

Chamando um serviço XSJS por uma aplicação SAP UI5

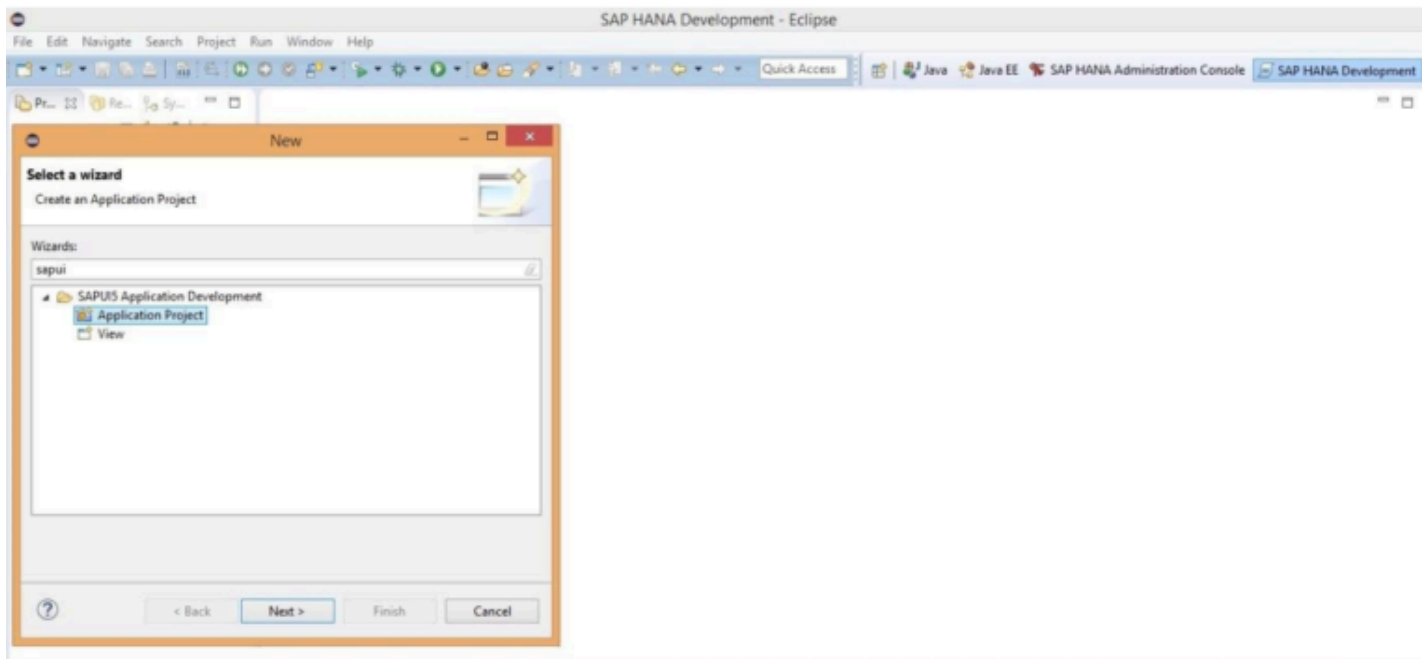
13/04/2017 13:38

Antes de mais nada, vamos começar explicando um pouco sobre o que consiste um arquivo "xsjs" e o que é uma aplicação "SAP UI5". Um arquivo "xsjs" nada mais é que um Javascript que roda no servidor do Hana, já o "SAP UI5", bom eu poderia detalhar e contar toda a história do SAP "UI5" mas nosso foco no momento não é esse, porém é de suma importância conhecermos as ferramentas, linguagens e bibliotecas que vamos utilizar no desenvolvimento de qualquer aplicação, para quem deseja se aprofundar no assunto recomendo o livro de meu amigo Fabio Pagoti, "UI5 para desenvolvedores SAP/ABAP", mas em curtas palavras podemos definir o "SAP UI5" como sendo uma biblioteca Javascript da SAP para criação de aplicações baseadas em HTML5.

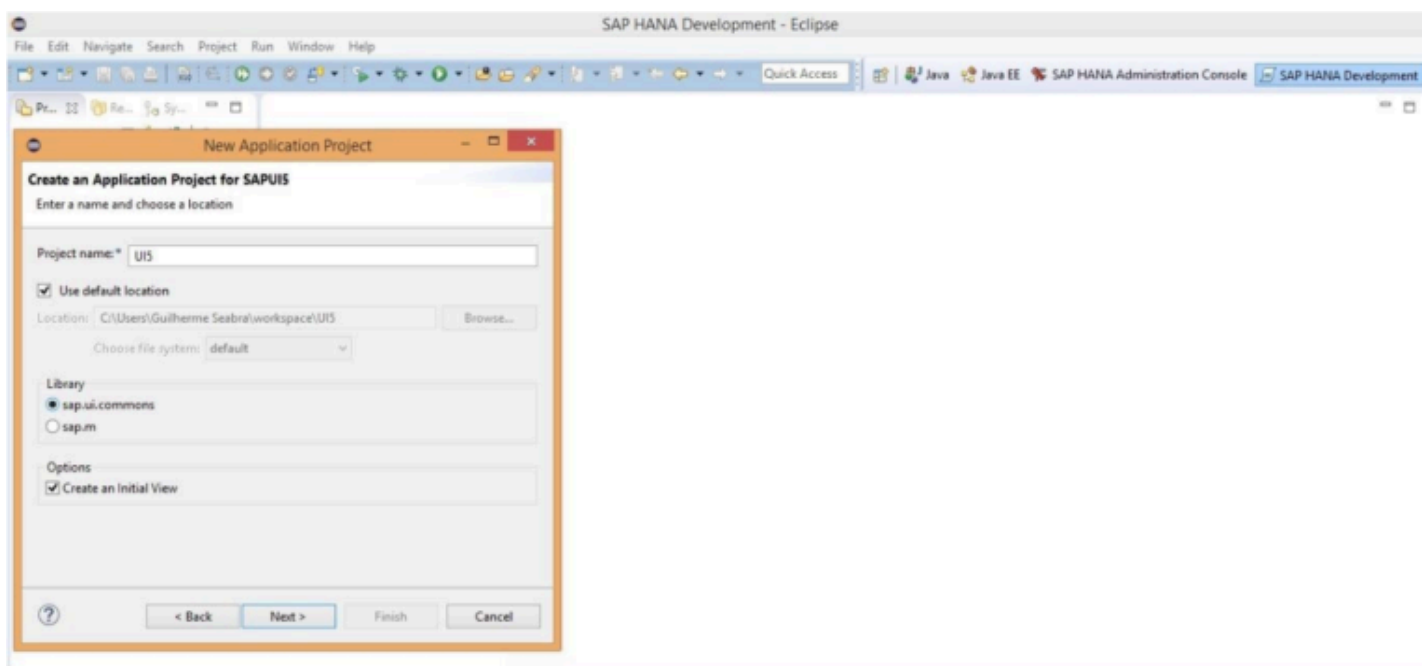
Agora que conhecemos um pouco no que consiste um arquivo "xsjs" e uma aplicação "UI5" vamos criar nossa primeira aplicação SAP UI5 chamando um serviço "xsjs".

Vamos abrir nosso Eclipse (HanaStudio) e clicar em new -> Other.

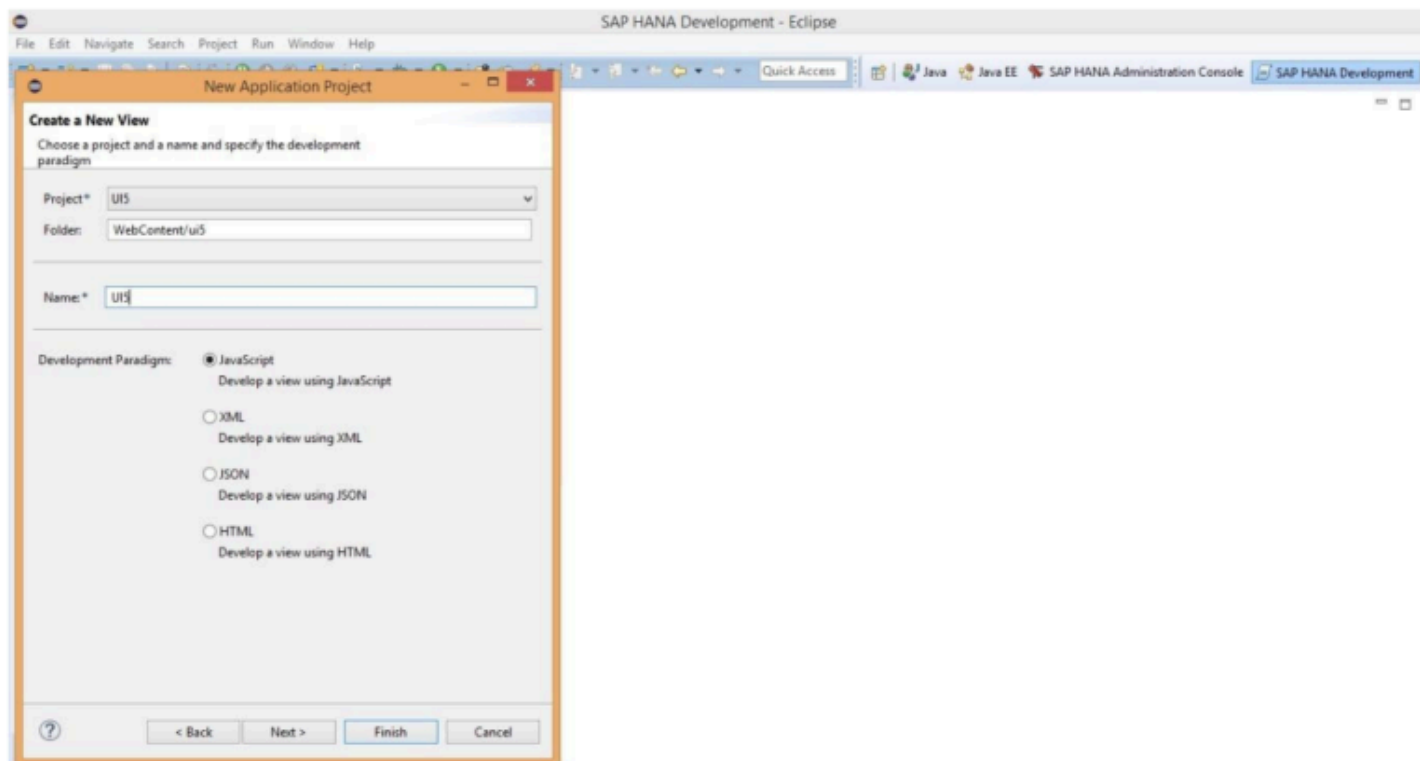
Logo em seguida abra o diretório "SAPUI5" e escolha a opção "Application Project".



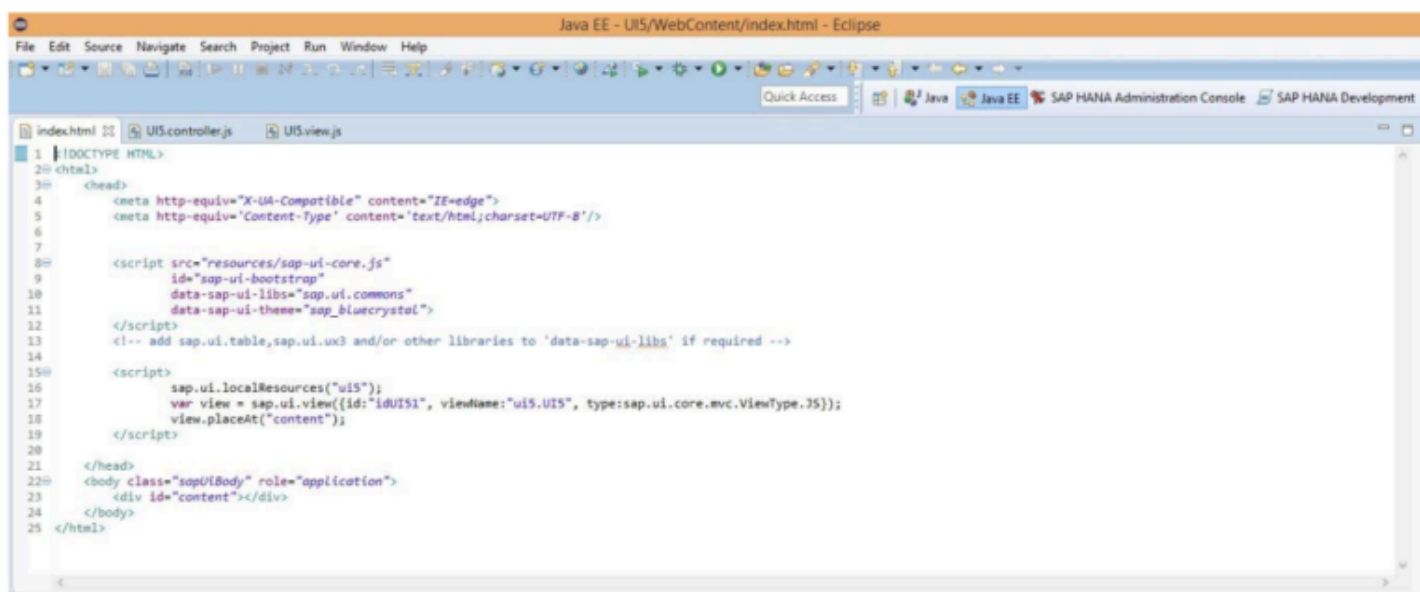
Vamos agora dar um nome em nossa aplicação, reparem que existe um campo que se chama "Library" e dois checkboxes, que são o "sap.ui.commons" que nada mais é que um conjunto de controles de interface gráfica próprios para desktops e o "sap.ui.m" que são um conjunto de controles de interface gráfica próprios para dispositivos móveis, em nossa aplicação vamos escolher o recurso para desenvolvimento desktop e prosseguir clicando em "next".



Bom, agora vamos escolher o projeto e em seguida definir um nome, logo abaixo escolheremos a opção JavaScript, porém é possível de se utilizar também "xml", "JSON" e "HTML", bom fazendo isso vamos finalizar a criação de nosso projeto clicando em "Finish".



Ao se criar o projeto reparem que foram criados três objetos em nosso contexto que são eles: index.html, UI5.controller.js, UI5.view.js. Vamos abrir a opção "index.html".



Vamos agora criar alguns controles em nossa aplicação UI5, vamos utilizar o JQuery, que é um conjunto de interações com curadoria de interface de usuário, efeitos, widgets e temas construídos em cima da biblioteca para isso copie e cole o seguinte trecho de código em seu arquivo "UI5.view.js" conforme mostra a **Listagem 1**.

Listagem 1: Trecho de código para a criação de alguns controles no arquivo "UI5.view.js".

```
sap.ui.jsview("ui5.UI5",
{
  getControllerName : function() {
    return "ui5.UI5";
  },

  createContent : function(oController) {
    var oMatrix = new sap.ui.commons.layout.MatrixLayout({
      layoutFixed: true,
      width: "300px",
```

```

columns:3
});

var oLayout = new sap.ui.commons.form.SimpleForm("formId",{
title:"Primeira Aplicacao em UI5",
content:[
new sap.ui.commons.Label({text:"ID"}),
new sap.ui.commons.TextField("id",{width:"200px"}),

new sap.ui.commons.Label({text:"Primeiro nome"}),
new sap.ui.commons.TextField("p_nome",{width:"200px"}),

new sap.ui.commons.Label({text:"Segundo nome"}),
new sap.ui.commons.TextField("s_sobrenome",{width:"200px"}),

new sap.ui.commons.Label({text:"Idade"}),
new sap.ui.commons.TextField("idade",{width:"200px"}),

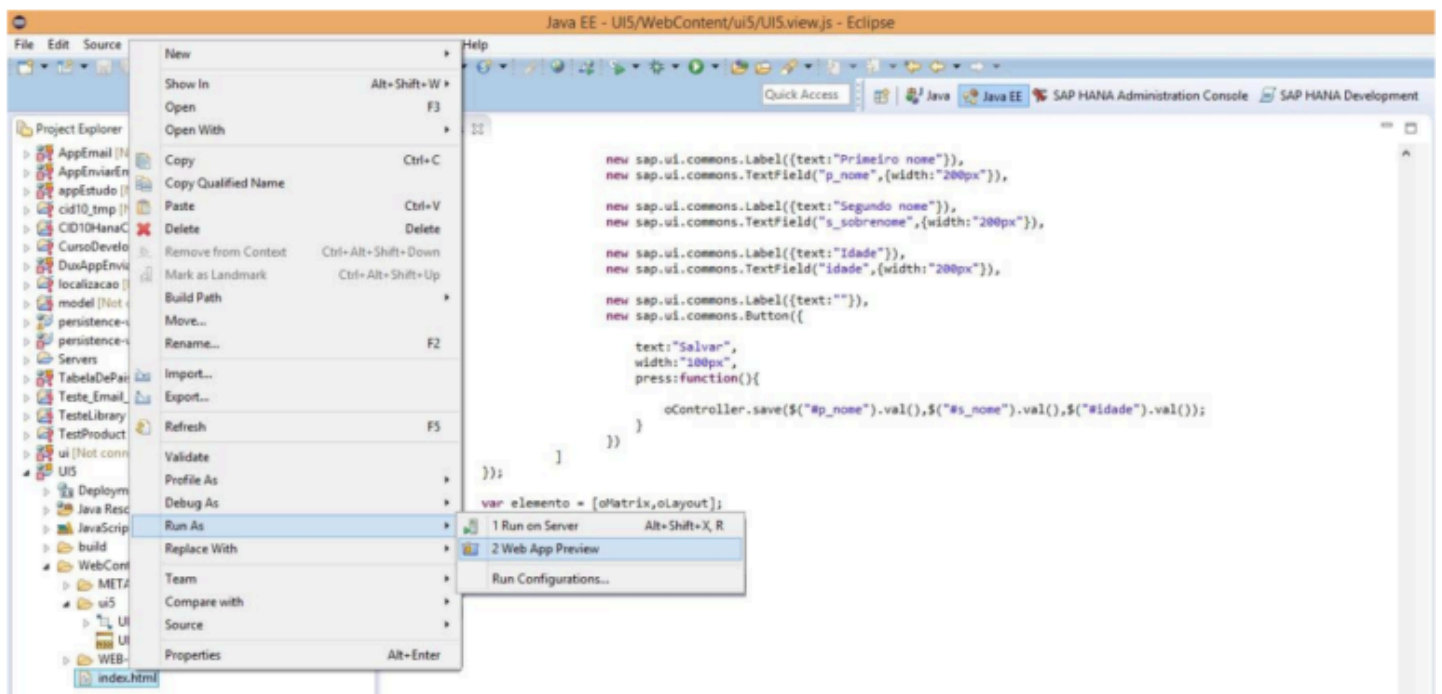
new sap.ui.commons.Label({text:""}),
new sap.ui.commons.Button({

text:"Salvar",
width:"100px",
press:function(){
oController.save(

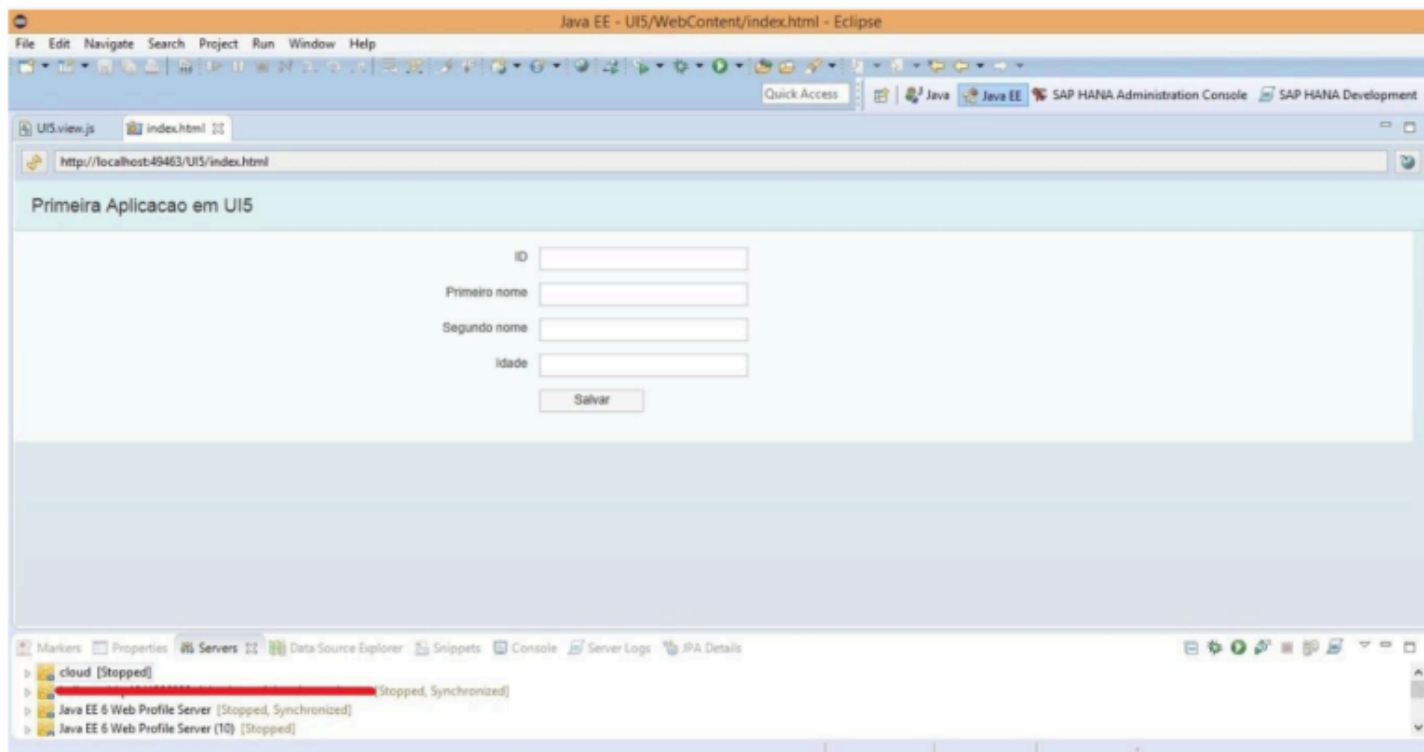
sap.ui.getCore().byId("id").getValue(),
sap.ui.getCore().byId("p_nome").getValue(),
sap.ui.getCore().byId("s_nome").getValue(),
sap.ui.getCore().byId("idade").getValue(),
)
}
});
var elemento = [oMatrix,oLayout];
return elemento;
});

```

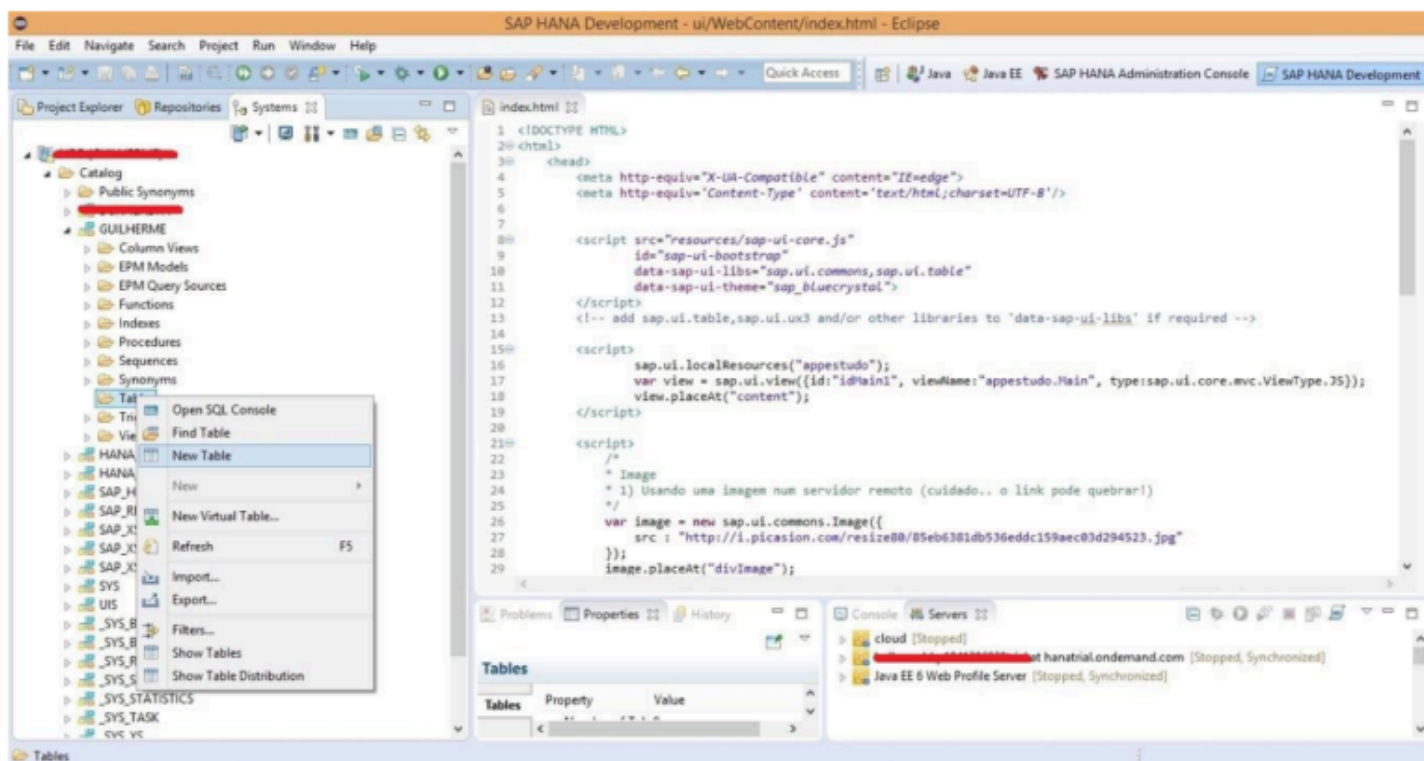
Para testar a interface de nossa aplicação vamos com o botão direito em index -> Run As -> Web App Preview.



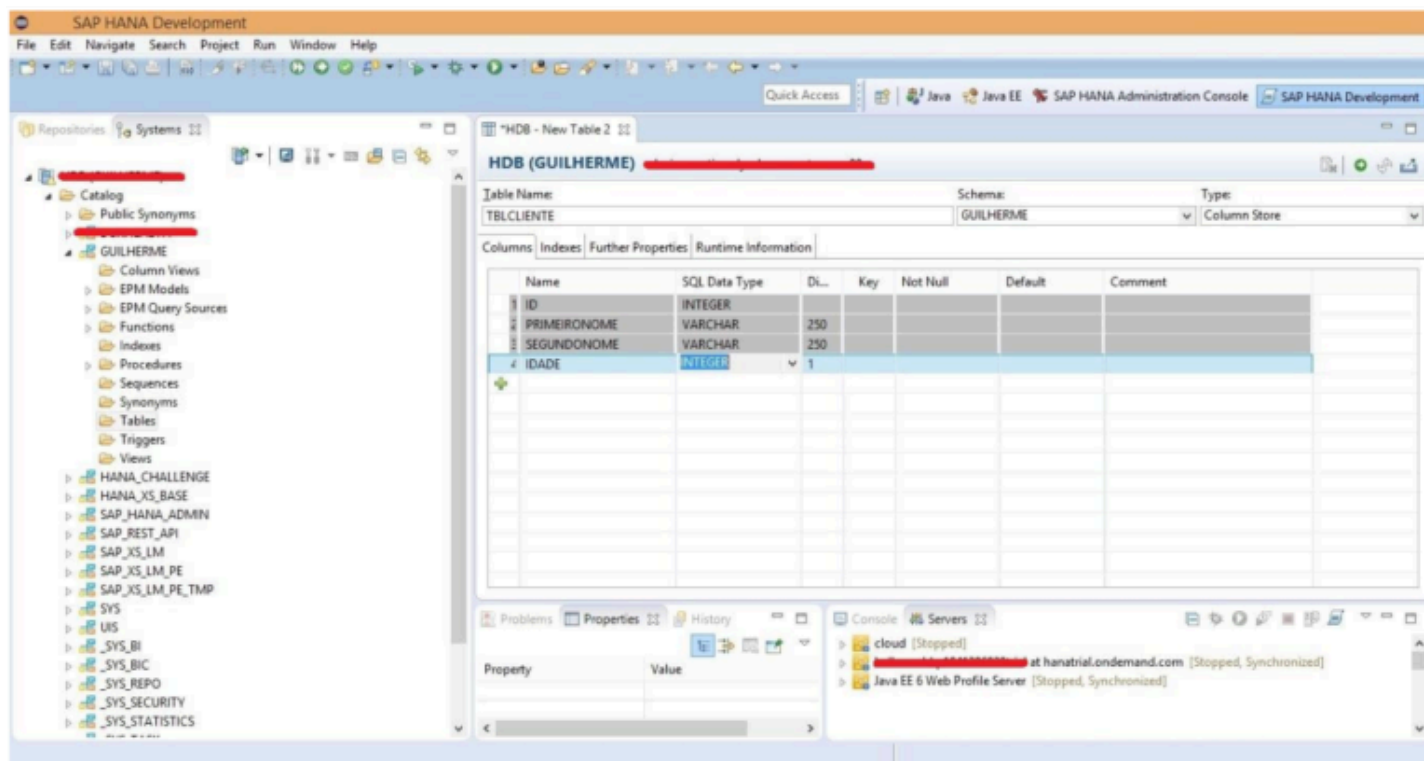
Ao executar o "index.html" de nossa aplicação teremos o seguinte resultado.



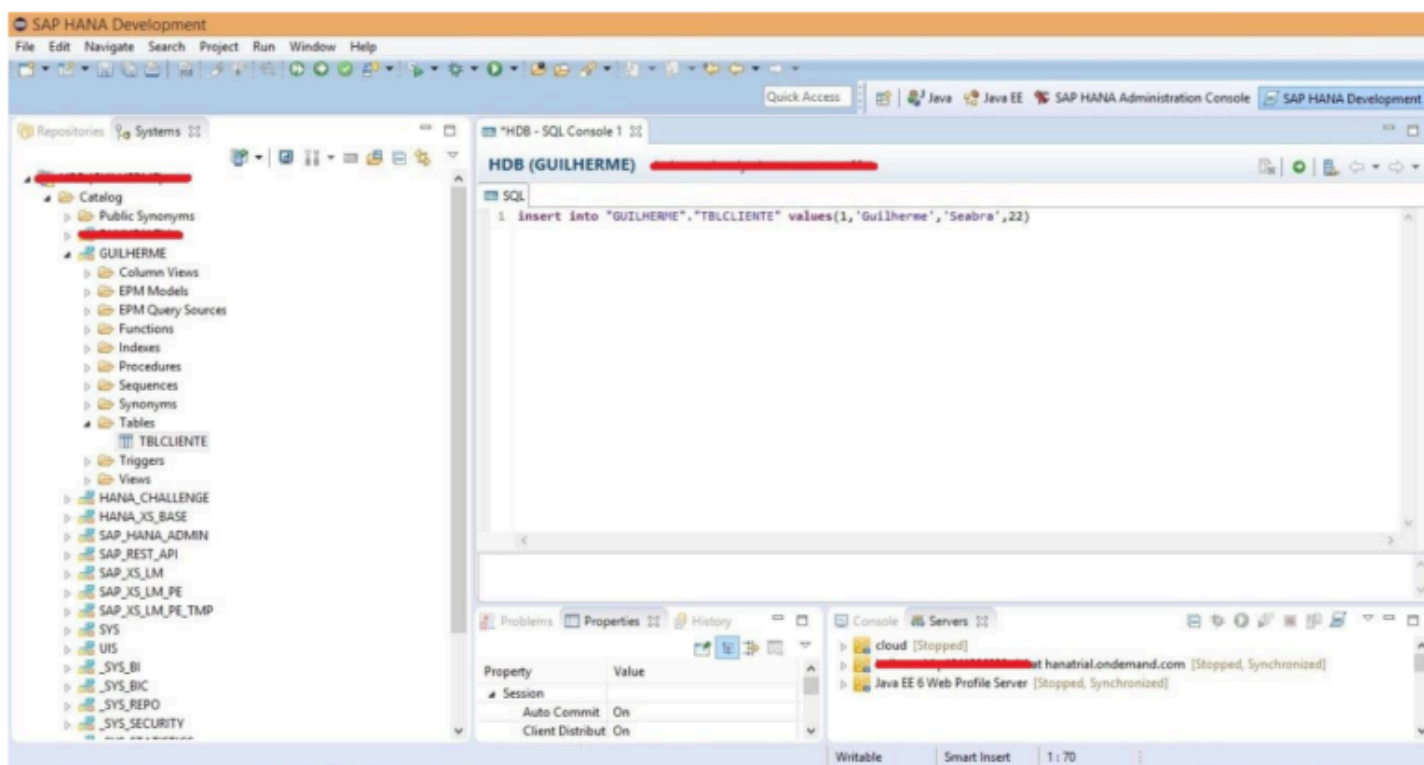
Bom, nossa aplicação "SAP UI5" está criada e funcionando perfeitamente, o próximo passo é criar um banco de dados no "Hana". Vou acessar meu diretório, em seguida escolher a opção Table -> New table.



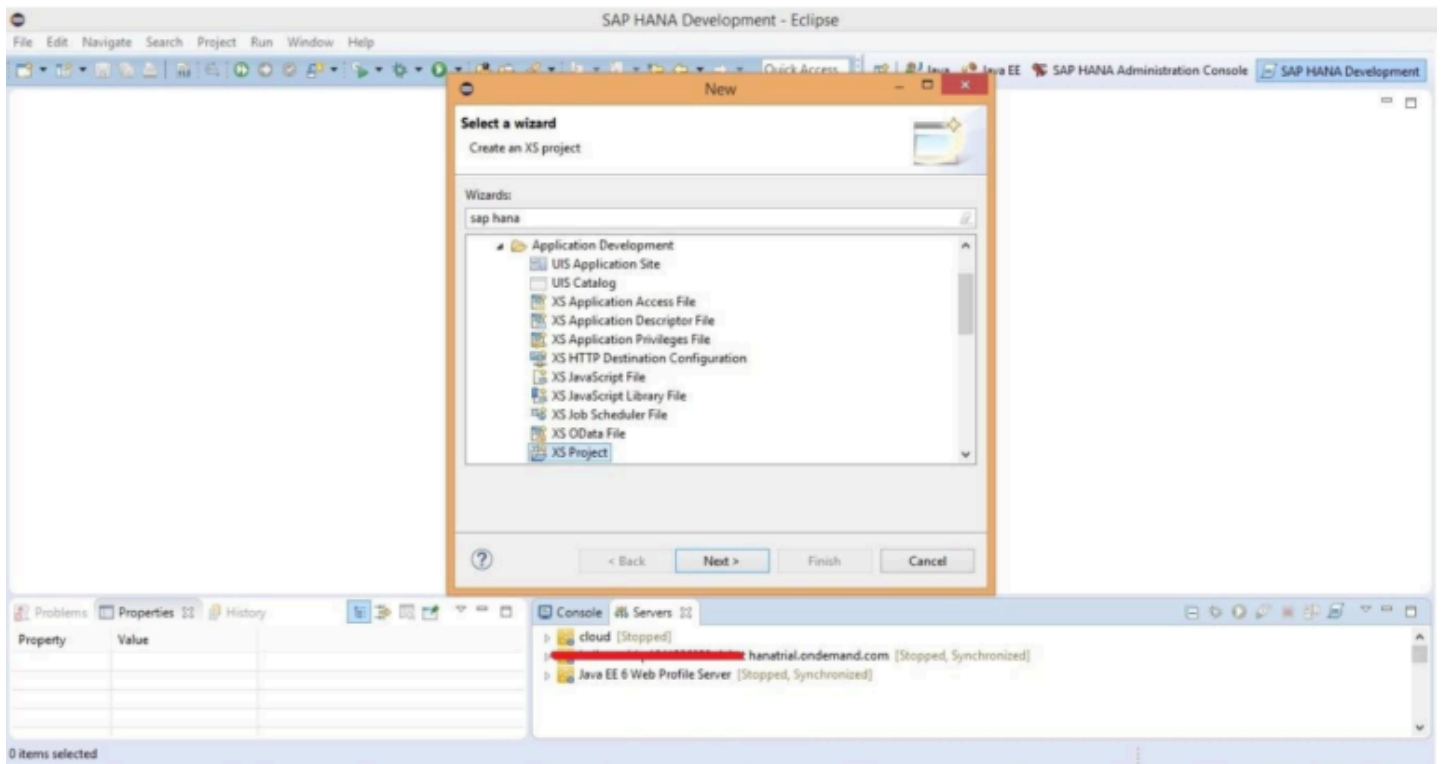
Agora vamos dar um nome a essa tabela, vou chama-la de "TBLCLIENTE" e criar quatro atributos que irá receber o conteúdo dos controles no arquivo "UI5.view.js" de nossa aplicação.



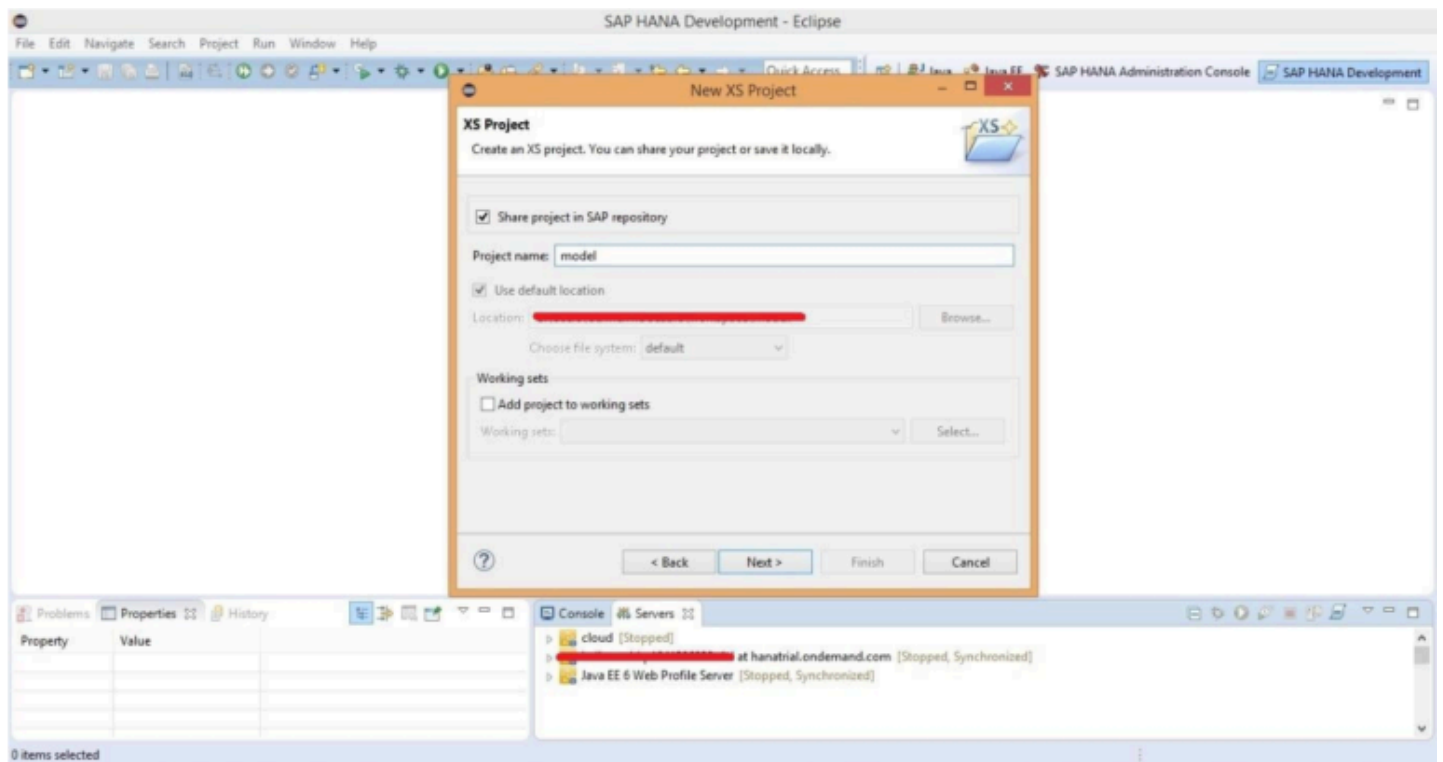
Pronto, nossa tabela está criada, vamos testar se ela está de acordo com o que precisamos, para isso vamos dar uma carga a ela, nesse caso vou inserir apenas um registro.



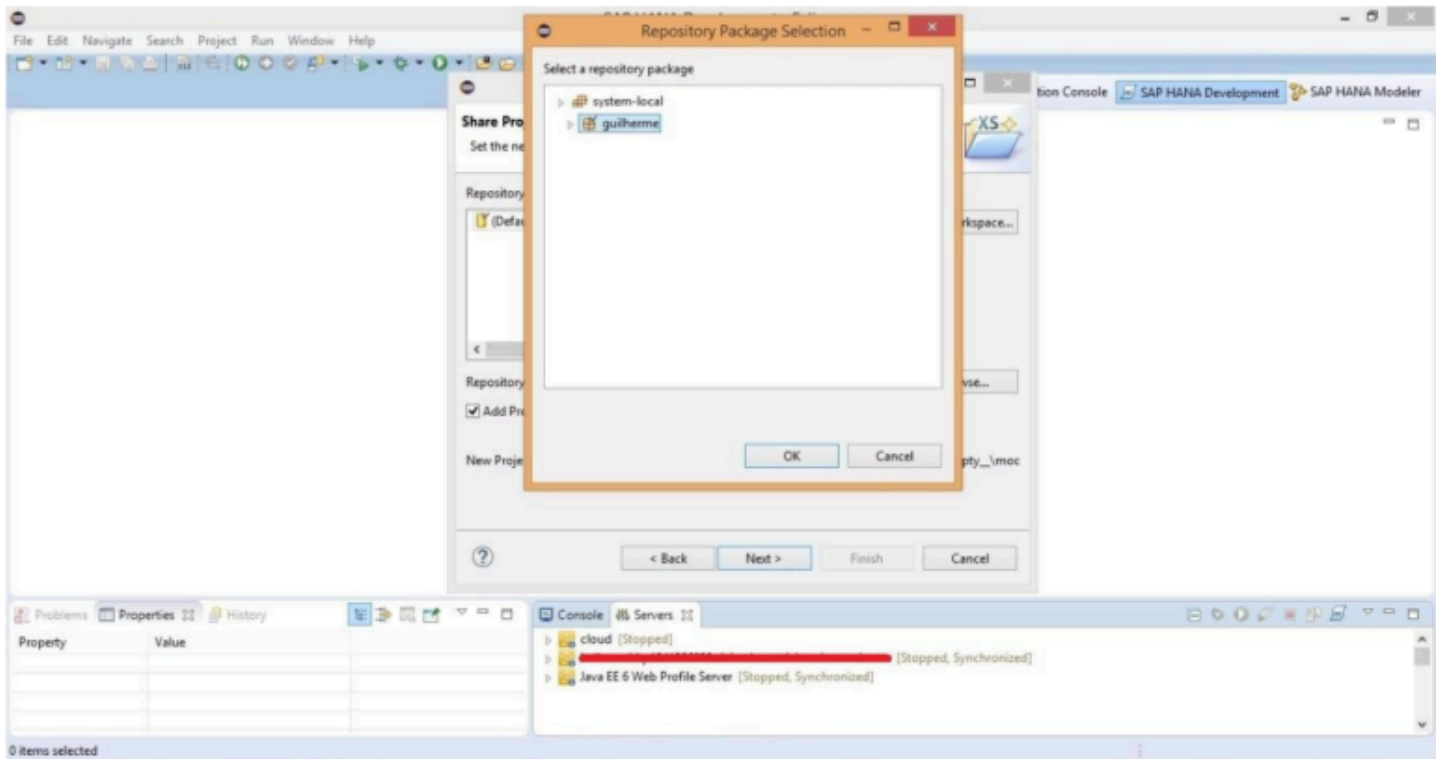
Nossa tabela está recebendo registro corretamente, agora precisamos criar um novo arquivo que vai conter um arquivo "xsjs" com a função de inserir nessa tabela, para que podemos inserir usando parâmetros, para isso vamos clicar em File -> New -> Other -> SAP Hana -> Application Development -> XS Project.



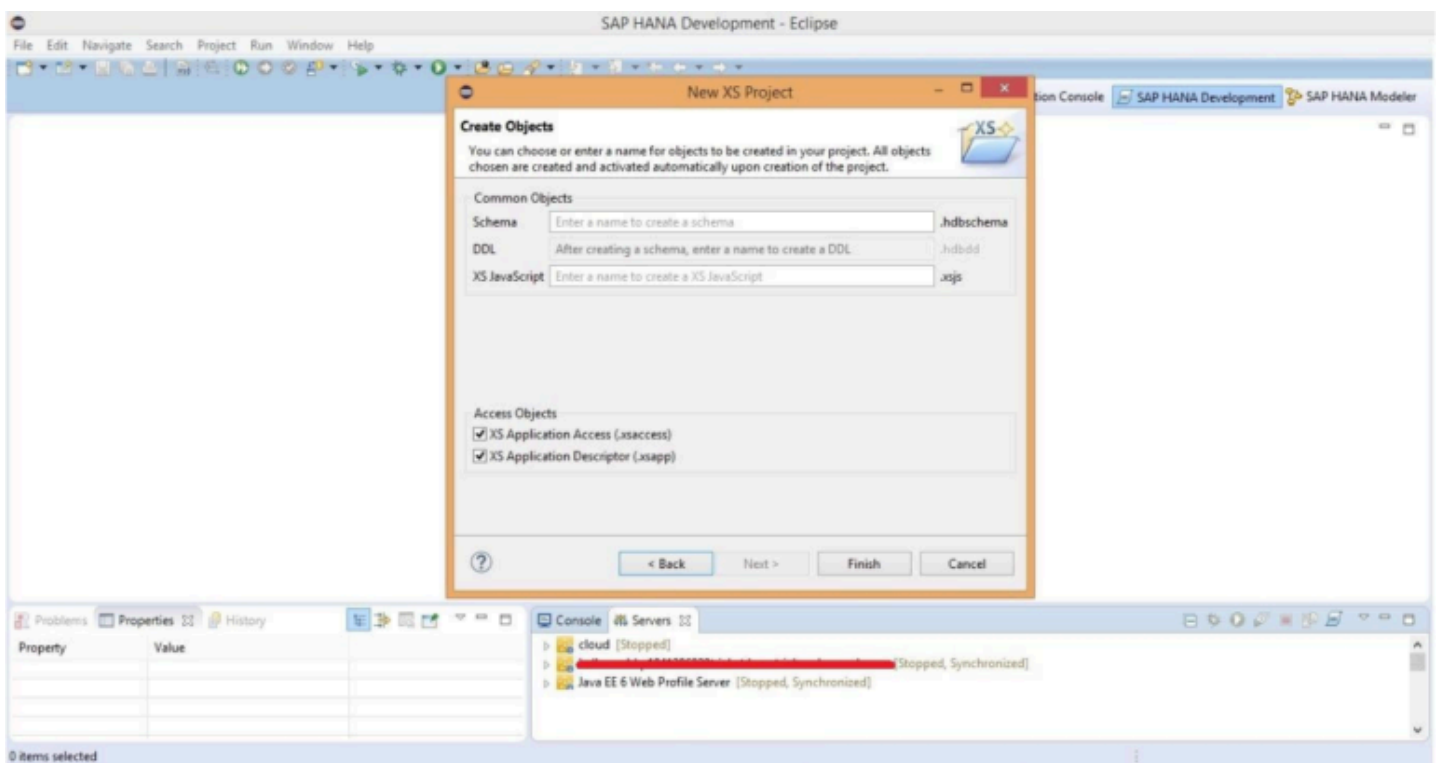
Logo em seguida vamos dar um nome a esse projeto e subi-lo para algum servidor Hana.



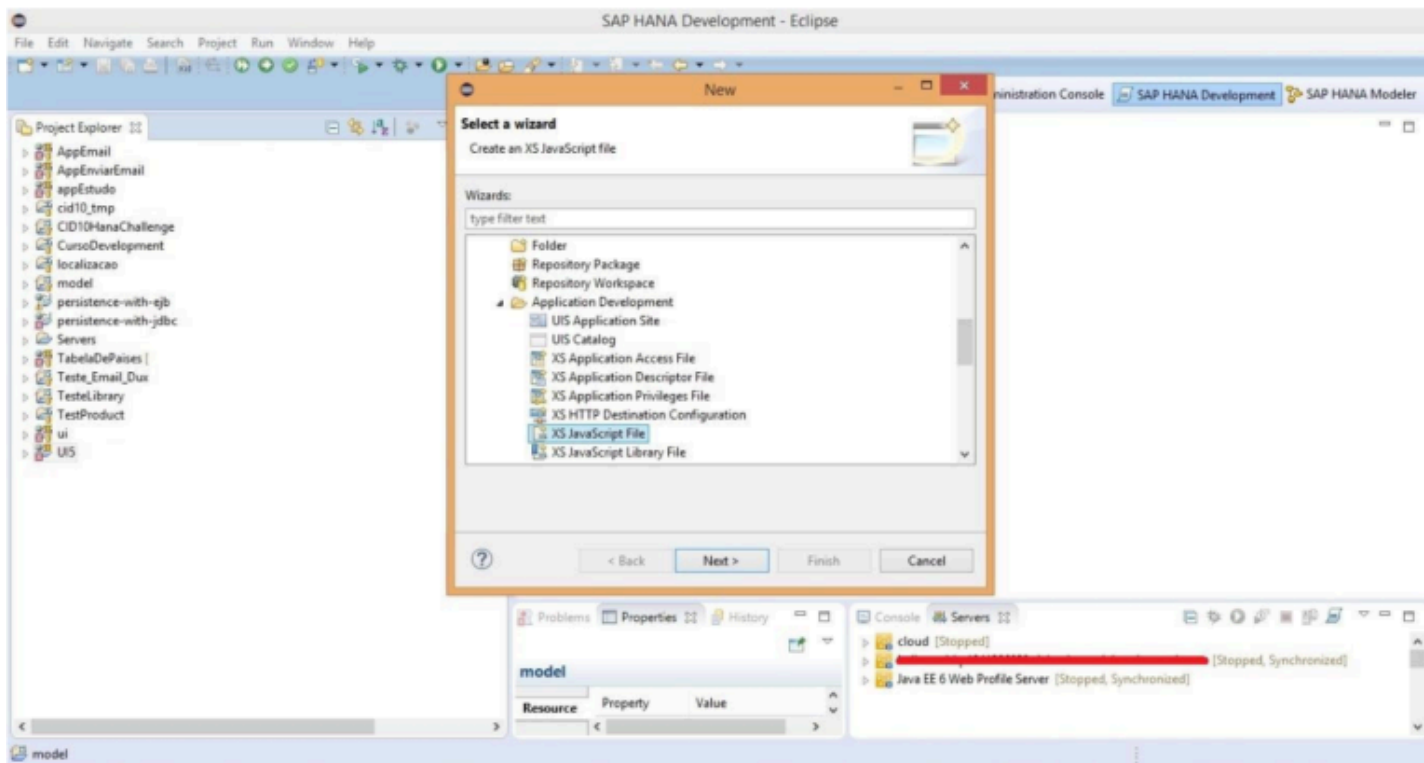
Agora vamos escolher o diretório onde iremos subir nossa aplicação.



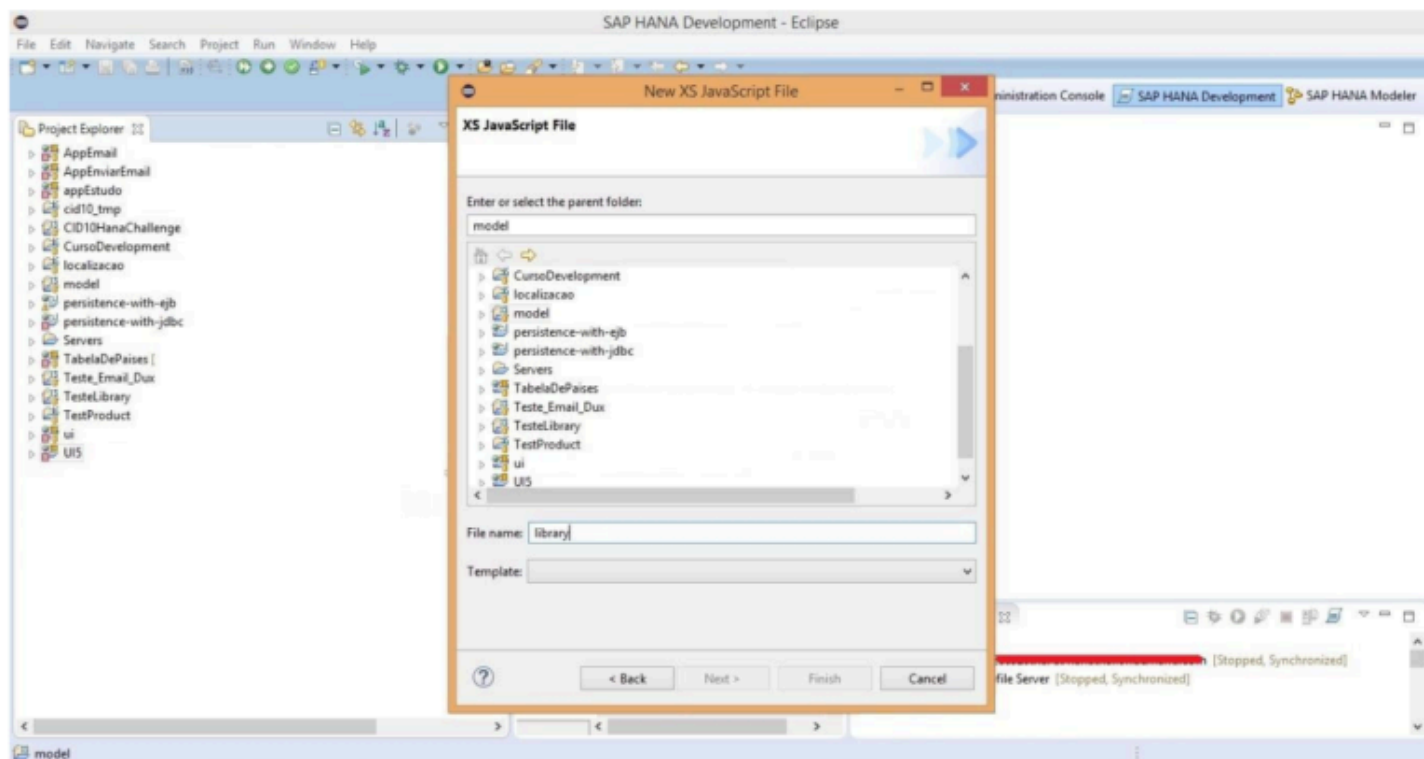
Vamos agora somente finalizar a criação de nosso projeto "XS Project".



Dentro desse projeto vamos criar um novo arquivo do tipo "xsjs" selecionando nosso diretório "model" clicando em New -> Other -> XS JavaScript File.



Vamos agora dar um nome a esse arquivo, eu vou chama-lo de "library".



Bom, vai ser na "library.xsjs" que vai ter todas as funções que vamos chama-las em nossa aplicação UI5. Vamos dar dois cliques sobre ele e inserir a função para inserir um registro no Hana, para isso vamos usar o seguinte trecho de código como mostra a Listagem 2.

Listagem 2: Criação da função se inserção no arquivo "library.xsjs".

```
function inserir() {
var txid = $.request.parameters.get("id");
var txp_nome = $.request.parameters.get("p_nome");
var txs_nome = $.request.parameters.get("s_nome");
var txidade = $.request.parameters.get("idade");
```

```

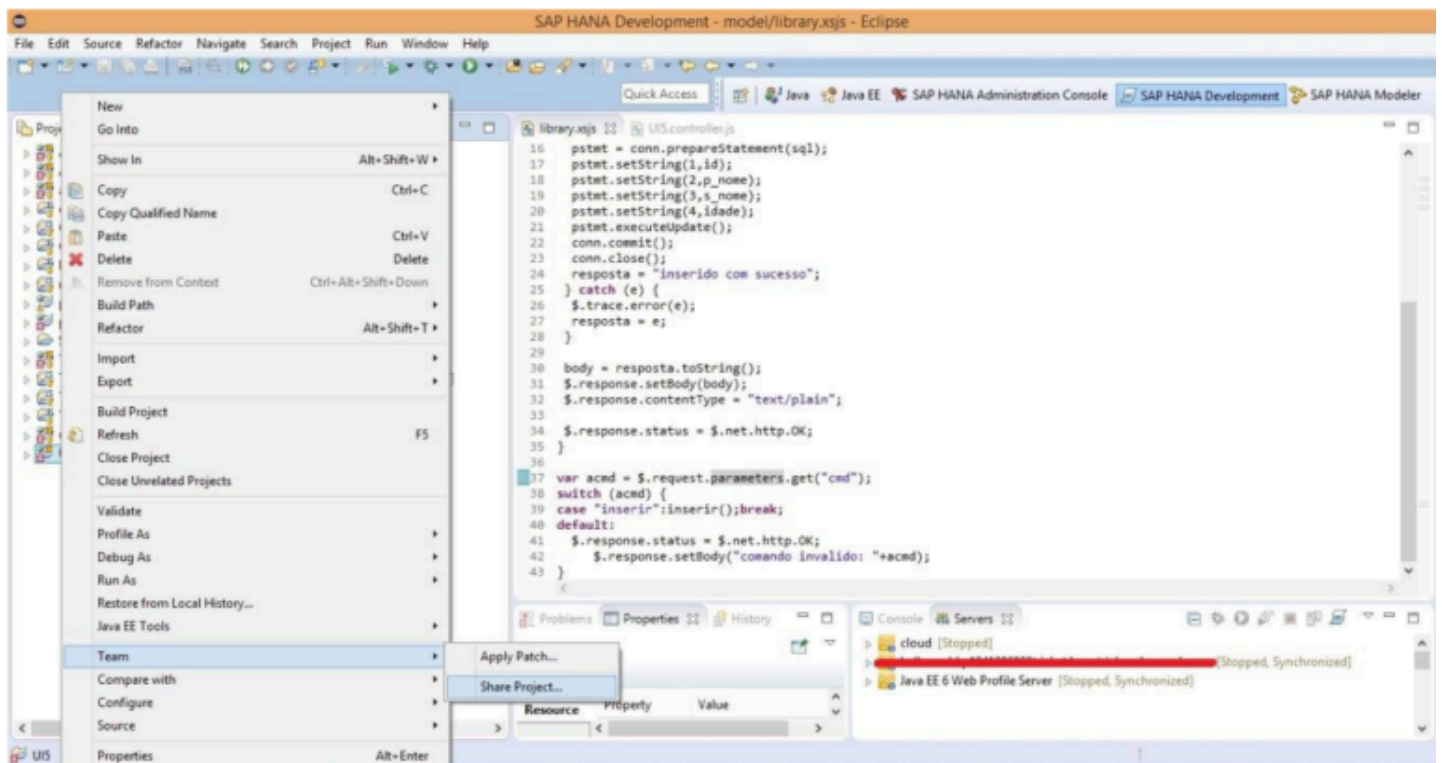
var body = "";
var conn = $.db.getConnection();
var pstmt = null;
var resposta;
try {
var sql = "insert into GUILHERME.TBLCLIENTES values (?, ?, ?, ?) ";
pstmt = conn.prepareStatement(sql);
pstmt.setString(1, id);
pstmt.setString(2, p_nome);
pstmt.setString(3, s_nome);
pstmt.setString(4, idade);
pstmt.executeUpdate();

conn.commit();
conn.close();

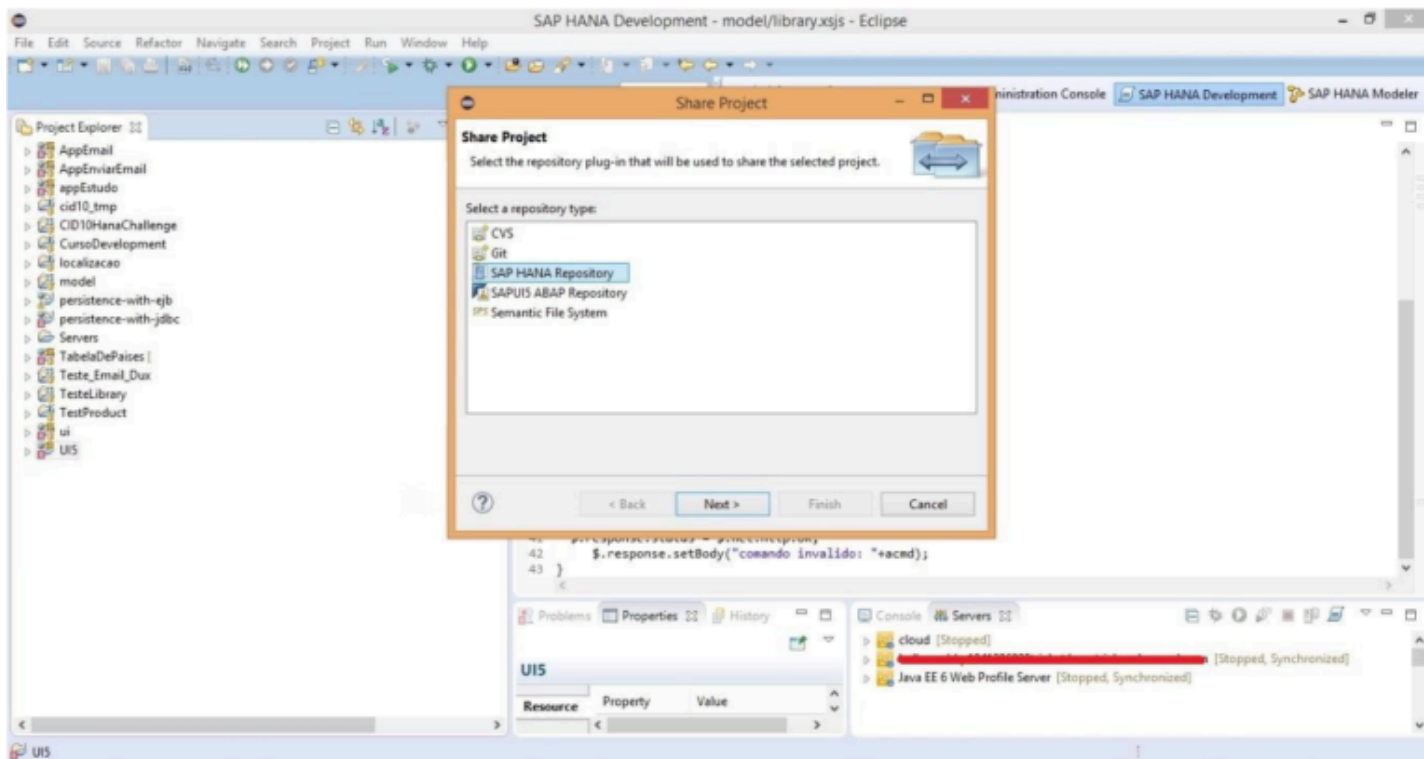
resposta = "inserido com sucesso";
} catch (e) {
$.trace.error(e);
resposta = e;
}
body = resposta.toString();
$.response.setBody(body);
$.response.contentType = "text/plain";
$.response.status = $.net.http.OK;
}
var acmd = $.request.parameters.get("cmd");
switch (acmd) {
case "inserir": inserir(); break;
default:
$.response.status = $.net.http.OK;
$.response.setBody("comando invalido: " + acmd);
}

```

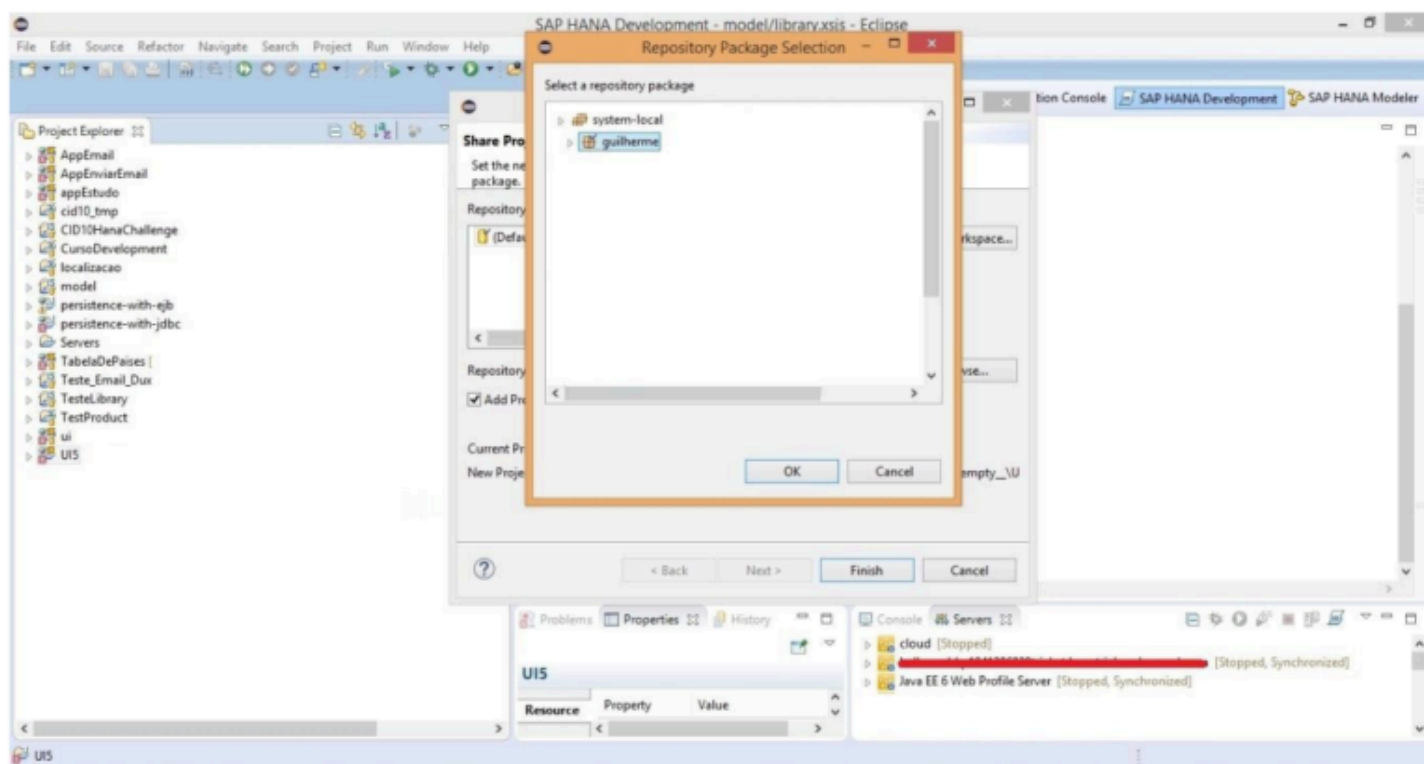
Agora que criamos nossa função de inserção vamos subir para o Hana esse projeto como fizemos com o model. Para isso vamos clicar em cima do projeto, selecionando a opção Team -> Share Project.



Vamos agora escolher a opção SAP Hana Repository e clicar em next.



Vamos agora escolher nosso repositório para subir nossa aplicação, e logo após podemos clicar em "Finish".



Bom, agora que subimos nossas duas aplicações vamos acessar o arquivo "UI5.controller.js" para apontar nossa aplicação UI5 para o arquivo "xsjs". Vamos dar dois cliques no arquivo "UI5.controller.js" e colocar o seguinte trecho de código conforme ilustra a Listagem 3.

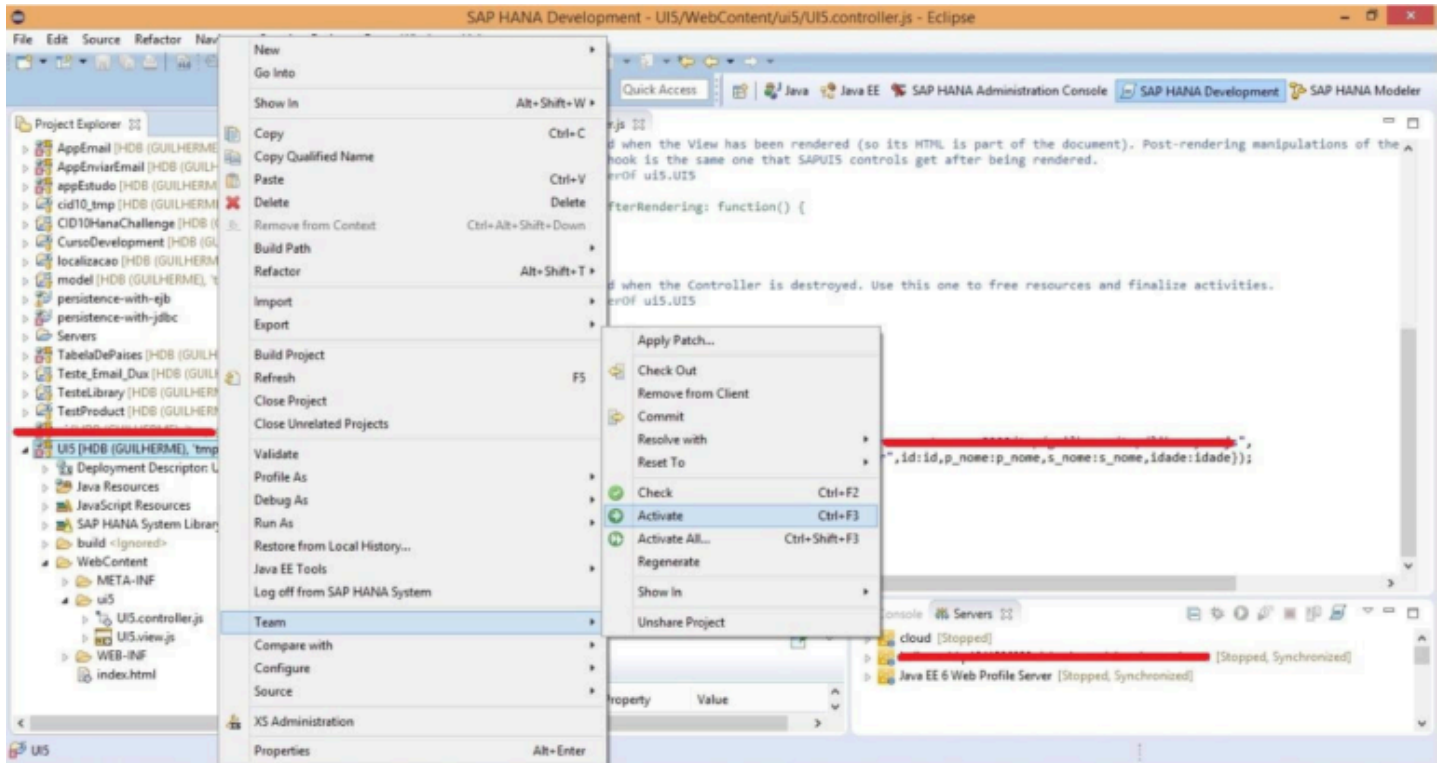
Listagem 3: Passando o caminho e o parâmetro no arquivo "UI5.controller.js" da aplicação.

```
sap.ui.controller("ui5.UI5", {
  save:function(id,p_nome,s_nome,idade) {

  try {
```

```
$.get("http://caminho_servidor/nome_arquivo.xsjs",
{cmd:"inserir",id:id,p_nome:p_nome,s_nome:s_nome,idade:idade});
} catch (e) {
}
});
```

Bom, agora vamos subir essa última modificação que fizemos no arquivo "UI5.controller.js, clicando com o botão direito no projeto vamos selecionar a opção Team -> Activate.



Pronto, nossa aplicação já se encontra em nosso servidor, agora já é possível testar nossa aplicação, antes de rodar nossa aplicação e preciso abrir o index.html e acrescentar a "src" do projeto como mostra a Listagem 4.

Listagem 4: Arquivo "Index.html" modificado para ser acessado na web.

```
<html>
<head>
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
<script src="/sap/ui5/1/resources/sap-ui-core.js"
id="sap-ui-bootstrap"
data-sap-ui-libs="sap.ui.commons"
data-sap-ui-theme="sap_bluecrystal">
</script>
<!-- add sap.ui.table, sap.ui.ux3 and/or other libraries to 'data-sap-ui-libs' if required -->
<script>
sap.ui.localResources("ui5");
```

```

var view = sap.ui.view({id:"idUI51", viewName:"ui5.UI5", type:sap.ui.core.mvc.ViewType.JS});
view.placeAt("content");

</script>

</head>

<body class="sapUiBody" role="application">

<div id="content"></div>

</body>

</html>

```

Bom pessoal, espero que tenham gostado desse artigo, tentei ser o mais claro possível e levantar as principais dúvidas que eu também encontrei na criação de minhas primeiras aplicações, bom desenvolvimento a todos e até a próxima.

Comentários

matheus — 02/06/2017 16:44

Nesse caso o arquivo javascript no bd Hana está fazendo o papel do Model no MVC ?

No caso eu tendo uma instância gateway e codificar no ECC um serviço em abap pela SEGW teria o mesmo resultado só que de outra forma, certo ?

carlos — 27/04/2017 15:21

Boa tarde.

As imagens não aparecem, já tentei em redes diferentes.

Com navegadores diferentes e nada.

carlos — 23/04/2017 07:53

Bom dia.

As imagens não estão carregando.