

Um Relato de Experiência sobre a Integração do Primefaces com Google Maps API

Pablo Sebastian Veinberg, Fausto Vêras Maranhão Ayres

Instituto Federal de Educação, Ciência e Tecnologia da Paraíba – IFPB

Caixa Postal 720 – 58.015-430 – João Pessoa – PB – Brasil

pveinberg@gmail.com, fausto.ayres@ifpb.edu.br

Abstract. *This paper describes an experience of the integrated use of Primefaces and Google Maps technologies in the context of the development of a Web Geographic Information System to the biodiversity area. It is expected that this work will contribute to the development of systems that require interaction with web maps.*

Resumo. *Este trabalho relata uma experiência do uso integrado das tecnologias Primefaces e Google Maps API no contexto do desenvolvimento de um Sistema de Informação Geográfica na Web para a área de biodiversidade. Espera-se que este trabalho possa contribuir no desenvolvimento de sistemas que requerem a interação com mapas na web.*

1. Introdução

Atualmente, muitas aplicações web necessitam interagir com seus usuários através de adição e visualização de marcadores em mapas – em particular, usando a tecnologia de criação de mapas Google Maps. Para tanto, é necessário que tal aplicação invoque comandos da Google Maps API. Uma abordagem é utilizar a linguagem Javascript, para adicionar os marcadores no mapa, a partir de dados geográficos (latitude e longitude) pré-determinados ou lidos do usuário. No entanto, esta abordagem não oferece facilidade para a transferência de dados do mapa para um banco de dados e vice-versa. No contexto da plataforma Java/JSF, existem APIs que facilitam esta transferência, como [GMaps4JSF 2013] e Google Maps API do Primefaces (2013).

Este artigo apresenta um relato de experiência baseado no uso das tecnologias Primefaces e Google Maps API. Tal integração é feita no contexto do desenvolvimento de um Sistema de Informação Geográfica na Web [Veinberg 2013], o qual permite aos usuários interagirem com mapas para adicionar e visualizar informações geográficas sobre espécies da fauna ameaçada de extinção.

2. Integração entre Primefaces e Google Maps API

A integração apresentada neste trabalho trata de duas operações de adição de marcadores num mapa que está inserido dentro de um documento *xhtml*, através da *tag* `<p:googleMap/>` do Primefaces.

Na primeira operação, o usuário pode adicionar um ou mais marcadores no mapa, clicando com o mouse em qualquer ponto deste e digitando uma observação sobre cada marcador. Além disso, ao clicar no botão de submissão do formulário, associado ao documento, os valores de latitude, longitude e observação dos marcadores adicionados ao mapa são capturados e enviados para um objeto *backingbean* que, por

sua vez, os persistirá num banco de dados. Na segunda operação, os marcadores são adicionados ao mapa a partir de dados provenientes do banco de dados.

3. Materiais e Métodos

O sistema web [Veinberg 2013], utilizado como base para este artigo, foi desenvolvido em Java/JSF com uma arquitetura em camadas usando o padrão MVC (*Model, View, Controller*). É na camada de visão (*View*) que é feita a integração do Primefaces com a API de mapas do Google, dentro de alguns documentos web (*xhtml*). A camada de modelo (*Model*) agrupa as classes de negócio do sistema, as quais são mapeadas para um banco de dados relacional com o padrão JPA (*Java Persistence API*). A camada de controle é o elo que conecta as duas camadas, através das classes *backingbeans*, as quais obtêm informações do banco de dados para apresentar na camada de visão e vice-versa. O sistema se comunica com o banco de dados através de uma camada DAO (*Data Access Object*).

Esse sistema possui várias funcionalidades, mas neste experimento, serão descritas apenas as funcionalidades de registro e consulta de ocorrências de espécies em extinção, cujas telas são mostradas nas Figuras 1(a) e 1(b), respectivamente. Num registro de ocorrência de espécie, um usuário autenticado adiciona no mapa um ou mais marcadores para a espécie selecionada, cujos dados (latitude e longitude) serão enviados para o banco de dados na submissão do formulário web. No caso de consulta de registros de ocorrências, o usuário visualiza no mapa os marcadores de todas as ocorrências da espécie selecionada, obtidos a partir dos dados inseridos no banco de dados.

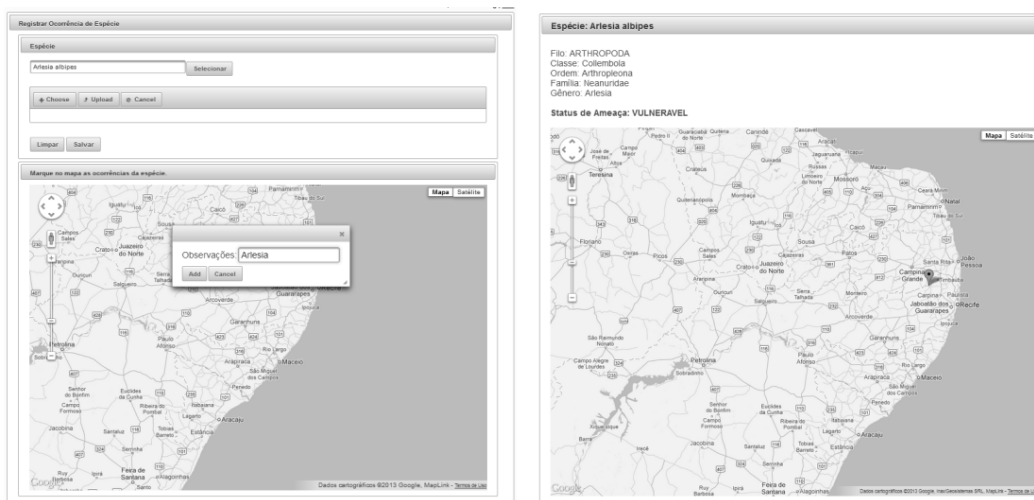


Figura 1 - Telas de registro (a) e de consulta (b) de ocorrências de espécie

4. Resultados

Na Figura 2 é mostrado parte do documento da tela de entrada de dados (Figura 1(a)), onde a tag `<p:gmap/>` é usada para renderizar o mapa de acordo com as configurações das suas propriedades. Observa-se que existem vários acessos a atributos e métodos do *backingbean* `gmapBean`, responsável pela inserção dos dados dos marcadores no banco de dados.

A cada clique do usuário no mapa, o evento `OnPointClick` é disparado, provocando a chamada da função Javascript `handlePointClick()`, mostrada na Figura 3, que preenche os campos ocultos do formulário (*tag form*), a partir dos dados do marcador oriundo do evento. Esses campos ocultos são capturados pelo `gmapBean` através de uma requisição *ajax*. O método `addMarker()` do `gmapBean` armazena os dados do marcador na variável `emptyModel` para depois persisti-los no banco.

```
<p:gmap id="gmap" center="{gmapBean.LATITUDE}, {gmapBean.LONGITUDE}" zoom="{gmapBean.ZOOM}" type="{gmapBean.TYPE}"
  style="width:916px;height:610px" model="{gmapBean.emptyModel}" onPointClick="handlePointClick(event);"
  widgetVar="map" />
<p:dialog widgetVar="dlg" effect="FADE" effectDuration="0.5" close="false" fixedCenter="true">
  <h:form prependId="false">
    <h:panelGrid columns="2">
      <h:outputLabel for="observacoes" value="Observações:" />
      <p:inputText id="observacoes" value="{gmapBean.observacoes}" />
      <f:facet name="footer">
        <p:commandButton value="Add" update="messages" ajax="false"
          actionListener="{gmapBean.addMarker}"
          onComplete="markerAddComplete()" />
        <p:commandButton value="Cancel" onclick="return cancel()" />
      </f:facet>
    </h:panelGrid>
    <h:inputHidden id="Lat" value="{gmapBean.Lat}" />
    <h:inputHidden id="Lng" value="{gmapBean.Lng}" />
  </h:form>
</p:dialog>
```

Figura 2 - Detalhe do documento de entrada de dados

O atributo `emptyModel` do `gmapBean`, que é um objeto do tipo `DefaultMapModel`, disponibiliza propriedades e métodos para o processamento dos dados dos marcadores. Por último, o marcador é exibido no mapa do documento web através do método `addOverlay()` da variável `map` associada à propriedade `widgetVar` da *tag* `gmap`. Adicionalmente, todas as manipulações de dados do lado cliente são gerenciadas através do código Javascript.

```
var currentMarker = null;
function handlePointClick(event) {
  if (currentMarker == null) {
    document.getElementById('formulario-especie:lat').value = event.latLng.lat();
    document.getElementById('formulario-especie:lng').value = event.latLng.lng();
    currentMarker = new google.maps.Marker({
      position : new google.maps.LatLng(event.latLng.lat(),event.latLng.lng())
    });
    map.addOverlay(currentMarker);
    dlg.show();
  }
}
```

Figura 3 – Programação do evento Click do mapa

A Figura 4 mostra a *tag* `<p:gmap/>` do documento de saída de dados (arquivo `detalhe-registro.xhtml`) referente à Figura 1(b), onde o *backingbean* `gmapBean` é usada para configurar a apresentação do mapa e o *backingbean* `listagemBean` é usado para recuperar os dados do banco de dados e os enviar para a apresentação no mapa, utilizando os métodos `montarDetalhe()` e `adicionarMarcadores()` mostrados na Figura 5.

```
<p:gmap
  center="{gmapBean.LATITUDE}, {gmapBean.LONGITUDE}"
  zoom="{gmapBean.ZOOM}"
  type="{gmapBean.TYPE}" style="width:990px;height:742px"
  model="{listagemBean.simpleModel}" />
</p:panel>
```

Figura 4 - Detalhe da tag map no documento de saída de dados

O método *montarDetalhe()* localiza no banco de dados os registros de ocorrência da espécie consultada, através do método *buscaRegistroEspecie()*, resultando numa lista de coordenadas geográficas e observações dos marcadores que serão adicionadas ao mapa através do método *adicionarMarcadores()*, que por sua vez, instancia cada marcador e o adiciona ao mapa, através do método *addOverlay()*, tornando-os visíveis para o usuário. A variável *simpleModel* representa um objeto do tipo *DefaultMapModel* que funciona como interface de comunicação com o mapa que está sendo exibido.

```
public String montarDetalhe(int id) {
    detalhe = fachada.buscaRegistroEspecie(id);

    data = Helper.getDataFormatada(detalhe.getDataCadastro());
    adicionarMarcadores();
    return ("/public/admin/detalhe-registro.xhtml");
}

private void adicionarMarcadores() {
    simpleModel = new DefaultMapModel();
    for (LocalizacaoRegistroEspecie lre : detalhe.getLocalizacoes()) {
        LatLng coord = new LatLng(Double.parseDouble(lre.getLatitude()), Double.parseDouble(lre.getLongitude()));
        simpleModel.addOverlay(new Marker(coord, lre.getObservacoes()));
    }
}
```

Figura 5 - Detalhe do listagemBean

5. Considerações finais

Este trabalho apresentou um relato de experiência, envolvendo a integração das tecnologias Primefaces e Google Maps API, através do desenvolvimento de um sistema de informação geográfica na Web capaz de reunir e disponibilizar dados geográficos sobre espécies da fauna ameaçada de extinção. A principal contribuição deste trabalho é mostrar os detalhes de como implementar as funcionalidades de adição e visualização de marcadores num mapa do Google Maps®, utilizando tais tecnologias.

Referências

- Gmaps4jsf. (2013) Disponível em <<http://code.google.com/p/gmaps4jsf/>> Acesso em 21 de maio de 2013.
- Primefaces. (2013) Primefaces API. Disponível em <<http://goo.gl/vWbW8>> Acesso em 21 de maio de 2013.
- Veinberg, P., Ayres, F., de Sá, I. C.; Cavalcante, B. B. (2012) Um sistema de informação na web para o georeferenciamento de espécies ameaçadas de extinção. Em: Anais do Congresso Norte Nordeste de Pesquisa e Inovação - CONNEPI 2012.