Close;
SQL.Clear;
SQL.Add('Select Distinct CodigoComputador from OS');
SQL.Add('Where CodigoSetor = :vSetor and DataHoraFechamento = NULL and
DataHoraRegistro between :vData1 and :vData2');
Params[0].AsInteger := Setor.FieldName('CodSetor').AsInteger;
Params[1].AsDateTime := Data1;
Params[2].AsDateTime := Data2;
Open;
end;
Query1.First;
//Liste as Ordens de Serviço que pertencem ao computador selecionado
with Query2 do
begin
Close;
SQL.Clear;
SQL.Add('Select CodOS, DataHoraRegistro, DataHoraFechamento from OS');
SQL.Add('Where CodigoComputador = :vComp and DataHoraFechamento = NULL and
DataHoraRegistro between :vData1 and :vData2');
SQL.Add('Order By CodOS');
Params[0].AsString := Query1.FieldName('CodigoComputador').AsString;
Params[1].AsDateTime := Data1;
Params[2].AsDateTime := Data2;
Open;
end;
end

```

## B. Consulta realizada para ordens de serviço já fechadas

```

//Liste os computadores que possuem ordens de serviço nesse estado e que estão dentro do intervalo de
//consulta
with Query1 do
begin
Close;
SQL.Clear;
SQL.Add('Select Distinct CodigoComputador from OS');
SQL.Add('Where CodigoSetor = :vSetor and DataHoraFechamento <> NULL and
DataHoraRegistro between :vData1 and :vData2');
Params[0].AsInteger := Setor.FieldName('CodSetor').AsInteger;
Params[1].AsDateTime := Data1;
Params[2].AsDateTime := Data2;
Open;
end;
Query1.First;
//Liste as Ordens de Serviço que pertencem ao computador selecionado
with Query2 do
begin
Close;
SQL.Clear;
SQL.Add('Select CodOS, DataHoraRegistro, DataHoraFechamento from OS');
SQL.Add('Where CodigoComputador = :vComp and DataHoraFechamento <> NULL and

```

```

DataHoraRegistro between :vData1 and :vData2');
SQL.Add('Order By CodOS');
Params[0].AsString := Query1.FieldName('CodigoComputador').AsString;
Params[1].AsDateTime := Data1;
Params[2].AsDateTime := Data2;
Open;
end;
end

```

### C. Consulta realizada para todas as ordens de serviço

```

//Liste os computadores que possuem ordens de serviço nesse estado e que estão dentro do intervalo de
//consulta
with Query1 do
begin
Close;
SQL.Clear;
SQL.Add('Select Distinct CodigoComputador from OS');
SQL.Add('Where CodigoSetor = :vSetor and DataHoraRegistro between :vData1 and :vData2');
Params[0].AsInteger := Setor.FieldName('CodSetor').AsInteger;
Params[1].AsDateTime := Data1;
Params[2].AsDateTime := Data2;
Open;
end;
Query1.First;
//Liste as Ordens de Serviço que pertencem ao computador selecionado
with Query2 do
begin
Close;
SQL.Clear;
SQL.Add('Select CodOS, DataHoraRegistro, DataHoraFechamento from OS');
SQL.Add('Where CodigoComputador = :vComp and DataHoraRegistro between :vData1 and
:vData2');
SQL.Add('Order By CodOS');
Params[0].AsString := Query1.FieldName('CodigoComputador').AsString;
Params[1].AsDateTime := Data1;
Params[2].AsDateTime := Data2;
Open;
end;
end;

```

### D. Código válido para qualquer um dos três tipos de consulta, na obtenção das listas de técnicos e sintomas para uma ordem selecionada

```

//Listar todos os técnicos que foram vinculadas a ordem de serviço selecionada, juntamente com a data e a
//hora em que as vinculações e desvinculações ocorreram
with Query4 do
begin
Close;
SQL.Clear;
SQL.Add('Select NomeTecnico, DataHoraVinculacao, DataHoraDesvinculacao from OSTec,
Tecnico');
SQL.Add('Where CodigoOS = :vOS and CodigoTecnico = CodTecnico');
Params[0].AsInteger := Query2.FieldName('CodOS').AsInteger;

```

```

    Open;
end;
//Listar os sintoma apresentados pela OS quando da sua abertura
with Query5 do
begin
    Close;
    SQL.Clear;
    SQL.Add('Select NomeSintoma from OSSintoma, Sintoma');
    SQL.Add('Where CodigoOS = :vOS and CodigoSintoma = CodSintoma');
    Params[0].AsInteger := Query2.FieldByName('CodOS').AsInteger;
    Open;
end;
end

```

## CONSULTA A ORDENS DE SERVIÇO POR TÉCNICO

### A. Consulta realizada para ordens de serviço em aberto

```

//Liste todas as ordens de serviço em aberto que foram atribuídas ao técnico e que encontram-se dentro do
intervalo de consulta
with Query1 do
begin
    Close;
    SQL.Clear;
    SQL.Add('Select Distinct CodOS, DataHoraRegistro, DataHoraFechamento, CodigoComputador
from OS, OSTec');
    SQL.Add('Where CodOS = CodigoOS and CodigoTecnico = :vTecnico and DataHoraFechamento =
NULL and DataHoraRegistro between :vData1 and :vData2');
    Params[0].AsInteger := Tecnico.FieldByName('CodTecnico').AsInteger;
    Params[1].AsDateTime := Data1;
    Params[2].AsDateTime := Data2;
    Open;
end

```

### B. Consulta realizada para ordens de serviço já fechadas

```

//Liste todas as ordens de serviço fechadas que foram atribuídas ao técnico e que encontram-se dentro do
//intervalo de consulta
with Query1 do
begin
    Close;
    SQL.Clear;
    SQL.Add('Select Distinct CodOS, DataHoraRegistro, DataHoraFechamento, CodigoComputador
from OS, OSTec');
    SQL.Add('Where CodOS = CodigoOS and CodigoTecnico = :vTecnico and DataHoraFechamento
<> NULL and DataHoraRegistro between :vData1 and :vData2');
    Params[0].AsInteger := Tecnico.FieldByName('CodTecnico').AsInteger;
    Params[1].AsDateTime := Data1;
    Params[2].AsDateTime := Data2;
    Open;

```



end

### C. Consulta realizada para todas as ordens de serviço

```
//Liste todas as ordens de serviço que foram atribuídas ao técnico e que encontram-se dentro do intervalo de
//consulta
with Query1 do
begin
Close;
SQL.Clear;
SQL.Add('Select Distinct CodOS, DataHoraRegistro, DataHoraFechamento, CodigoComputador
from OS, OSTec');
SQL.Add('Where CodOS = CodigoOS and CodigoTecnico = :vTecnico and DataHoraRegistro
between :vData1 and :vData2');
Params[0].AsInteger := Tecnico.FieldName('CodTecnico').AsInteger;
Params[1].AsDateTime := Data1;
Params[2].AsDateTime := Data2;
Open;
end;
end;
```

### D. Código válido para qualquer um dos três tipos de consulta, na obtenção das listas de sintomas, de vinculações/desvinculações e de atividades para uma ordem selecionada

```
//Listar os sintoma apresentados pela ordem de serviço selecionada quando da sua abertura
with Query2 do
begin
Close;
SQL.Clear;
SQL.Add('Select NomeSintoma from OSSintoma, Sintoma');
SQL.Add('Where CodigoOS = :vOS and CodigoSintoma = CodSintoma');
Params[0].AsInteger := Query1.FieldName('CodOs').AsInteger;
Open;
end;
//Liste a data e a hora das vinculações e desvinculações estabelecidas entre a ordem de serviço e o técnico
with Query3 do
begin
Close;
SQL.Clear;
SQL.Add('Select DataHoraVinculacao, DataHoraDesvinculacao from OSTec');
SQL.Add('Where CodigoOS = :vOS and CodigoTecnico = :vTec');
Params[0].AsInteger := Query1.FieldName('CodOS').AsInteger;
Params[1].AsInteger := Tecnico.FieldName('CodTecnico').AsInteger;
Open;
end;
//Liste as atividades realizadas pelo técnico quando vinculado a ordem de serviço, assim como a data e a hora
//que foram registradas
with Query4 do
begin
Close;
SQL.Clear;
```

```

SQL.Add('Select NomeAtividade, DataHora, Complemento from AtividadeOS, Atividade');
SQL.Add('Where CodigoOS = :vOS and CodigoTecnico = :vTec and CodigoAtividade =
CodAtividade');
Params[0].AsInteger := Query1.FieldName('CodOS').AsInteger;
Params[1].AsInteger := Tecnico.FieldName('CodTecnico').AsInteger;
Open;
end;

```

### SOLICITAÇÃO DE SERVIÇO

```

Setor.Locate('NomeSetor', ComboBox3.Text, []);
Computador.Locate('CodComputador', ComboBox2.Text, []);
//Se o computador não pertencer ao setor, envie uma mensagem de erro
if Computador.FieldName('CodigoSetor').AsInteger <> Setor.FieldName('CodSetor').AsInteger then
begin
    MessageDlg('Computador não vinculado ao Setor', mtError, [mbOK], 0);
    ...
end
//Se pertencer
else
begin
//Faça a contagem de quantas ordens de serviço em aberto existem para o computador
with Query1 do
begin
    Close;
    SQL.Clear;
    SQL.Add('Select NUM = COUNT(*) from OS');
    SQL.Add('Where CodigoComputador = :vComputador and CodigoSetor = :vSetor');
    Params[0].AsString := ComboBox2.Text;
    Params[1].AsInteger := Setor.FieldName('CodSetor').AsInteger;
    Open;
end;
//Se não existir nenhuma, grave os dados fornecidos na tabela OS
if Query1.FieldName('NUM').AsInteger = 0 then
begin
    OS.Insert;
    OS.FieldName('CodigoSetor').AsInteger := Setor.FieldName('CodSetor').AsInteger;
    OS.FieldName('CodigoComputador').AsString := ComboBox2.Text;
    GetLocalTime(SystemTime);
    OS.FieldName('DataHoraRegistro').AsDateTime := SystemTimeToDateTime(SystemTime);
    OS.Post;
    ...
//Para cada sintoma na lista de sintomas apresentados, grave-o na tabela OSSintoma
for i:= 0 to (ListBox1.Items.Count - 1) do
begin
    Sintoma.Locate('NomeSintoma', ListBox1.Items.Strings[i], []);
    OSSintoma.Insert;
    OSSintoma.FieldName('CodigoOS').AsInteger := OS.FieldName('CodOS').AsInteger;
    OSSintoma.FieldName('CodigoSintoma').AsInteger :=
        Sintoma.FieldName('CodSintoma').AsInteger;
end;
OSSintoma.Refresh;

```

```

...
    MessageDlg('Ordem de Serviço aberta com sucesso'#13'Código: ' +
              IntToStr(OS.FieldByName('CodOs').AsInteger) + ", mtInformation, [mbOk], 0);
end
//Se já existir uma ordem de serviço em aberto para o computador, envie uma mensagem de erro
else
begin
    MessageDlg('Já existe uma ordem de serviço em aberto'#13'para o computador de código ' +
              ComboBox2.Text + ", mtError, [mbOK], 0);
...
end;
end;

```

### DISTRIBUIÇÃO DE SERVIÇO

```

//Verifique se o código da ordem de serviço existe na tabela OS
OS.SetKey;
OS.FieldByName('CodOS').AsInteger := StrToInt(Edit1.Text);
OS.KeyFieldCount := 1;
//Se não existir, envie uma mensagem de erro
if not OS.GotoKey then
begin
    MessageDlg('Código da Ordem de Serviço inválido', mtError, [mbOK], 0);
    Edit1.SetFocus;
end
//Se existir
else
begin
//Verifique se a ordem de serviço já foi vinculada a um outro técnico
with Query1 do
begin
    Close;
    SQL.Clear;
    SQL.Add('Select * from OSTec');
    SQL.Add('Where CodigoOS = :vOS and DataHoraDesvinculacao = NULL');
    Params[0].AsInteger := StrToInt(Edit1.Text);
    Open;
end;
//Se foi, envie uma mensagem de erro
if Query1.Locate('CodigoOS', StrToInt(Edit1.Text), []) then
begin
    MessageDlg('Ordem de Serviço já se encontra'#13'vinculada a um Técnico', mtError, [mbOK], 0);
...
end
//Senão
else
...
//Verifique se o técnico já se encontra vinculado a uma outra ordem de serviço
with Query1 do
begin
    Close;
    SQL.Clear;
    SQL.Add('Select * from OSTec');

```

```

    SQL.Add('Where CodigoTecnico = :vTecnico and DataHoraDesvinculacao = NULL');
    Params[0].AsInteger := Tecnico.FieldName('CodTecnico').AsInteger;
    Open;
    end;
//Se não estiver
    if not Query1.Locate('CodigoTecnico', Tecnico.FieldName('CodTecnico').AsInteger, []) then
        begin
//Grave os dados na tabela OSTec
            with Query2 do
                begin
                    Close;
                    SQL.Clear;
                    SQL.Add('Insert into OSTec(CodigoOS, CodigoTecnico)');
                    SQL.Add('Values (:vOS,:vTecnico)');
                    Params[0].AsInteger := StrToInt(Edit1.Text);
                    Params[1].AsInteger := Tecnico.FieldName('CodTecnico').AsInteger;
                    ExecSQL;
                end;
                ...
                MessageDlg('Técnico vinculado à Ordem de Serviço com sucesso', mtInformation, [mbOK], 0);
            end
//Se o técnico estiver vinculado a uma outra ordem de serviço, envie uma mensagem de erro
        else
            begin
                MessageDlg('Técnico já se encontra vinculado'#13'a outra Ordem de Serviço', mtError, [mbOk], 0);
                ...
            end;
        end;
    end;
    ...

```

### DISPENSA DE TÉCNICO

```

//Procure pelo código da ordem de serviço na tabela OS
OS.SetKey;
OS.FieldName('CodOS').AsInteger := StrToInt(Edit1.Text);
OS.KeyFieldCount := 1;
//Se não encontrá-lo, envie uma mensagem de erro
if not OS.GotoKey then
    begin
        MessageDlg('Código da Ordem de Serviço inválido', mtError, [mbOK], 0);
        Edit1.SetFocus;
    end
//Se encontrá-lo
else
//Se não encontrar o código do técnico na tabela OSTec, envie uma mensagem de erro
    if not OSTec.Locate('CodigoOS', StrToInt(Edit1.Text), []) then
        begin
            MessageDlg('Ordem de Serviço ainda não vinculada a um Técnico', mtWarning, [mbOK], 0);
            ...
        end
//Se encontrar
    else
        ...
//Verifique se existe uma vinculação do técnico a ordem de serviço

```

```

with Query1 do
begin
  Close;
  SQL.Clear;
  SQL.Add('Select * from OSTec');
  SQL.Add('Where CodigoOs = :vOS and CodigoTecnico = :vTecnico');
  Params[0].AsInteger := StrToInt(Edit1.Text);
  Params[1].AsInteger := Tecnico.FieldByName('CodTecnico').AsInteger;
  Open;
end;
//Se existir
if Query1.Locate('CodigoTecnico', Tecnico.FieldByName('CodTecnico').AsInteger, []) then
//Se a data e a hora de desvinculação ainda não tiverem sido gravadas
if Query1.Locate('DataHoraDesvinculacao', NULL, []) then
begin
//Grave-as na tabela OSTec
with Query2 do
begin
  Close;
  SQL.Clear;
  SQL.Add('Update OSTec');
  SQL.Add('Set DataHoraDesvinculacao = getdate()');
  SQL.Add('Where CodigoOS = :vOS and CodigoTecnico = :vTecnico');
  Params[0].AsInteger := StrToInt(Edit1.Text);
  Params[1].AsInteger := Tecnico.FieldByName('CodTecnico').AsInteger;
  ExecSQL;
end;
  MessageDlg('Técnico dispensado', mtInformation, [mbOk], 0);
  ...
end
//Se a data e a hora já tiverem sido gravadas, envie uma mensagem de erro
else
begin
  MessageDlg('Técnico já foi dispensado', mtError, [mbOk], 0);
  ...
end
//Se não existir uma vinculação técnico à ordem de serviço, envie uma mensagem de erro
else
begin
  MessageDlg('Técnico não vinculado a essa Ordem de Serviço', mtError, [mbOk], 0);
  ...
end;
  ...

```

## REGISTRO DE ATIVIDADE

```

//Verifique se existe alguma vinculação do técnico à ordem de serviço
With Query2 do
begin
  Close;
  SQL.Clear;
  SQL.Add('Select NUM = COUNT(*) from OSTec');

```



```

SQL.Add('Where CodigoOS = :vOS and CodigoTecnico = :vTecnico');
Params[0].AsInteger := StrToInt(ComboBox1.Text);
Params[1].AsInteger := Tecnico.FieldByName('CodTecnico').AsInteger;
Open;
end;
//Se não houver, envie uma mensagem de erro
if Query2.FieldByName('NUM').AsInteger = 0 then
begin
  MessageDlg('Técnico não vinculado à Ordem de Serviço', mtError, [mbOK], 0);
  ...
end
//Se houver
else
begin
//Verifique se o técnico ainda permanece vinculado à ordem de serviço
  With Query2 do
    begin
      Close;
      SQL.Clear;
      SQL.Add('Select NUM = COUNT(*) from OSTec');
      SQL.Add('Where CodigoOS = :vOS and CodigoTecnico = :vTecnico and DataHoraDesvinculacao =
NULL');
      Params[0].AsInteger := StrToInt(ComboBox1.Text);
      Params[1].AsInteger := Tecnico.FieldByName('CodTecnico').AsInteger;
      Open;
    end;
//Se não estiver mais vinculado, envie mensagem de erro
    if Query2.FieldByName('NUM').AsInteger = 0 then
      begin
        MessageDlg('Técnico não mais responsável pela Ordem de Serviço', mtError, [mbOK], 0);
        ...
      end
//Se ainda estiver vinculado
    else
      if ComboBox3.Text = " then
        begin
          MessageDlg('Selecione uma Atividade', mtWarning, [mbOK], 0);
          ...
        end
      else
        begin
          ...
//Verifique se já existe um registro dessa atividade para esse técnico e essa ordem
de serviço
          with Query2 do
            begin
              Close;
              SQL.Clear;
              SQL.Add('Select NUM = COUNT(*) from AtividadeOS');
              SQL.Add('Where CodigoOS = :vOS and CodigoTecnico = :vTecnico and CodigoAtividade =
:vAtividade');
              Params[0].AsInteger := StrToInt(ComboBox1.Text);
              Params[1].AsInteger := Tecnico.FieldByName('CodTecnico').AsInteger;
              Params[2].AsInteger := Atividade.FieldByName('CodAtividade').AsInteger;
              Open;
            end;

```

```

//Se não existe, grave os dados na tabela AtividadeOS
if Query2.FieldByName('NUM').AsInteger = 0 then
begin
AtividadeOS.Insert;
AtividadeOS.FieldByName('CodigoOS').AsInteger := StrToInt(ComboBox1.Text);
AtividadeOS.FieldByName('CodigoTecnico').AsInteger :=
Tecnico.FieldByName('CodTecnico').AsInteger;
AtividadeOS.FieldByName('CodigoAtividade').AsInteger :=
Atividade.FieldByName('CodAtividade').AsInteger;
AtividadeOS.FieldByName('Complemento').AsString := Memo1.Text;
AtividadeOS.Post;
MessageDlg('Atividade registrada com sucesso', mtInformation, [mbOk], 0);
...
end
//Se existe, envie uma mensagem de erro
else
begin
MessageDlg('Atividade já registrada', mtError, [mbOK], 0);
...
end;
end;
end;

```

## FECHAMENTO DE ORDEM DE SERVIÇO

```

//Verifique se a ordem de serviço já foi vinculada a algum técnico
with Query2 do
begin
Close;
SQL.Clear;
SQL.Add('Select NUM = COUNT(*) from OSTec');
SQL.Add('Where CodigoOS = :vOS');
Params[0].AsInteger := StrToInt(ComboBox1.Text);
Open;
end;
//Se não foi, então não pode ser fechada
if Query2.FieldByName('NUM').AsInteger = 0 then
begin
MessageDlg('Ordem de Serviço não pode ser fechada. #13'Ainda não vinculada a um Técnico', mtError,
[mbOK], 0);
...
end
//Se já foi vinculada a pelo menos um técnico
else
begin
//Verifique se existe algum técnico que precisa ser desvinculado da ordem de serviço
with Query2 do
begin
Close;
SQL.Clear;
SQL.Add('Select NUM = COUNT(*) from OSTec');
SQL.Add('Where CodigoOS = :vOS and DataHoraDesvinculacao = NULL');
Params[0].AsInteger := StrToInt(ComboBox1.Text);
Open;

```

```

end;
//Se não houver
if Query2.FieldName('NUM').AsInteger = 0 then
begin
//Grave a data e a hora do fechamento da ordem de serviço na tabela OS
with Query2 do
begin
Close;
SQL.Clear;
SQL.Add('Update OS');
SQL.Add('Set DataHoraFechamento = getdate()');
SQL.Add('Where CodOS = :vOS');
Params[0].AsInteger := StrToInt(ComboBox1.Text);
ExecSQL;
end;
MessageDlg('Ordem de Serviço fechada com sucesso', mtInformation, [mbOK], 0);
...
end
//Se houver um técnico a ser desvinculado da ordem de serviço
else
begin
case MessageDlg('Técnico vinculado à Ordem de Serviço ainda não dispensado'#13'Deseja dispensá-lo
e, depois, fechá-la', mtConfirmation, [mbYes, mbNo], 0) of
mrYes:
begin
//Grave a data e a hora da desvinculação na tabela OSTec
with Query2 do
begin
Close;
SQL.Clear;
SQL.Add('Update OSTec');
SQL.Add('Set DataHoraDesvinculacao = getdate()');
SQL.Add('Where CodigoOS = :vOS and DataHoraDesvinculacao = NULL');
Params[0].AsInteger := StrToInt(ComboBox1.Text);
ExecSQL;
end;
//Grave a data e a hora do fechamento da ordem de serviço na tabela OS
with Query2 do
begin
Close;
SQL.Clear;
SQL.Add('Update OS');
SQL.Add('Set DataHoraFechamento = getdate()');
SQL.Add('Where CodOS = :vOS');
Params[0].AsInteger := StrToInt(ComboBox1.Text);
ExecSQL;
end;
MessageDlg('Ordem de Serviço fechada com sucesso', mtInformation, [mbOK], 0);
...
end;
...
end;
end;

```