

# Protótipo SGS-TI

[Digite o subtítulo do documento]

01/05/2010

PUCPR

André N. Schuster, Denis F. Gurski e Vanessa Iwamoto

## LOGIN

---

**A)Funcionalidade:** O Login tem como objetivo identificar o usuário que irá logar no sistema, esta funcionalidade é essencial para limitar o acesso dos usuário de acordo com seu perfil.

**B)Autor:** Denis F. Gurski

**C)Protótipo da funcionalidade:**



**D)Informações técnicas:** O Login realizará uma consulta ao banco de dados para validar o usuário e senha, após a autenticação o sistema iniciará uma Thread que se responsabilizará por verificar o tempo de inatividade do usuário, cujo qual será determinado pelo administrador ou seguindo um padrão da aplicação.

## CADASTRAR CHAMADO

**A)Funcionalidade:** Um chamado corresponde a uma atividade a ser realizada pelo técnico. Um chamado é composto de um cliente, responsável pelo atendimento, endereço do cliente, tipo de falha, prioridade e sua descrição. Caso o usuário seja do tipo Atendente, o mesmo poderá selecionar um cliente através do botão BUSCAR, cujo qual abrirá uma MODAL, responsável por buscar clientes (Genérica para todas as telas que utilizarem tal recurso). Após selecionado o cliente, os campos Responsável e endereço serão preenchidos automaticamente.

Caso seja do tipo Cliente, os campos estarão preenchidos automaticamente, podendo alterar apenas Tipo falha, prioridade e sua descrição.

**B)Autor:** Denis F. Gurski

**C)Protótipo da funcionalidade:**

O protótipo da funcionalidade de cadastrar um chamado apresenta a seguinte estrutura:

- Barra Superior:** Contém as abas "Chamados", "Clientes" e "Agenda".
- Menu Lateral:** Localizado à esquerda, contém os botões "Cadastrar Chamado", "Editar Chamado" e "Listar Chamados".
- Formulário Principal:**
  - Cliente:** Campo de texto com o valor "PUC-PR" e um botão "Buscar" à direita.
  - Responsável:** Campo de texto com o valor "Márcio Fuckner".
  - Endereço:** Campo de texto com o valor "Rua Imaculada Conceição, 1155".
  - Tipo falha:** Menu suspenso com o valor "Periféricos" selecionado.
  - Prioridade:** Menu suspenso com o valor "Alta" selecionado.
  - Descrição:** Área de texto com o conteúdo: "Os ACCESS POINT's da PUC não estão funcionando, os alunos estão irritados, precisamos de uma solução o mais rápido possível."
  - Setor de TI:** Campo de texto com o valor "PUC-PR".
- Botões de Ação:** Localizados no rodapé do formulário, são os botões "Salvar" e "Cancelar".

**D)Informações técnicas:** Ao cadastrar um chamado, o mesmo será encaminhado para um repositório comum de objetos remotos, para que seja listado na funcionalidade LISTAR CHAMADOS.

## EDITAR CHAMADO

A) **Funcionalidade:** A tela poderá ser visualizada e editada pelo atendente, técnico e o gerente operacional. Basicamente poderá alterar o Responsável (Pessoa do Cliente que solicita ajuda), Tipo da Falha, Prioridade e sua Descrição.

B) **Autor:** Vanessa Iwamoto

C) **Protótipo da funcionalidade:**

O protótipo da funcionalidade 'Editar Chamado' apresenta uma interface com uma barra superior contendo as abas 'Chamados', 'Clientes' e 'Agenda'. À esquerda, há um menu lateral com os botões 'Cadastrar Chamado', 'Editar Chamado' e 'Listar Chamados'. O formulário principal é dividido em campos para preenchimento de dados:

- Data Abertura:** 25/03/2010
- Aberto por:** vaneiw
- Cliente:** PUC Paraná
- Responsável:** Márcio Fuckner
- Tipo de Falha:** Hardware
- Status:** Agendado
- Prioridade:** Alta
- Data:** 25/03/2010
- Hora:** 17:00
- Descrição:** Roteadores wireless não estão funcionando corretamente.

Na base da tela, há dois botões: 'Salvar' e 'Cancelar'.

D) **Informações técnicas:** Ao editar um chamado será encaminhado para um repositório comum de objetos remotos, para que seja listado na funcionalidade LISTAR CHAMADOS. Caso o técnico determine uma data e hora, devera implementar um Observable para avisar a LISTAR AGENDA, que foi alterado/inserido um agendamento.

### C)Protótipo da funcionalidade:

**D) Informações técnicas:** Utilizaremos RMI para troca de objetos com o service que irá conter os chamados. Para o “auto-refresh” da lista, utilizaremos uma thread que irá realizar consultas ao service periodicamente. Para a notificação de mensagens do “tray icon” utilizaremos o padrão observer/observable.

**A)Funcionalidade:** A tela poderá ira exibir a agenda do técnico. Onde ele poderá fazer uma busca por um determinado chamado, ou filtrar mensagem com certa palavra. Podendo também visualizar todos seus chamados a serem atendidos, em qual cliente, qual prioridade, qual endereço e contato do cliente.

### C) Protótipo da funcionalidade:

**D) Informações técnicas:** No listar agenda terá uma thread que ira alertar o técnico quando uma visita ira ocorrer no dia, e próximo ao horário atual. Terá um Observer que ficara esperando um chamado de editar chamado avisando que mudou ou inseriu um novo agendamento. E uma área comum onde pegara os dados para listar.