**Documento de Requisitos para o trabalho da disciplina**

**Software for Smartphone and Cloud Computer**

**Aluno: David Estevam de Britto Jr**

**Disciplina: EEL 970 – Software for SmartPhones and Cloud Computing**

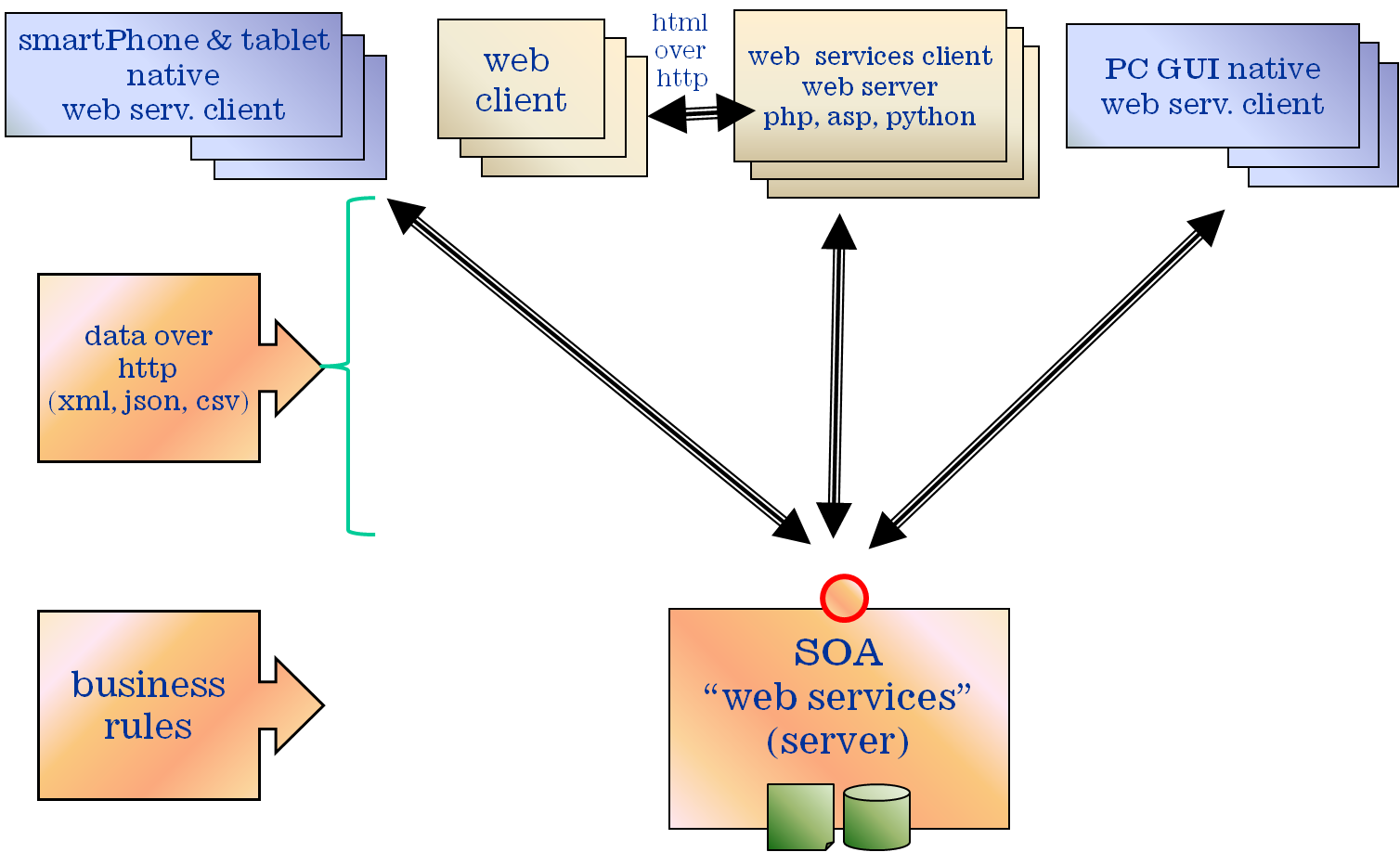
**UFRJ – DEL -Departamento de Engenharia Eletrônica e de Computação**

**Professor: Sergio Barbosa Villas-Boas (sbVB)** [**www.sbvb.com.br**](http://www.sbvb.com.br/)

# **Titulo: IndyCrawler – Indiana Jones, the Museum Crawler (Museum/Gallery Crawler)**

**1. Introdução**

Esse é um sistema de software utilizando a arquitetura SOA\_MC, que busca dados de diversos espaços de eventos culturais (museus e galerias de arte), e apresenta tais eventos para o usuário de forma concisa em um aplicativo mobile.



**1.1 Propósito**

Como usuário, gostaria de poder consultar rapidamente eventos e atividades que posso realizar no Rio de Janeiro, recebendo um título, descrição, data/hora, local e preço de tal evento.

**1.2 Escopo**

Há 2 módulos no sistema, como listado abaixo.

1. Servidor na cloud, que reúne os dados de eventos culturais em um banco MySQL, e alimenta um aplicativo mobile por meio de web service.
2. Aplicativo mobile, que recebe os dados da cloud e exibe para o usuário.

**1.3 Definições**

* Evento: atividade qualquer que é realizada em um museu e/ou galeria, pode ser uma exposição, workshop, visitação, etc. Deve possuir um título, data de inicio e fim, e localização associados. Além disso, pode possuir uma descrição e preço do evento.
* Localização: museu/galeria/teatro/cinema/espaço cultural onde ocorre um evento. Deve possuir um nome e um endereço associado.

**2. Descrição Geral**

**2.1 Descrição:**

* Inicialização do sistema:
  + O usuário abre o aplicativo mobile.
  + O aplicativo mobile acessa o servidor, e busca todos os eventos disponiveis, começando na data atual, ordenado por data e título.
* Consulta a eventos por data:
  + O usuário pressiona o botão “Busca por data” no aplicativo mobile.
  + O usuário escolhe uma data de inicio.
  + O aplicativo exibe para o usuário uma lista de eventos que atende a especificação de data desejada, começando na data de início, ordenado por data e título.
* Consulta a eventos por localização:
  + O usuário pressiona o botão “Busca por localidade” no aplicativo mobile.
  + O usuário escolhe de uma lista qual localidade ele deseja. É possível selecionar mais de uma localidade.
  + O aplicativo exibe para o usuário uma lista de eventos que iram ocorrer apenas nas localidades escolhidas, começando na data atual, ordenado por data e título.
* Informações detalhadas de um Evento:
  + O usuário seleciona um Evento na lista exibida pelo aplicativo.
  + O aplicativo acessa a cloud para receber a descrição completa do Evento.
  + O aplicativo exibe uma tela com todos os dados do Evento.
* Informações detalhadas de uma Localização:
  + O usuário seleciona no menu a opção “Localizações”
  + O aplicativo exibe uma lista de localizações possíveis.
  + O usuário seleciona uma localização.
  + O aplicativo exibe uma tela com as informações detalhadas da localização escolhida.

**2.2 Tecnologias utilizadas**

As seguintes tecnologias serão utilizadas no projeto:

* Android (Módulo mobile)
* Apache Tomcat (Servidor Java)
* Axis 2 (Web Services)
* Crawler4j (Web Crawler)
* Jsoup (parser HTML)
* MySQL (banco de dados)
* Jaxb (Geração de Strings XML)

**3. Requisitos**

**3.1 Requisitos funcionais (**Casos de uso)

Módulo 1 (Servidor na cloud e banco de dados):

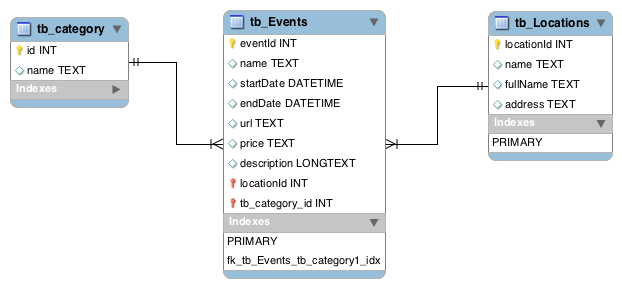
* Acessa bases de dado do CCBB, MAM, MAR, MAC e filtra os futuros eventos.
* Guarda os dados dos eventos em um banco de dados.
* Oferece uma interface para um aplicativo mobile acessar esses dados, por meio de Web Services.
* O aplicativo pode realizar quatro requisições:
  + Requisição de informação das localidades.
  + Requisição de todos os eventos disponíveis
  + Requisição de todos os eventos a partir de uma determinada data.
  + Requisição de todos os eventos em uma localização.

Módulo 2 (Aplicativo Mobile):

* Exibe uma lista de eventos futuros.
* Filtra a lista de eventos de acordo com data e localização.
* Exibe dados detalhados de um evento.
* Exibe dados detalhados de uma localização.

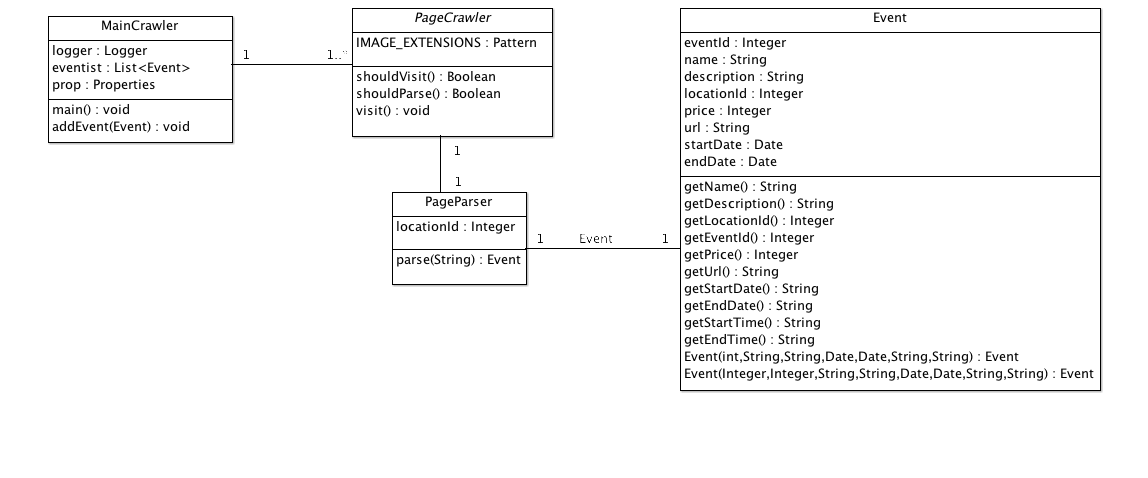
## 4. Diagramas

## 4.1 Modelo do Banco de Dados:

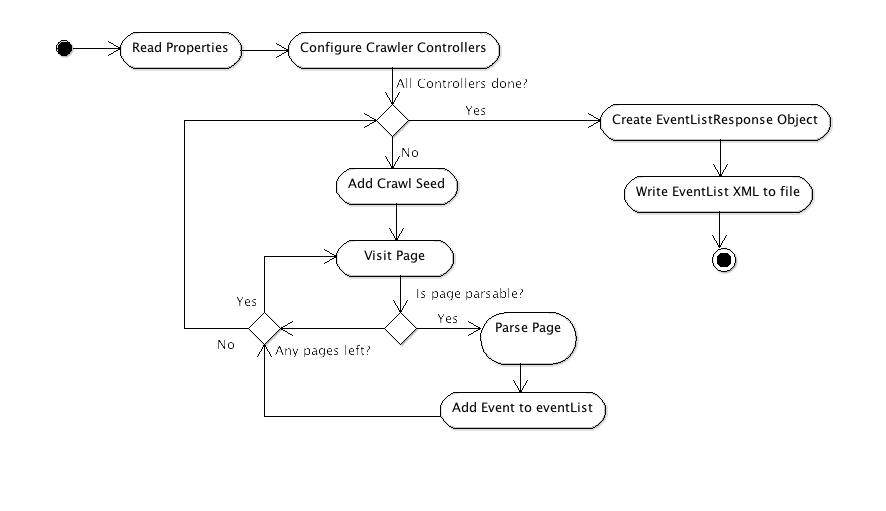


## 4.2 Modelo do Crawler:

### 4.2.1 Diagrama de Classes:



### 4.2.2 Diagrama de Atividades:

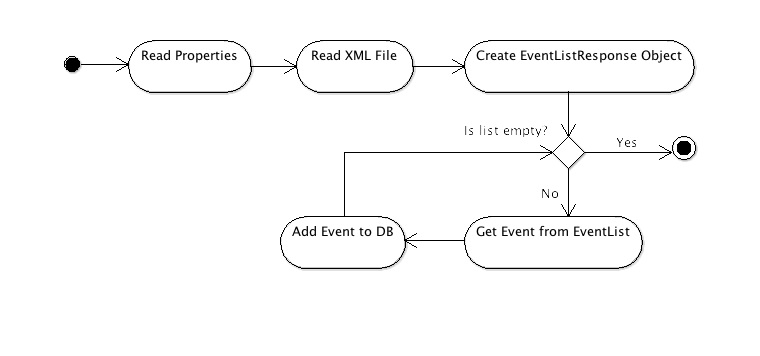


## 4.3 Modelo do Loader:

### 4.3.1 Diagrama de Classes:

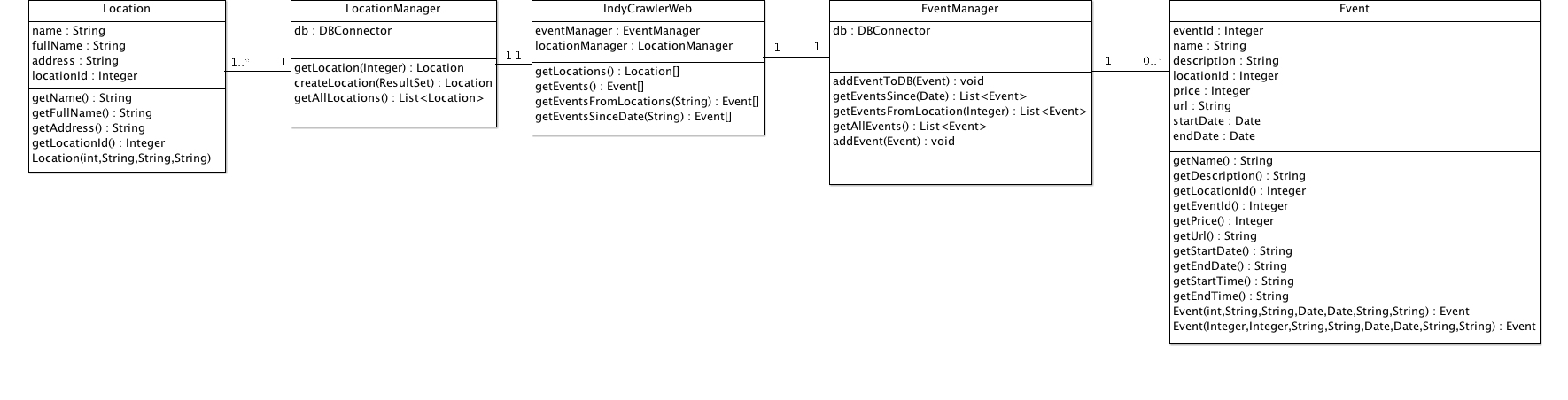


### 4.3.2 Diagrama de Atividades:



## 4.3 Web Services:

### 4.3.1 Diagrama de Classes:



## 5. TODO:

Desacoplar os módulos

1. crowler para arquivo (IndyCrawler, que depende do IndiCrawlerUtils)
2. injetor: de arquivo para banco de dados (na nuvem ou não)
3. server na nuvem (web-services). Estudar forma de retornar array de objetos (não usar contenedor no retorno), para de fato usar o axis2.
4. Android – verificar dependencias (infelizmente não conseguimos em sala :( )