

Este documento é um resumo do trabalho desenvolvido em conjunto pelas fábricas que compõem a turma do Mestrado de Engenharia de Software 2014.2 do C.E.S.A.R Edu. As fábricas em questão são: **Fábrica M8**, **Fun Factory**, **F2** e **SwCraft**.

Os detalhes da solução empregada, bem como links úteis para download de códigos-fonte e vídeos de apresentação encontram-se nas seções seguintes.

1. Diagrama com a arquitetura do sistema

A figura abaixo representa o diagrama de arquitetura do sistema. Os diversos componentes da solução implementam uma interface única: *IServicoChamado*. No diagrama é possível observar o servlet de chamada principal (*ServicoChamadoServlet*), que é responsável em prover as operações REST da aplicação para criação e manutenção dos eventos, e os Servlets que retornam mapas (*Google Maps* e *Open Street*).

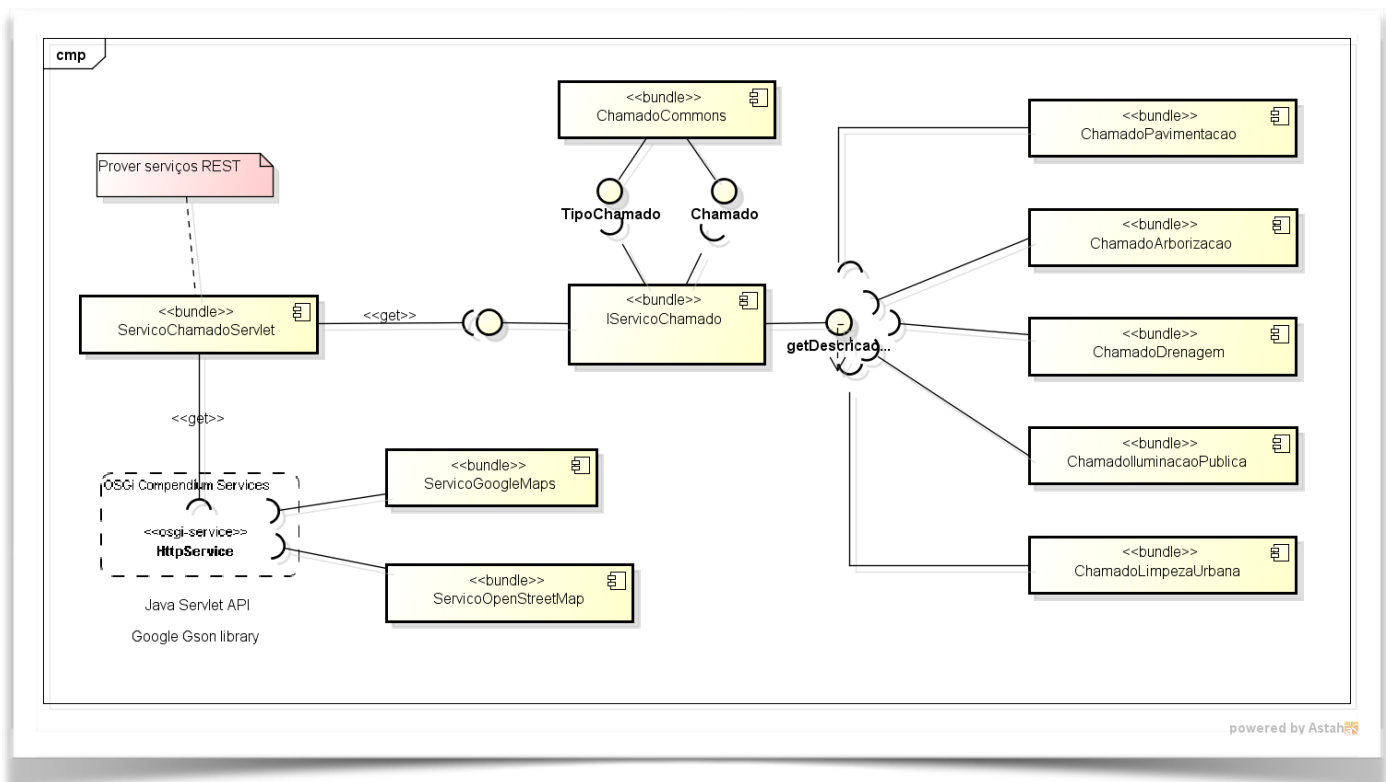


Figura 1: Diagrama de Arquitetura da Solução

2. Código-fonte

Anexado a este documento, segue arquivo ZIP contendo a plataforma OSGi já configurada e pronta para rodar com todos os componentes necessários para o projeto. Neste arquivo estão incluídos os projetos:

- Bundles de tipos de chamados
 - ChamadoArborizacao (SwCraft)
 - ChamadoDrenagem (SwCraft)
 - ChamadoIluminacaoPublica (Fun Factory)
 - ChamadoLimpezaUrbana (Fábrica M8)
 - ChamadoPavimentacao (F2)

- Servlets OSGi
 - ServicoChamadoServlet (Fun Factory)
 - ServicoMapas (F2)
 - ServicoOpenStreetMap (F2)
- libs (bibliotecas necessárias para executar os servlets)
- Aplicação Web
 - Solução em Grails para consumir serviços dos Componentes (Fábrica M8)

Para integração dos componentes dos tipos de chamados e a interface web, serviços REST foram criados a partir dos Servlets OSGi, são eles:

- **GET**
http://localhost:8080/chamados/tiposchamados (application/json)
- **POST**
http://localhost:8080/chamados (application/x-www-form-urlencoded)
 - *headers* (Content-Type:application/x-www-form-urlencoded)
 - *body* (tipo_chamado=PP&descricao=buraco+na+minha+rua&latitude=-9.6463367&longitude=-35.7352281)
- **PUT**
http://localhost:8080/chamados (application/x-www-form-urlencoded)
 - *headers* (Content-Type:application/x-www-form-urlencoded)
 - *body* (id=1430794777158&status=P)
- **GET**
http://localhost:8080/chamados/lista (application/json)
- **GET**
http://localhost:8080/chamados/detalhes?id=1430793960316 (application/json)

Os códigos-fonte (também) podem ser acessados nos repositórios (no *GitHub*) abaixo.

- <https://github.com/aerciofilho/reuso-osgi>
- <https://github.com/brunobarros/mpes-reuso-basico>
- <https://github.com/pcabrantess/mpes-reuso-basico-f2>

3. Instruções de Execução

3.1 Bundles de Tipos de Chamados e Servlets OSGi

Abaixo estão listados os passos necessários para executar o sistema:

- Importar projetos no Eclipse
- Importar o Launch configuration na raiz do repositório: Servico Chamado Servlet (OSGi).launch (este contém todos os componentes necessários para rodar o projeto)
- Run do framework OSGi com os bundles do projeto
- executar stop e start do bundle de servlet para deixá-lo por último na prioridade dos bundles e que os tipos de chamados fiquem disponíveis
 - lb (listar os bundles)
 - stop [id do bundle ServicoChamadoServlet]
 - start [id do bundle ServicoChamadoServlet]
- Acessar <http://localhost:8080/chamados>

3.2 Interface Web

Passos para executar a interface web:

- Fazer o download do pacote de instalação smartCity.war, disponível em: <https://github.com/aerciofilho/reuso-osgi/raw/master/download/smartCity.war>
- Fazer o deploy do smartCity.war em um container web (Tomcat ou Jetty)
 - O container deve rodar em uma porta diferente da 8080 (visto que ela é utilizada pelos Servlets)
- Acessar <http://localhost:<NOVA-PORTA>/smartCity>

4. Vídeo de Demonstração

Como requisito do projeto, um vídeo foi produzido para demonstrar as funcionalidades dos componentes desenvolvidos pelas fábricas ilustrando as funcionalidades do sistema, bem como a maneira como o sistema lida com dinamismo, para isto, alguns bundles de tipos de chamados são parados e é demonstrado a indisponibilidade do tipo de chamado na interface web.

O link abaixo deve ser acessado para visualizar o vídeo de demonstração:

Link: <http://screencast.com/t/wSob2C2Mlc>

Funcionamento do Sistema

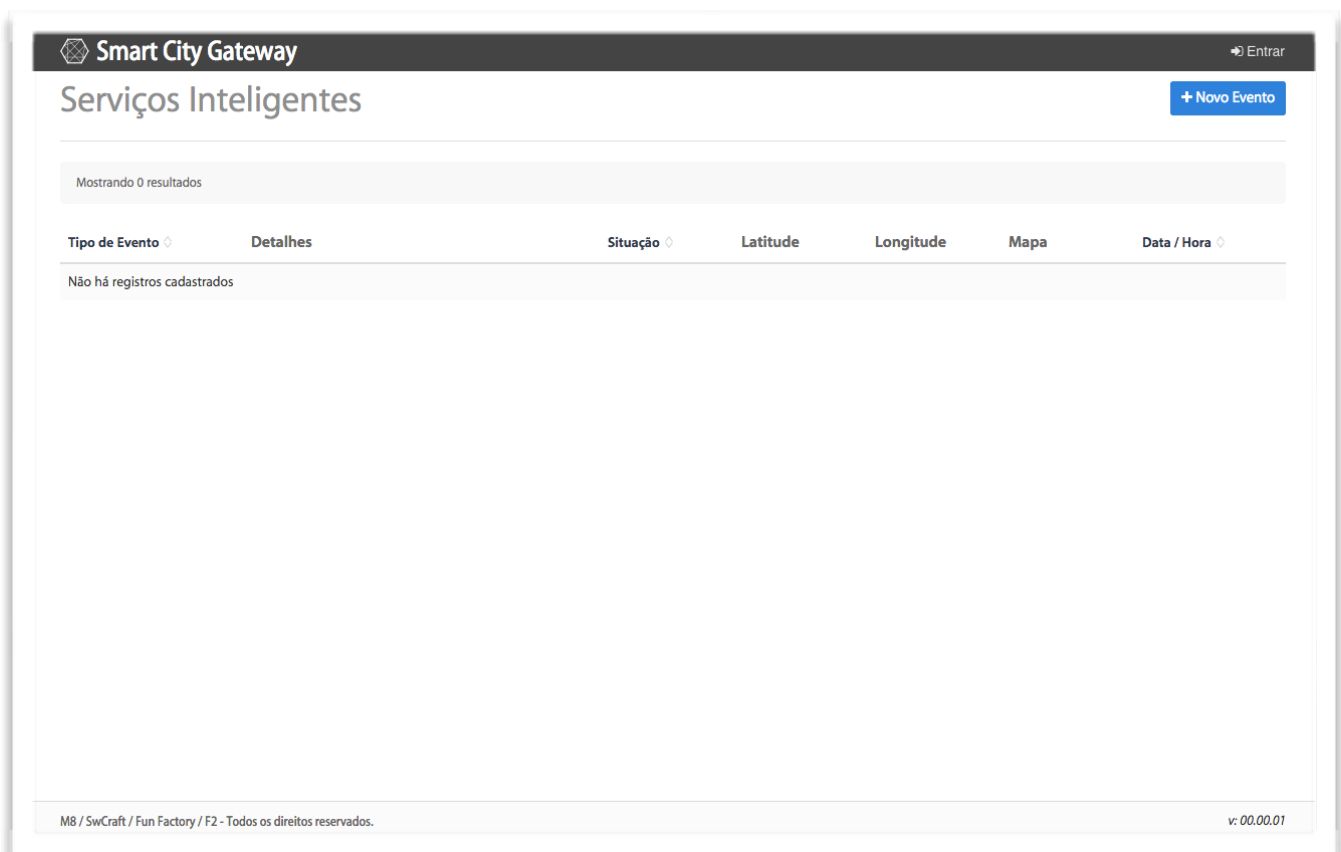


Figura 2: Tela de listagem de eventos (vazia)

Smart City Gateway

Entrar

Criar Evento

Tipo de Evento *

Iluminação Pública (poste com lâmpada queimada)

Latitude *

Longitude *

Descrição do Evento

[Cancelar](#) [Confirmar](#)

M8 / SwCraft / Fun Factory / F2 - Todos os direitos reservados. v: 00.00.01

Figura 3: Tela de criação de eventos

Smart City Gateway

Entrar

Serviços Inteligentes

Evento criado com sucesso ✕

+ Novo Evento

Mostrando 1-1 de 1 resultados

Tipo de Evento	Detalhes	Situação	Latitude	Longitude	Mapa	Data / Hora
Drenagem (desentupimento de galeria)	Esgoto a céu aberto na rua X	Em Aberto	-8.062733634	-34.8679344	ver	13/05/2015 08:56 AM

M8 / SwCraft / Fun Factory / F2 - Todos os direitos reservados. v: 00.00.01

Figura 4: Tela de listagem de eventos (com eventos salvo)

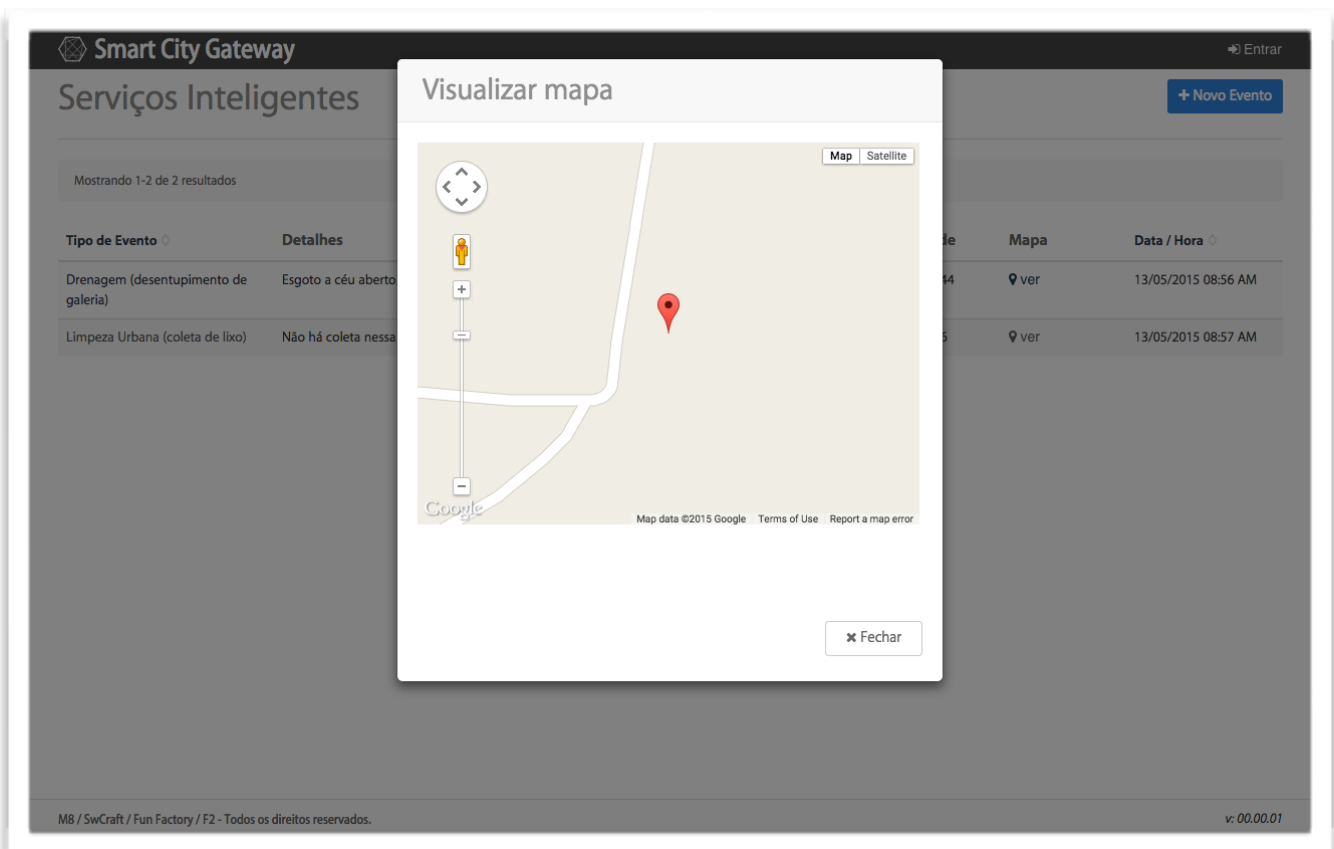


Figura 5: Visualização do mapa (Google Maps)

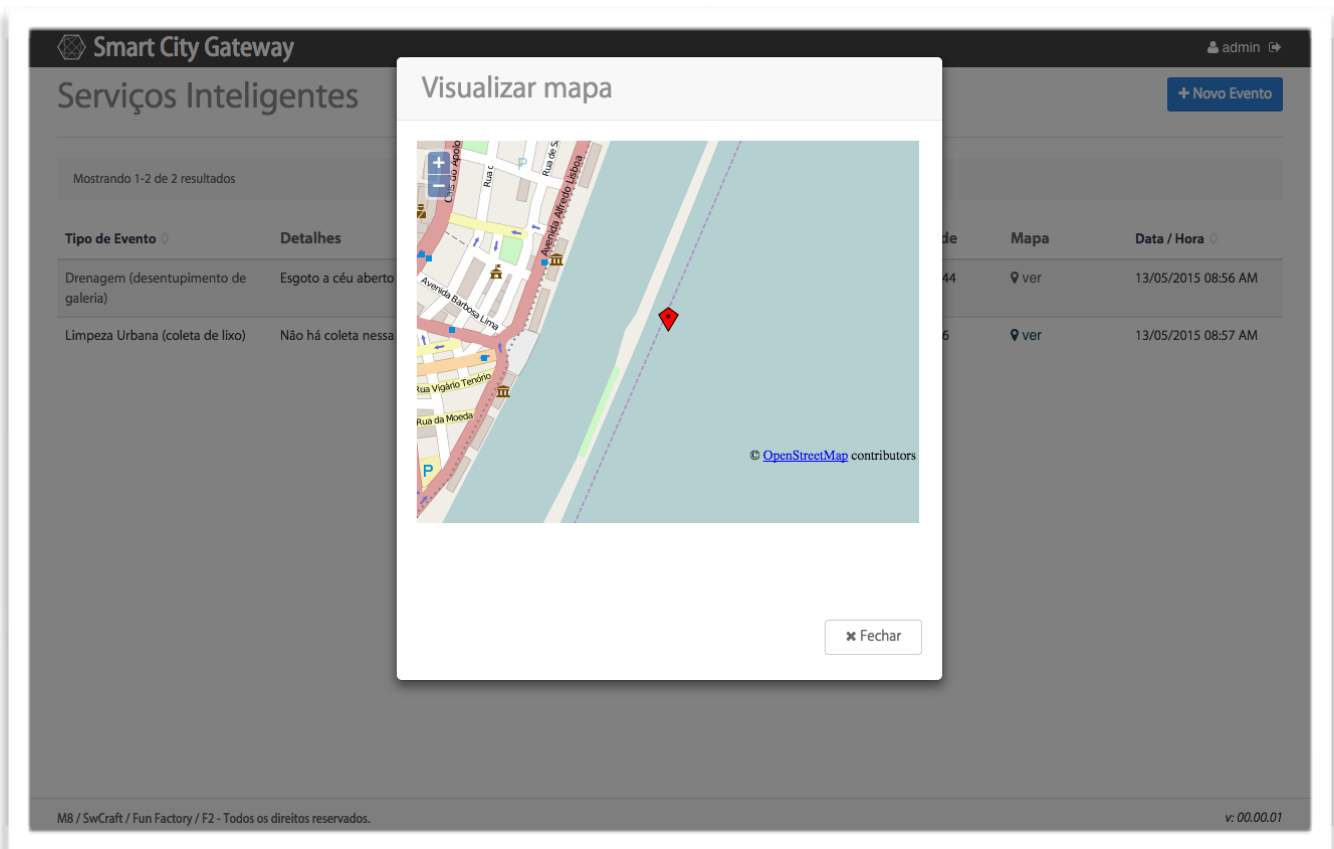


Figura 6: Visualização do mapa (Open Street)

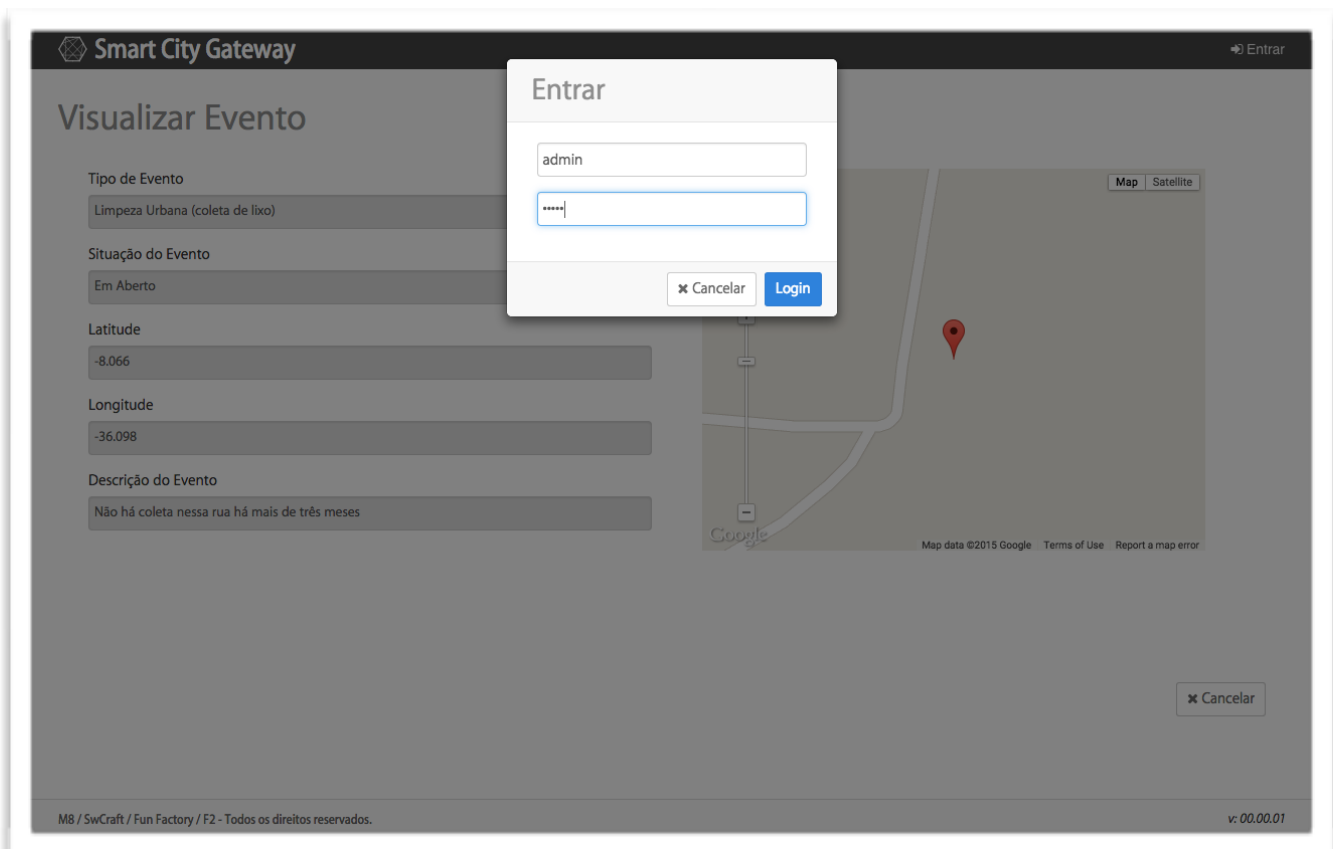


Figura 7: Tela de login (Spring Security)

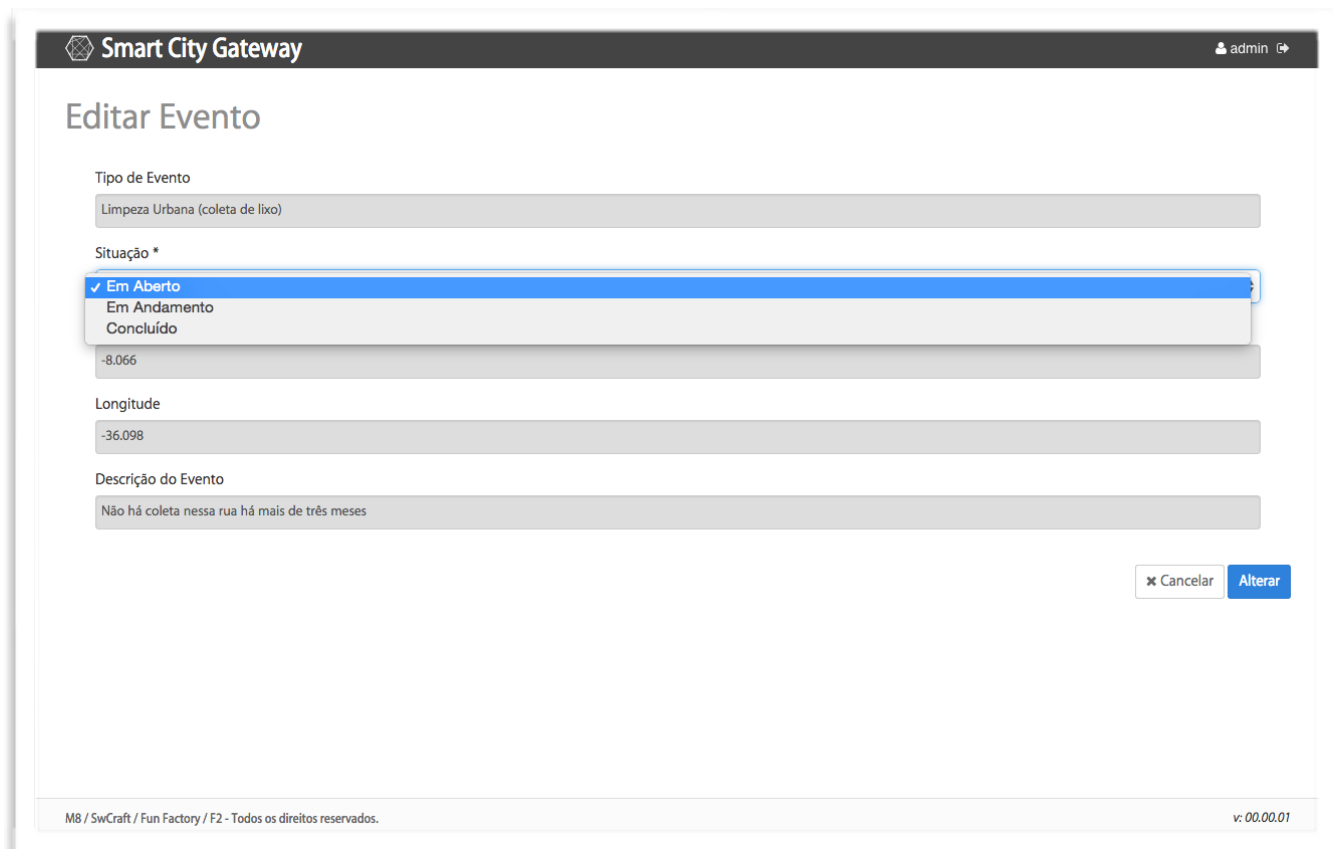


Figura 8: Edição de evento (trocar situação - apenas admin)