# 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.

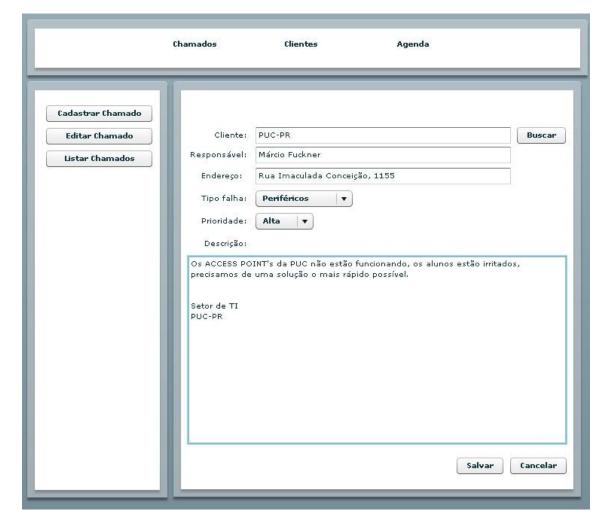
#### CADASTRAR CHAMADO

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:



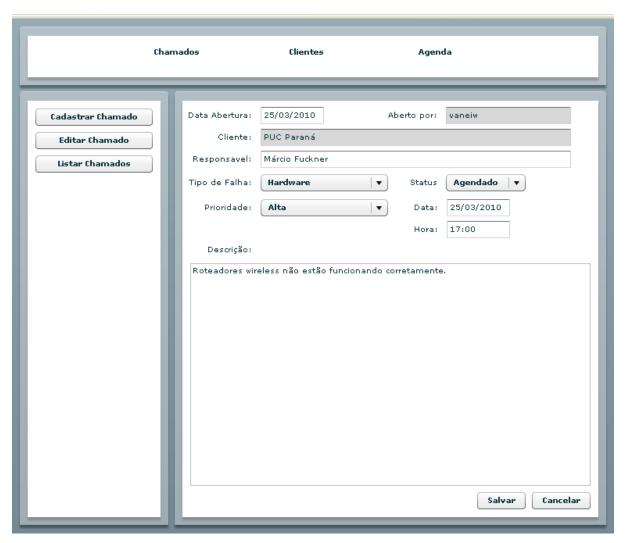
D)Informações técnicas: Ao cadastrar um chamado, o mesmo será encaminhado para um repositório comum de objetos remotos, para que seja listado na funcionalidade LISTAR CHAMADOS.

#### 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:



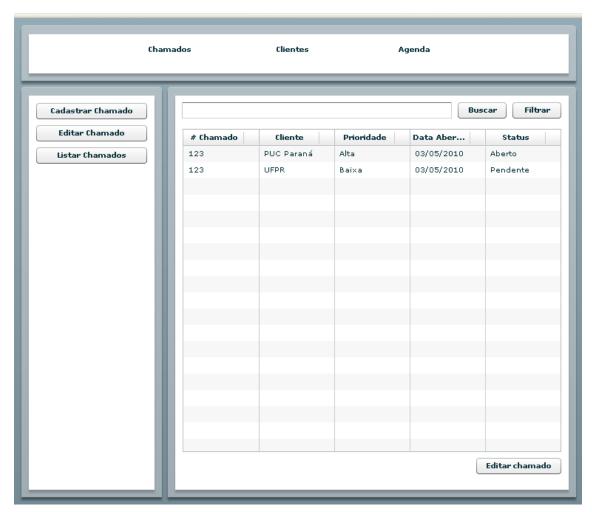
D)Informações técnicas: Ao editar um chamado será encaminhado para um repositório comum de objetos remotos, para que seja listado na funcionalidade LISTAR CHAMADOS. Caso o técnico determine uma data e hora, devera implementar um Observable para avisar a LISTAR AGENDA, que foi alterado/inserido um agendamento.

#### LISTAR CHAMADO

A)Funcionalidade: Esta tela demonstra a lista de chamados dos sistema. Nela podemos aplicar um filtro, clicando no botão "Filtrar", que abrirá uma modal para que o usuário selecione os parâmetros desejados. Nesta mesma janela, podemos localizar um chamado através do campo de busca. Quando esta tela estiver aberta, e a aplicação for minimizada, um "tray icon" irá aparecer no conto inferior da tela. Este mesmo "tray icon" irá notificar novos chamados, caso estes ocorram.

B) Autor: André N. Schuster

#### C)Protótipo da funcionalidade:



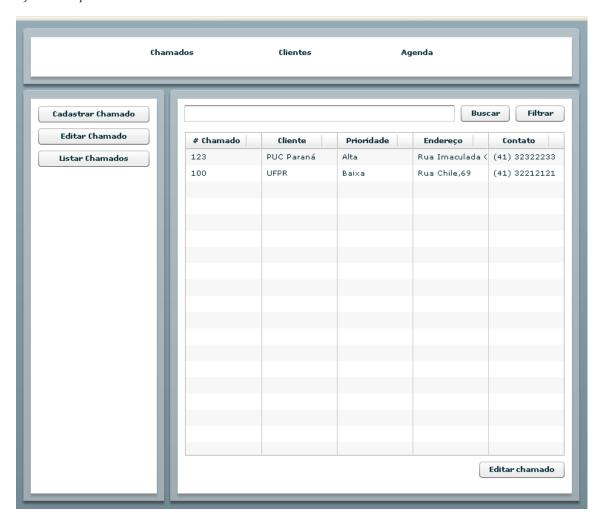
**D)Informações técnicas:** Utilizaremos RMI para troca de objetos com o service que irá conter os chamados. Para o "auto-refresh" da lista, utilizaremos uma thread que irá realizar consultas ao service periodicamente. Para a notificação de mensagens do "tray icon" utilizaremos o padrão observer/observable.

## LISTAR AGENDA

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.

B) Autor: Vanessa Iwamoto

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.