# Case FIAP – Gulliver – Projeto Traveller

## Problema

Todas as vezes que nosso explorador programa suas viagens, tem que navegar por diversos blogs, sites, companhias aéreas, hotéis, locais, pontos turísticos (principalmente os mais recomendados e imperdíveis). E a dificuldade aumenta quando procura pelos melhores lugares para se hospedar, clima, temperatura, melhor época do ano para conhecer, roteiros gastronômicos para todos os gostos e bolsos, comidas regionais, roteiros de compras (com referências e como chegar) serviços diversos e onde localizá-los, além de indicações de serviços, desde hospitais até lavanderias.

## Versão Piloto

▪ Como chegar;

▪ Dicas úteis;

▪ Onde se hospedar;

▪ Alugue seu carro;

▪ História;

▪ Cultura;

▪ Parques;

▪ Gastronomia;

▪ Vida Noturna e;

▪ Entretenimento.

1. Objetivo da aula
   1. Overview sobre o case da FIAP
   2. Entendimento da arquitetura Client / Server e o Protocolo HTTP
   3. Compreender o Tomcat, Servlet, JSP,Scriptlets (<% %>), expression language (EL) e JSTL

através de um Hands on de uma listagem e inclusão para a funcionalidade de Parques do Case

* 1. Transpor o nosso projeto para uma arquitetura MVC

1. Falar sobre o Case da FIAP
2. Desenhos (1,2 e 3) DrawIO
3. Mostrar o desenho (4)
4. Abrir eclipse
5. Procurar a aba Servers, ou senão Window > Show view > Other> Servers
   1. Configurar o Tomcat
      1. https://tomcat.apache.org/download-90.cgi
6. File > New dynamic Web Project
   1. Project Name = fiap=traveller-parque
7. Entender a estrutura das pastas que foram geradas
8. Arrastar o projeto fiap-traveller-parque para o Tomcat server
9. Criar uma página hello.html dentro de webapp
10. Abrir navegador e testar o resultado e comparar com o desenho
11. Mostrar o desenho (5) e hands-on
    1. Explicar o que é Servlet e Diferença para o Tomcat
12. Criar um pacote br.com.fiap.traveller.parque.servlet
    1. Criar uma classe HelloServlet
       1. Incluir anotação @WebServlet(urlPatterns=”/oi”)
       2. Digitar service e dar um Ctrl + espaço e gera o código automático
       3. Incluir o PrintWriter out = response.getWriter();
       4. out.println e montar um html
       5. incluir um println (“O servlet OiMundo foi chamado”)
    2. Criar Servlet ListaParquesServlet (via opção new/ servlet)
       1. Renomear url
       2. Marcar somente a opção “service”
       3. Apagar sugeira de código
       4. Incluir no código Banco banco = new Banco();
          1. Criar a classe Banco
             1. private List<Parque> listaParques = new Arraylist<>();

criar a classe Parque

criar atributos id e nome e adicionar os Getters and Setters

* + 1. incluir List<Parque> listaParques = banco.getParques();
       1. criar o método getParques na classe Banco
          1. Alterar o return para retornar a lista de parques
    2. Código

Interface gráfica do usuário, Texto, Aplicativo

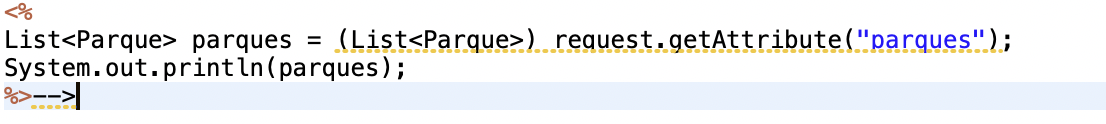
Descrição gerada automaticamente

* + 1. Testar e mostrar a visão inspeção de pagina pra mostrar o html com a tag <ul>
    2. Voltar a classe Banco e incluir um static para simular um banco com informações existentes e depois testar novamente

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

1. Mostrar o desenho (6)
2. Dividir responsabilidades e criar pagina formListaParques.jsp
   1. Apagar linhas q já tem no código html
   2. Explicar que precisamos fazer a servlet chamar o nosso JSP request.getRequestDispatcher("/listaParques.jsp");
   3. Agora precisamos pedir a Servlet levar a resposta adiante, para isso rd.forward(request, response);
   4. Observar que nosso html não esta entendendo o código java dentro dele.
   5. Na ListaParquesServlet, mostrar o request.setAttribute("parques", listaParques);
   6. Agora no listaParques.jsp, mostrar o que é Scriptlets



* + 1. Na classe Banco, incluir o toString para testar a servlet novamente
    2. Incluir scriptlet para fazer o foreach e testar novamente

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

* + 1. JSTL/Taglib: Explicar o que é (<http://www.dsc.ufcg.edu.br/~jacques/cursos/daca/html/jstl/jstl.htm>)
       1. Incluir a tag <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
       2. Dar erro e ir no site baixar a lib
       3. Ajustar código jsp para o abaixo e testar

Uma imagem contendo Texto

Descrição gerada automaticamente

* 1. Incluir novo Parque
     1. Criar um servlet incluir NovoParqueServlet
        1. Codificar igual abaixo

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

* + - 1. Alterar o método para doPost e testar novamente e explicar
      2. Criar jsp novoParqueCriado.jsp

Linha do tempo

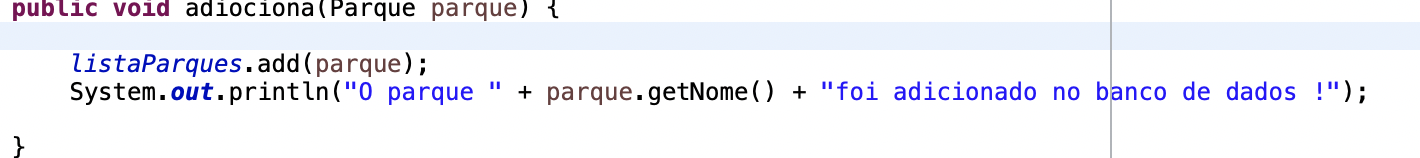
Descrição gerada automaticamente com confiança baixa

* + - 1. Fazer cadastrar este parque em nosso banco em memória
         1. No Servlet NovoParqueServlet, incluir a linha

Texto

Descrição gerada automaticamente

* + - * 1. Dentro da classe banco, implementar o método adicionar



* + - * 1. Agora ir na pagina <http://localhost:8080/fiap-traveller-parque/listaparques> e verificar os itens
        2. Ir na pagina <http://localhost:8080/fiap-traveller-parque/formNovoParque.jsp> e adicionar um novo parque e depois checar novamente na pagina de listaparques para ver se o item foi adicionado.
  1. Mostrar desenho (7)
  2. Fazer o excluir
     1. No jsp listaParques.jsp, incluir o <a href=*"/fiap-traveller-parque/removeParque?id=*${parque.id}*"*>Excluir </a>
     2. Incluir na classe banco um atributo **private** **static** Integer *chaveSequencial* = 1; e incluir o valor chaveSequencial++ nos setIds.
     3. Criar uma Servlet RemoveParque
        1. Interface gráfica do usuário, Texto, Aplicativo, Email

           Descrição gerada automaticamente
     4. No método removeParque ajustar para Iterator

Texto, Carta

Descrição gerada automaticamente

* 1. Fazer o Alterar
     1. \*\*\*\* Criar uma Servlet chamada AlteraParqueServlet (doPost) com o código abaixo (copiar do NovoParqueServlet) e mostrar que precisamos do ID para chave

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

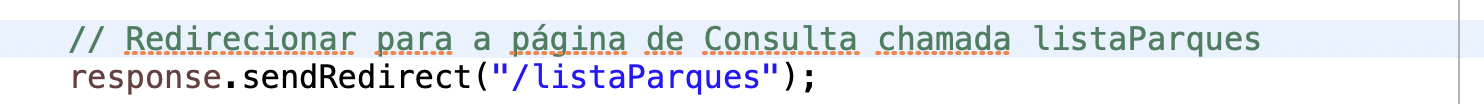
* + 1. Alterar pagina listaParques.jsp
       1. Incluir <a href=*"/fiap-traveller-parque/atualizaParque?id=*${parque.id}*"*>Alterar </a>
    2. Alterar o código da Servlet AlteraParqueServlet para receber o id e o nome oriundo do JSPInterface gráfica do usuário, Texto, Aplicativo

       Descrição gerada automaticamente
    3. Testar a aplicação para ver se esta chegando o id na Servlet
    4. Incluir código para usar o banco, bater as chaves através do ID e atualizar o registro (lembrando que isso num banco tradicional é utilizado o UPDATE)

Texto

Descrição gerada automaticamente

1. Agora fazer o redirecionamento para o ListaEmpresas



1. Criar Servlet MostraParqueServlet

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

1. Na classe banco, criar o buscaParquePorId

Interface gráfica do usuário, Texto

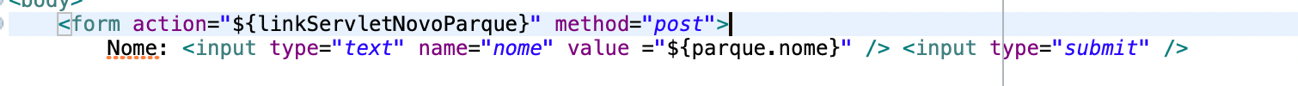
Descrição gerada automaticamente

1. Terminar o código da Servlet MostraParqueServlet

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

1. Criar uma jsp formAlteraParque.jsp e adicionar o value =



1. Alterar o formAlteraParque.jsp para <input type=*"hidden"* name=*"id"* value=*"*${parque.id}*"*/>
2. MVC
   * + 1. br.com.fiap.traveller.parque.controller
       2. br.com.fiap.traveller.parque.model
       3. criar pastinha /view dentro de WEB-INF e adicionar as JSPs la.