



Universitat
Autònoma
de Barcelona



**PROYECTO N° 2017. IMPLEMENTACIÓN DE MOVIMIENTOS SUAVES EN
ROBOTS INDUSTRIALES MEDIANTE SPLINES CÚBICOS. APLICACIÓN A UN
LABORATORIO REMOTO DE ROBÓTICA**

ANEXOS

Anexos del Proyecto Fin de Carrera
De Ingeniería en Informàtica

Realizado por

Marcos Mohedano Gómez

y dirigido por

Asier Ibeas Hernández

Bellaterra, Junio 20

ÍNDICE DE ANEXOS

- Anexo 1.** Código de la interfaz de la página web “Login.aspx”.
- Anexo 2.** Código de la interfaz de la página web “laboratorio.aspx”.
- Anexo 3.** Código del archivo “Global.asax”.
- Anexo 4.** Código del archivo “web.config”.
- Anexo 5.** Código de la capa de negocio de la página web “Login.aspx”.
- Anexo 6.** Código de la capa de negocio de la página web “laboratorio.aspx”.
- Anexo 7.** Manual de usuario.
- Anexo 8.** Gestión de cuentas de usuario.
- Anexo 9.** Captura de vídeo del sistema en funcionamiento (Archivo de vídeo adjunto en el CD de la memoria en formato electrónico).



Universitat
Autònoma
de Barcelona



**PROYECTO N° 2017. IMPLEMENTACIÓN DE MOVIMIENTOS SUAVES EN
ROBOTS INDUSTRIALES MEDIANTE SPLINES CÚBICOS. APLICACIÓN A UN
LABORATORIO REMOTO DE ROBÓTICA**

ANEXO 1

Anexo del Proyecto Fin de Carrera
De Ingeniería en Informàtica

Realizado por

Marcos Mohedano Gómez

y dirigido por

Asier Ibeas Hernández

Bellaterra, Junio 2010

```

<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="Login.aspx.cs" Inherits="_Default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title title="Laboratório RAI"></title>

    <style type="text/css">
        #form1
        {
            height: 556px;
        }
    </style>

</head>
<body bgcolor="#F0F0FF">
    <form id="form1" runat="server" style="height:600px; width:734px">
        <div id="label"
            style="width: 600px; height: 100px; top: 50px; left: 330px;
position: absolute;">
            <asp:Label ID="Label1" runat="server"
                Text="Laboratório de Robótica: trayectorias suaves
mediante splines cúbicos"
                Font-Bold="True" Font-Size="XX-Large" ForeColor="#3366CC"
Width="600px"></asp:Label>
        </div>

        <div id="imagen"

            style="position: absolute; top: 33px; left: 12px; width:
268px; height: 320px;">
            <asp:Image ID="Image1" runat="server"
                ImageUrl="http://localhost/imagenes/laboratoriorobot2.png"
/>
        </div>
        <div id="imagen2"

            style="left:690px; top:356px; position: absolute; height:
252px; width: 327px;">
            <asp:Image ID="Image2" runat="server"
                ImageUrl="http://localhost/imagenes/NaturalCubicSpline.png"
" Height="262px"
                style="margin-left: 0px" Width="319px" />
        </div>
        <div id="cuadrologin"

            style="position: absolute; width: 404px; height: 437px; top:
140px; left: 330px">

            <asp:Login ID="Login1" runat="server" BackColor="#EFF3FB"
BorderColor="#B5C7DE"
                BorderPadding="4" BorderStyle="Solid" BorderWidth="1px"
Font-Names="Verdana"
                Font-Size="0.8em" ForeColor="#333333" Height="160px"
Width="340px"

```

```

        onauthenticate="Login1_Authenticate"
MembershipProvider="provedorSql"
        onloggedin="Login1_LoggedIn">
        <TextBoxStyle Font-Size="0.8em" />
        <LoginButtonStyle BackColor="White" BorderColor="#507CD1"
BorderStyle="Solid"
            BorderWidth="1px" Font-Names="Verdana" Font-
Size="0.8em"
                ForeColor="#284E98" />
        <InstructionTextStyle Font-Italic="True" ForeColor="Black"
/>
        <TitleTextStyle BackColor="#507CD1" Font-Bold="True" Font-
Size="0.9em"
            ForeColor="White" />
    </asp:Login>
    <br />
    &nbsp;

<asp:CreateUserWizard ID="CreateUserWizard1" runat="server"
BackColor="#EFF3FB"
    BorderColor="#B5C7DE" BorderStyle="Solid"
    BorderWidth="1px"
    Font-Names="Verdana" Font-Size="0.8em" Width="350px"
    ContinueDestinationPageUrl="~/laboratorio.aspx"
    RequireEmail="False"
    MembershipProvider="provedorSql">
    <SideBarStyle BackColor="#507CD1" Font-Size="0.9em"
        VerticalAlign="Top" />
    <SideBarButtonStyle Font-Names="Verdana" ForeColor="White"
        BackColor="#507CD1" />
    <ContinueButtonStyle BackColor="White"
    BorderColor="#507CD1"
        BorderStyle="Solid" BorderWidth="1px" Font-
Names="Verdana"
            ForeColor="#284E98" />
    <NavigationButtonStyle BackColor="White"
    BorderColor="#507CD1"
        BorderStyle="Solid" BorderWidth="1px" Font-
Names="Verdana"
            ForeColor="#284E98" />
    <HeaderStyle BackColor="#284E98" BorderStyle="Solid" Font-
Bold="True"
        Font-Size="0.9em" ForeColor="White"
    HorizontalAlign="Center"
        BorderColor="#EFF3FB" BorderWidth="2px" />
    <CreateUserButtonStyle BackColor="White"
    BorderColor="#507CD1"
        BorderStyle="Solid" BorderWidth="1px" Font-
Names="Verdana"
            ForeColor="#284E98" />
    <TitleTextStyle BackColor="#507CD1" Font-Bold="True"
    ForeColor="White" />
    <StepStyle Font-Size="0.8em" />
    <WizardSteps>
        <asp:CreateUserWizardStep runat="server" />
        <asp:CompleteWizardStep runat="server" />
    </WizardSteps>
    </asp:CreateUserWizard>

</div>
</form>

```

```
</body>  
</html>
```



Universitat
Autònoma
de Barcelona



escola
d'enginyeria

**PROYECTO Nº 2017. IMPLEMENTACIÓN DE MOVIMIENTOS SUAVES EN
ROBOTS INDUSTRIALES MEDIANTE SPLINES CÚBICOS. APLICACIÓN A UN
LABORATORIO REMOTO DE ROBÓTICA**

ANEXO 2

Anexo del Proyecto Fin de Carrera
De Ingeniería en Informàtica

Realizado por

Marcos Mohedano Gómez

y dirigido por

Asier Ibeas Hernández

Bellaterra, Junio 2010

```
<%@Page Language="C#" AutoEventWireup="true" CodeFile="laboratorio.aspx.cs"
Inherits="laboratorio" %>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head runat="server">
```

```
<title>LABORATORIO RAI - TRAYECTORÍAS SUAVES MEDIANTE SPLINES
CÚBICOS</title>
```

```
<style type="text/css" media="screen">
```

```
#form1
```

```
{
```

```
height: 752px;
```

```
font-style: normal;
```

```
font-variant: normal;
```

```
font-weight: normal;
```

```
font-size: medium;
```

```
line-height: normal;
```

```
font-family: "Arial Black";
```

```
}
```

```
</style>
```

```
</head>
```

```
<body runat="server" id="mybody" enableviewstate="false" bgcolor="#F0F0FF">
```

```
<form id="form1" runat="server" style="width: 985px; height: 674px">
```

```
<asp:SqlDataSource ID="SqlDataSource3" runat="server"
```

```
ConnectionString="<%%$ ConnectionStrings:usuariosConnectionString %>"
```

```
SelectCommand="SELECT [botones], [comando_sc32] FROM
```

```
[equivalencias]"></asp:SqlDataSource>
```

```
<asp:ScriptManager ID="ScriptManager1" runat="server">
```

```
</asp:ScriptManager>
```

```
<div id="imagen_manual"
```

```
style="position: absolute; top: 16px; left: 338px; height: 77px; width:
420px;">
```

```
<asp:ImageButton ID="ImageButton1" runat="server"
```

```
ImageUrl="~/imagenes/imagen_manual.png" Width="212px"
```

```
Height="76px"
```

```
CausesValidation="False" onclick="ImageButton1_Click" />
```

```
</div>
```

```
<div id="imagen_verificador"
```

```
style="position: absolute; top: 22px; left: 600px; height: 77px; width:
154px;">
```

```

        <asp:ImageButton ID="ImageButton4" runat="server"
            ImageUrl="~/imagenes/imagen_verificador.png" Height="51px"
Width="133px"
            CausesValidation="False" onclick="ImageButton4_Click" />
    </div>

    <div id="imagen_etse" style="position: absolute; top: 10px; left: 800px">
        <asp:Image ID="Image1" runat="server"
ImageUrl="~/imagenes/etse1.png"
            Width="200px" />
    </div>
    <div id="imagen_robot"
style="position: absolute; top: 310px; left: 282px; height: 349px; width:
430px;">
        <asp:Image ID="Image2" runat="server"
            ImageUrl="~/imagenes/fig3.png" Height="419px"
            Width="452px" style="margin-left: 62px" />
    </div>
    <div id="imagen_spline" style="position: absolute; top: 340px; left: 400px">
        <asp:Image ID="Image3" runat="server" ImageUrl="~/imagenes/2010-01-
30_2347.png"
            Height="329px" style="margin-left: 182px; margin-top: 91px" />
    </div>
    <div id="subirarchivos" style="left: 340px; width:
548px; height: 100px; top: 90px; position: absolute">
        <hr />
        <asp:FileUpload ID="FileUpload1" runat="server" />
        <asp:Button ID="botonsubir" runat="server" Text="Subir archivo"
            onclick="botonsubir_Click" CausesValidation="False" />
        <br />
        <asp:Label ID="Label1" runat="server" ForeColor="Red"></asp:Label>
        <hr />
    </div>
    <asp:UpdatePanel ID="UpdatePanel1" runat="server"
UpdateMode="Conditional">
        <ContentTemplate>
            <asp:Timer ID="Timer1" runat="server" Interval="5000"
ontick="Timer1_Tick">
                </asp:Timer>
                <div id="colausuarios"
                    style="position: absolute; top: 98px; height: 66px; width: 106px; left:
895px;">
                    <asp:GridView ID="GridView2" runat="server"
AutoGenerateColumns="False"
                        CellPadding="4" DataSourceID="SqlDataSource1"
GridLines="None"
                            HorizontalAlign="Justify"
onrowcreated="GridView2_RowCreated"

```

```

        style="margin-left: 0px" Width="100px"
onrowdeleted="GridView2_RowDeleted" ForeColor="#333333"
        >
        <RowStyle BackColor="#F7F6F3" ForeColor="#333333"
HorizontalAlign="Justify"
        VerticalAlign="Middle" />
        <Columns>
        <asp:BoundField DataField="username" HeaderText="Cola
usuarios"
        SortExpression="username" ReadOnly="True">
        <ItemStyle HorizontalAlign="Center" VerticalAlign="Middle"
/>
        </asp:BoundField>
        </Columns>
        <FooterStyle BackColor="#5D7B9D" Font-Bold="True"
ForeColor="White" />
        <PagerStyle BackColor="#284775" ForeColor="White"
HorizontalAlign="Center" />
        <SelectedRowStyle BackColor="#E2DED6" Font-Bold="True"
ForeColor="#333333" />
        <HeaderStyle BackColor="#5D7B9D" Font-Bold="True"
ForeColor="White" />
        <EditRowStyle BackColor="#999999" />
        <AlternatingRowStyle BackColor="White" ForeColor="#284775"
/>
        </asp:GridView>
        <asp:SqlDataSource ID="SqlDataSource1" runat="server"
        ConnectionString="<%$
ConnectionStrings:usuariosConnectionString %>"
        SelectCommand="SELECT [username]FROM [tanda] ORDER BY
[orden]"
        ondeleted="SqlDataSource1_Deleted"></asp:SqlDataSource>
    </div>

    <div id="trayectorias"
        style="left: 340px; width: 500px; height: 280px; top: 170px; position:
absolute">

        <asp:Button ID="Button1" runat="server" Text="Nuevo nodo"
Height="26px"
        onclick="Button1_Click" Width="91px" style="margin-bottom: 0px" />

        <asp:Button ID="Button2" runat="server" Text="Enviar" Height="26px"
onclick="Button2_Click" Width="91px" style="margin-top: 0px"
        CausesValidation="False" />

        <asp:Button ID="Button3" runat="server" Text="Borrar nodos"
Height="26px"

```

```

        onclick="Button3_Click" Width="91px" style="margin-top: 0px"
        CausesValidation="False" />
    <br />
    <br />
    <p />

    <asp:GridView ID="GridViewnodos" runat="server"
    AutoGenerateColumns="False"
        CellPadding="4" DataSourceID="SqlDataSource2"
    GridLines="None"
        Height="44px" Width="550px"
        ForeColor="#333333">
        <RowStyle BackColor="#F7F6F3" BorderStyle="None"
    HorizontalAlign="Center"
        ForeColor="#333333" />
        <Columns>
            <asp:BoundField DataField="circunferencia"
    HeaderText="circunferencia"
        SortExpression="circunferencia" ReadOnly="True" >
            <ItemStyle HorizontalAlign="Center" VerticalAlign="Middle"
    />
        </asp:BoundField>
            <asp:BoundField DataField="radio" HeaderText="radio"
    SortExpression="radio"
        ReadOnly="True" />
            <asp:BoundField DataField="plano" HeaderText="plano"
    SortExpression="plano" />
            <asp:BoundField DataField="altura" HeaderText="altura"
        SortExpression="altura" />
            <asp:BoundField DataField="zona" HeaderText="zona"
    SortExpression="zona" />
            <asp:BoundField DataField="pinza" HeaderText="pinza"
    SortExpression="pinza" />
            <asp:BoundField DataField="tiempo" HeaderText="tiempo"
        SortExpression="tiempo" />
        </Columns>
        <FooterStyle BackColor="#5D7B9D" Font-Bold="True"
    ForeColor="White" />
        <PagerStyle BackColor="#284775" ForeColor="White"
    HorizontalAlign="Center" />
        <SelectedRowStyle BackColor="#E2DED6" Font-Bold="True"
    ForeColor="#333333" />
        <HeaderStyle BackColor="#5D7B9D" Font-Bold="True"
    ForeColor="White" />
        <EditRowStyle BackColor="#999999" />
        <AlternatingRowStyle BackColor="White" ForeColor="#284775"
    />
    </asp:GridView>
    <asp:SqlDataSource ID="SqlDataSource2" runat="server"

```

```

ConnectionString="<%$
ConnectionStrings:usuariosConnectionString %>"

```

```

SelectCommand="SELECT [circunferencia], [radio], [plano],
[altura], [zona], [pinza], [tiempo] FROM [nodos_trayectoria] WHERE ([username] =
@username) ORDER BY [orden]">

```

```

<SelectParameters>
  <asp:SessionParameter DefaultValue="idusuario"
Name="username"

```

```

  SessionField="idusuario" Type="String" />
</SelectParameters>
</asp:SqlDataSource>
</div>

```

```

</ContentTemplate>
</asp:UpdatePanel>
<div id="dropdownlists" style="position: absolute; top: 200px; left: 341px">
<asp:DropDownList ID="DLcircunferencia" runat="server" >
  <asp:ListItem Value="0">Circunferencia</asp:ListItem>
  <asp:ListItem>C1</asp:ListItem>
  <asp:ListItem>C2</asp:ListItem>
  <asp:ListItem>C3</asp:ListItem>
  <asp:ListItem>C4</asp:ListItem>
  <asp:ListItem>C5</asp:ListItem>
  <asp:ListItem>C6</asp:ListItem>
</asp:DropDownList>
<asp:DropDownList ID="DLradio" runat="server" >
  <asp:ListItem Value="0">Radio</asp:ListItem>
  <asp:ListItem>RA</asp:ListItem>
  <asp:ListItem>RB</asp:ListItem>
  <asp:ListItem>RC</asp:ListItem>
  <asp:ListItem>RD</asp:ListItem>
  <asp:ListItem>RE</asp:ListItem>
  <asp:ListItem>RF</asp:ListItem>
  <asp:ListItem>RG</asp:ListItem>
  <asp:ListItem>RH</asp:ListItem>
  <asp:ListItem>RI</asp:ListItem>
  <asp:ListItem>RJ</asp:ListItem>
  <asp:ListItem>RK</asp:ListItem>
  <asp:ListItem>RL</asp:ListItem>
  <asp:ListItem>RM</asp:ListItem>
  <asp:ListItem>RN</asp:ListItem>
  <asp:ListItem>RO</asp:ListItem>
  <asp:ListItem>RP</asp:ListItem>
  <asp:ListItem>RQ</asp:ListItem>
</asp:DropDownList>
<asp:DropDownList ID="DLplano" runat="server" >
  <asp:ListItem Value="0">Plano</asp:ListItem>

```

```

        <asp:ListItem>P1</asp:ListItem>
        <asp:ListItem>P2</asp:ListItem>
        <asp:ListItem>P3</asp:ListItem>
        <asp:ListItem>P4</asp:ListItem>
        <asp:ListItem>P5</asp:ListItem>
    </asp:DropDownList>
    <asp:DropDownList ID="DLaltura" runat="server" >
        <asp:ListItem Value="0">Altura</asp:ListItem>
        <asp:ListItem>L</asp:ListItem>
        <asp:ListItem>H</asp:ListItem>
    </asp:DropDownList>
    <asp:DropDownList ID="DLzona" runat="server" >
        <asp:ListItem Value="0">Zona</asp:ListItem>
        <asp:ListItem>Z1</asp:ListItem>
        <asp:ListItem>TR</asp:ListItem>
        <asp:ListItem>Z2</asp:ListItem>
    </asp:DropDownList>
    <asp:DropDownList ID="DLpinza" runat="server" >
        <asp:ListItem Value="0">Pinza</asp:ListItem>
        <asp:ListItem>Abierta</asp:ListItem>
        <asp:ListItem>Cerrada</asp:ListItem>
    </asp:DropDownList>
    <asp:TextBox ID="TextBox1" runat="server" Width="80px"
        ontextchanged="TextBox1_TextChanged">Tiempo (ms)</asp:TextBox>
    <asp:TextBox ID="TextBox2" runat="server" Width="80px">Tiempo
(ms)</asp:TextBox>
        <asp:TextBox ID="TextBoxPrimerUsuario" runat="server"
Width="80px"></asp:TextBox>

    <asp:DropDownList ID="DLOculto" runat="server" Visible="False">
    </asp:DropDownList>
    <br />

</div>
<div id="cams" style="position: absolute; top: 10px; left: 10px">

    <asp:LoginView ID="LoginView1" runat="server">
        <LoggedInTemplate>
            Bienvenido,
        </LoggedInTemplate>
    </asp:LoginView>
    <asp:LoginName ID="LoginName1" runat="server" />
    <br />
    <asp:LoginStatus ID="LoginStatus1" runat="server" Font-Names="Arial"
        onloggingout="LoginStatus1_LoggingOut" />
    <br />
    <br />
    <br />

```

```

    <object id="MediaPlayer0" classid="CLSID:22D6f312-B0F6-11D0-94AB-
0080C74C7E95"
    codebase="http://activex.microsoft.com/activex/controls/mplayer/en/nsmp2inf.c
ab#Version=6,4,7,1112"
    height="286" standby="Loading Windows Media Player components..."
    type="application/x-oleobject" width="320">
    <param name="filename" value="http://192.168.0.2:3532" />
    <param name="Showcontrols" value="0" />
    <param name="autoStart" value="True" />
    <param name="wmode" value="transparent" />
    <embed autostart="True" height="286" name="MediaPlayer0"
showcontrols="0"
    src="http://192.168.0.2:3532" type="application/x-mplayer2" width="320"
    wmode="transparent"></embed>
</object>

```

```

<br />
<br />
    <object id="MediaPlayer" classid="CLSID:22D6f312-B0F6-11D0-94AB-
0080C74C7E95"
    codebase="http://activex.microsoft.com/activex/controls/mplayer/en/nsmp2inf.c
ab#Version=6,4,7,1112"
    height="286" standby="Loading Windows Media Player components..."
    type="application/x-oleobject" width="320">
    <param name="filename" value="http://192.168.0.2:4540" />
    <param name="Showcontrols" value="0" />
    <param name="autoStart" value="True" />
    <param name="wmode" value="transparent" />
    <embed autostart="True" height="286" name="MediaPlayer" showcontrols="0"
    src="http://192.168.0.2:4540" type="application/x-mplayer2" width="320"
    wmode="transparent"></embed>
</object>

```

```

</div>

```

```

<br />
<asp:ValidationSummary ID="ValidationSummary1" runat="server"
    ShowMessageBox="True" ShowSummary="False" />
<br />
<asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server"
    ErrorMessage="Elige circunferencia"
    ControlToValidate="DLcircunferencia" Display="None"
InitialValue="0"></asp:RequiredFieldValidator>
    <br />
    <asp:RequiredFieldValidator ID="RequiredFieldValidator2"
runat="server"
    ErrorMessage="Elige radio" ControlToValidate="DLradio"
Display="None"
    InitialValue="0"></asp:RequiredFieldValidator>
    <br />

```

```

        <asp:RequiredFieldValidator ID="RequiredFieldValidator3"
runat="server"
        ErrorMessage="Elige plano" ControlToValidate="DLplano"
Display="None"
        InitialValue="0"></asp:RequiredFieldValidator>
        <br />
        <asp:RequiredFieldValidator ID="RequiredFieldValidator4"
runat="server"
        ErrorMessage="Elige altura" ControlToValidate="DLaltura"
Display="None" InitialValue="0"></asp:RequiredFieldValidator>
        <br />
        <asp:RequiredFieldValidator ID="RequiredFieldValidator5"
runat="server"
        ErrorMessage="Elige zona" ControlToValidate="DLzona"
Display="None"
        InitialValue="0"></asp:RequiredFieldValidator>
        <br />
        <asp:RequiredFieldValidator ID="RequiredFieldValidator6"
runat="server"
        ErrorMessage="Elige pinza" ControlToValidate="DLpinza"
Display="None"
        InitialValue="0"></asp:RequiredFieldValidator>
        <br />
        <asp:RequiredFieldValidator ID="RequiredFieldValidator7"
runat="server"
        ErrorMessage="Escribe el tiempo" ControlToValidate="TextBox1"
Display="None" InitialValue="Tiempo (ms)"></asp:RequiredFieldValidator>
        <br />
        <asp:RangeValidator ID="RangeValidator1" runat="server"
        ControlToValidate="TextBox1" Display="None"
        ErrorMessage="500 &lt;= tiempo &lt;= 3000" MaximumValue="3000"
        MinimumValue="500" Type="Integer"></asp:RangeValidator>
        <br />
        <asp:CustomValidator ID="CustomValidator1" runat="server"
        ErrorMessage="El tiempo debe ser múltiplo de 10"
ControlToValidate="TextBox1"
        EnableClientScript="False"
onservervalidate="CustomValidator1_ServerValidate"></asp:CustomValidator>
        <div id="nodostemporal" style="position: absolute; left: 15px; top:475px">
        <asp:GridView ID="GridViewPrimerUsuario" runat="server"
AutoGenerateColumns="False"
        BackColor="White" BorderColor="White" BorderStyle="Ridge"
BorderWidth="2px"
        CellPadding="3" DataSourceID="SqlDataSource4"
GridLines="None"
        HorizontalAlign="Justify" CellSpacing="1" Height="44px">
        <RowStyle BackColor="#DEDFDE" ForeColor="Black" />
        <Columns>
        <asp:BoundField DataField="circunferencia"
HeaderText="circunferencia"

```

```

        SortExpression="circunferencia" >
        </asp:BoundField>
        <asp:BoundField DataField="radio" HeaderText="radio"
SortExpression="radio" />
        <asp:BoundField DataField="plano" HeaderText="plano"
SortExpression="plano" />
        <asp:BoundField DataField="altura" HeaderText="altura"
SortExpression="altura" />
        <asp:BoundField DataField="zona" HeaderText="zona"
SortExpression="zona" />
        <asp:BoundField DataField="pinza" HeaderText="pinza"
SortExpression="pinza" />
        <asp:BoundField DataField="tiempo" HeaderText="tiempo"
SortExpression="tiempo" />

    </Columns>
    <FooterStyle BackColor="#C6C3C6" ForeColor="Black" />
    <PagerStyle BackColor="#C6C3C6" ForeColor="Black"
HorizontalAlign="Right" />
    <SelectedRowStyle BackColor="#9471DE" Font-Bold="True"
ForeColor="White" />
    <HeaderStyle BackColor="#4A3C8C" Font-Bold="True"
ForeColor="#E7E7FF" />

    </asp:GridView>

</div>
<br />
<asp:SqlDataSource ID="SqlDataSource4" runat="server"
    ConnectionString="<%$ ConnectionStrings: usuariosConnectionString %>"
    SelectCommand="SELECT [circunferencia], [radio], [plano], [altura], [zona],
[pinza], [tiempo] FROM [nodos_trayectoria] WHERE ([username] = @username)
ORDER BY [orden]">
    <SelectParameters>
        <asp:ControlParameter ControlID="TextBoxPrimerUsuario"
Name="username"
PropertyName="Text" Type="String" />
    </SelectParameters>
</asp:SqlDataSource>

</form>
</body>

</html>

```



Universitat
Autònoma
de Barcelona



escola
d'enginyeria

**PROYECTO N° 2017. IMPLEMENTACIÓN DE MOVIMIENTOS SUAVES EN ROBOTS
INDUSTRIALES MEDIANTE SPLINES CÚBICOS. APLICACIÓN A UN LABORATORIO
REMOTO DE ROBÓTICA**

ANEXO 3

Anexo del Proyecto Fin de Carrera
De Ingeniería en Informàtica
Realizado por
Marcos Mohedano Gómez
y dirigido por
Asier Ibeas Hernández
Bellaterra, Junio 2010

```

<%@ Application Language="C#" %>
<%@ Import Namespace="System.Data.SqlClient" %>
<script runat="server">

    void Application_Start(object sender, EventArgs e)
    {
        // Código que se ejecuta al iniciarse la aplicación

        //creamos la variable global OCUPADO y la inicializamos a false:
        Application.Add("OCUPADO", "NO");

    }

    void Application_End(object sender, EventArgs e)
    {
        // Código que se ejecuta cuando se cierra la aplicación

    }

    void Application_Error(object sender, EventArgs e)
    {
        // Código que se ejecuta al producirse un error no controlado

    }

    void Session_Start(object sender, EventArgs e)
    {
        // Código que se ejecuta cuando se inicia una nueva sesión

    }

    void Session_End(object sender, EventArgs e)
    {
        // Código que se ejecuta cuando finaliza una sesión.
        // Nota: El evento Session_End se desencadena sólo con el modo
sessionstate
        // se establece como InProc en el archivo Web.config. Si el modo
de sesión se establece como StateServer
        // o SQLServer, el evento no se genera.

    }

</script>

```



Universitat
Autònoma
de Barcelona



escola
d'enginyeria

**PROYECTO N° 2017. IMPLEMENTACIÓN DE MOVIMIENTOS SUAVES EN ROBOTS
INDUSTRIALES MEDIANTE SPLINES CÚBICOS. APLICACIÓN A UN LABORATORIO
REMOTO DE ROBÓTICA**

ANEXO 4

Anexo del Proyecto Fin de Carrera
De Ingeniería en Informàtica
Realizado por
Marcos Mohedano Gómez
y dirigido por
Asier Ibeas Hernández
Bellaterra, Junio 2010

```

<?xml version="1.0"?>
<!--
    Nota: como alternativa para editar manualmente este archivo puede
    utilizar la
    herramienta Administración de sitios web para configurar los valores
    de la aplicación. Utilice
    la opción Sitio web->Configuración de Asp.Net en Visual Studio.
    Encontrará una lista completa de valores de configuración y
    comentarios en
    machine.config.comments, que se encuentra generalmente en
    \Windows\Microsoft.Net\Framework\v2.x\Config
-->
<configuration>
    <configSections>
        <sectionGroup name="system.web.extensions"
type="System.Web.Configuration.SystemWebExtensionsSectionGroup,
System.Web.Extensions, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35">
            <sectionGroup name="scripting"
type="System.Web.Configuration.ScriptingSectionGroup,
System.Web.Extensions, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35">
                <section name="scriptResourceHandler"
type="System.Web.Configuration.ScriptingScriptResourceHandlerSection,
System.Web.Extensions, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35" requirePermission="false"
allowDefinition="MachineToApplication"/>
                <sectionGroup name="webServices"
type="System.Web.Configuration.ScriptingWebServicesSectionGroup,
System.Web.Extensions, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35">
                    <section name="jsonSerialization"
type="System.Web.Configuration.ScriptingJsonSerializationSection,
System.Web.Extensions, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35" requirePermission="false"
allowDefinition="Everywhere"/>
                    <section name="profileService"
type="System.Web.Configuration.ScriptingProfileServiceSection,
System.Web.Extensions, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35" requirePermission="false"
allowDefinition="MachineToApplication"/>
                    <section name="authenticationService"
type="System.Web.Configuration.ScriptingAuthenticationServiceSection,
System.Web.Extensions, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35" requirePermission="false"
allowDefinition="MachineToApplication"/>
                    <section name="roleService"
type="System.Web.Configuration.ScriptingRoleServiceSection,
System.Web.Extensions, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35" requirePermission="false"
allowDefinition="MachineToApplication"/>
                </sectionGroup>
            </sectionGroup>
        </sectionGroup>
    </configSections>
    <appSettings/>
    <connectionStrings>

```

```

        <add name="usuariosConnectionString" connectionString="Data
Source=.\SQLEXPRESS;AttachDbFilename=&quot;C:\AppServ\www\App_Data\usuari
os.mdf&quot;;Integrated Security=True;Connect Timeout=30;User
Instance=True" providerName="System.Data.SqlClient"/>
    </connectionStrings>
    <system.web>
        <!--
simbolos
        Establezca debug="true" en la compilación para insertar
de depuración en la página compilada. Dado que este
proceso afecta al rendimiento, debe establecer este valor
como true
        durante la depuración.
-->
        <authorization>
            <deny users="?" />
        </authorization>
        <membership defaultProvider="proveedorSql">
            <providers>
                <add name="proveedorSql"
type="System.Web.Security.SqlMembershipProvider, System.Web,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
connectionStringName="LocalSqlServer" enablePasswordRetrieval="false"
enablePasswordReset="true" requiresQuestionAndAnswer="false"
applicationName="/" requiresUniqueEmail="false" passwordFormat="Hashed"
maxInvalidPasswordAttempts="5" minRequiredPasswordLength="7"
minRequiredNonalphanumericCharacters="1" passwordAttemptWindow="10"
passwordStrengthRegularExpression="" />
            </providers>
        </membership>
        <compilation debug="true">
            <assemblies>
                <add assembly="System.Core, Version=3.5.0.0,
Culture=neutral, PublicKeyToken=B77A5C561934E089"/>
                <add assembly="System.Web.Extensions,
Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>
                <add assembly="System.Data.DataSetExtensions,
Version=3.5.0.0, Culture=neutral, PublicKeyToken=B77A5C561934E089"/>
                <add assembly="System.Xml.Linq, Version=3.5.0.0,
Culture=neutral, PublicKeyToken=B77A5C561934E089"/>
                <add assembly="System.Web.DynamicData,
Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>
                <add assembly="System.Web.Routing,
Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>
                <add assembly="System.Web.Abstractions,
Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>
                <add
assembly="System.ComponentModel.DataAnnotations, Version=3.5.0.0,
Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>
                <add assembly="System.Data.Linq, Version=3.5.0.0,
Culture=neutral, PublicKeyToken=B77A5C561934E089"/>
                <add assembly="System.Data.Entity,
Version=3.5.0.0, Culture=neutral,
PublicKeyToken=B77A5C561934E089"/></assemblies>
            </compilation>
        <!--
        La sección <authentication> habilita la configuración

```

```

del modo de autenticación de seguridad utilizado por
ASP.NET para identificar a un usuario entrante.
-->
<authentication mode="Forms">
    <forms loginUrl="Login.aspx" protection="All"/>
</authentication>
<!--
La sección <customErrors> habilita la configuración de
las acciones que se deben realizar si un error no controlado
tiene lugar
durante la ejecución de una solicitud. Específicamente,
permite a los desarrolladores configurar páginas de error
html
que se mostrarán en lugar de un seguimiento de pila de
errores.

<customErrors mode="RemoteOnly"
defaultRedirect="GenericErrorPage.htm">
    <error statusCode="403" redirect="NoAccess.htm" />
    <error statusCode="404" redirect="FileNotFound.htm" />
</customErrors>
-->
<pages>
    <controls>
        <add tagPrefix="asp" namespace="System.Web.UI"
assembly="System.Web.Extensions, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35"/>
        <add tagPrefix="asp"
namespace="System.Web.UI.WebControls" assembly="System.Web.Extensions,
Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>
    </controls>
</pages>
<httpHandlers>
    <remove verb="*" path="*.asmx"/>
    <add verb="*" path="*.asmx" validate="false"
type="System.Web.Script.Services.ScriptHandlerFactory,
System.Web.Extensions, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35"/>
    <add verb="*" path="*_AppService.axd" validate="false"
type="System.Web.Script.Services.ScriptHandlerFactory,
System.Web.Extensions, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35"/>
    <add verb="GET,HEAD" path="ScriptResource.axd"
type="System.Web.Handlers.ScriptResourceHandler, System.Web.Extensions,
Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"
validate="false"/>
</httpHandlers>
<httpModules>
    <add name="ScriptModule"
type="System.Web.Handlers.ScriptModule, System.Web.Extensions,
Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>
</httpModules>
</system.web>
<system.codedom>
    <compilers>

```

```

        <compiler language="c#;cs;csharp" extension=".cs"
warningLevel="4" type="Microsoft.CSharp.CSharpCodeProvider, System,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089">
            <providerOption name="CompilerVersion"
value="v3.5"/>
            <providerOption name="WarnAsError" value="false"/>
        </compiler>
        <compiler language="vb;vbs;visualbasic;vbscript"
extension=".vb" warningLevel="4"
type="Microsoft.VisualBasic.VBCodeProvider, System, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089">
            <providerOption name="CompilerVersion"
value="v3.5"/>
            <providerOption name="OptionInfer" value="true"/>
            <providerOption name="WarnAsError" value="false"/>
        </compiler>
    </compilers>
</system.codedom>
<!--
    La sección system.webServer es necesaria para ejecutar ASP.NET
AJAX en Internet
    Information Services 7.0. No es necesaria para la versión
anterior de IIS.
-->
    <system.webServer>
        <validation validateIntegratedModeConfiguration="false"/>
        <modules>
            <remove name="ScriptModule"/>
            <add name="ScriptModule" preCondition="managedHandler"
type="System.Web.Handlers.ScriptModule, System.Web.Extensions,
Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>
        </modules>
        <handlers>
            <remove name="WebServiceHandlerFactory-Integrated"/>
            <remove name="ScriptHandlerFactory"/>
            <remove name="ScriptHandlerFactoryAppServices"/>
            <remove name="ScriptResource"/>
            <add name="ScriptHandlerFactory" verb="*" path="*.asmx"
preCondition="integratedMode"
type="System.Web.Script.Services.ScriptHandlerFactory,
System.Web.Extensions, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35"/>
            <add name="ScriptHandlerFactoryAppServices" verb="*"
path="*_AppService.axd" preCondition="integratedMode"
type="System.Web.Script.Services.ScriptHandlerFactory,
System.Web.Extensions, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35"/>
            <add name="ScriptResource"
preCondition="integratedMode" verb="GET,HEAD" path="ScriptResource.axd"
type="System.Web.Handlers.ScriptResourceHandler, System.Web.Extensions,
Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>
        </handlers>
    </system.webServer>
</runtime>
    <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
        <dependentAssembly>

```

```
        <assemblyIdentity name="System.Web.Extensions"
publicKeyToken="31bf3856ad364e35"/>
        <bindingRedirect oldVersion="1.0.0.0-1.1.0.0"
newVersion="3.5.0.0"/>
    </dependentAssembly>
    <dependentAssembly>
        <assemblyIdentity
name="System.Web.Extensions.Design" publicKeyToken="31bf3856ad364e35"/>
        <bindingRedirect oldVersion="1.0.0.0-1.1.0.0"
newVersion="3.5.0.0"/>
    </dependentAssembly>
</assemblyBinding>
</runtime>
</configuration>
```



Universitat
Autònoma
de Barcelona



escola
d'enginyeria

**PROYECTO N° 2017. IMPLEMENTACIÓN DE MOVIMIENTOS SUAVES EN ROBOTS
INDUSTRIALES MEDIANTE SPLINES CÚBICOS. APLICACIÓN A UN LABORATORIO
REMOTO DE ROBÓTICA**

ANEXO 5

Anexo del Proyecto Fin de Carrera

De Ingeniería en Informàtica

Realizado por

Marcos Mohedano Gómez

y dirigido por

Asier Ibeas Hernández

Bellaterra, Junio 2010

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.Security;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void Login1_Authenticate(object sender,
        AuthenticateEventArgs e)
    {
        if (Membership.ValidateUser(Login1.UserName, Login1.Password)) {
            FormsAuthentication.RedirectFromLoginPage(Login1.
                UserName, false);
        }
    }
}
```



Universitat
Autònoma
de Barcelona



escola
d'enginyeria

**PROYECTO N° 2017. IMPLEMENTACIÓN DE MOVIMIENTOS SUAVES EN ROBOTS
INDUSTRIALES MEDIANTE SPLINES CÚBICOS. APLICACIÓN A UN LABORATORIO
REMOTO DE ROBÓTICA**

ANEXO 6

Anexo del Proyecto Fin de Carrera

De Ingeniería en Informàtica

Realizado por

Marcos Mohedano Gómez

y dirigido por

Asier Ibeas Hernández

Bellaterra, Junio 201

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading;
using System.Drawing;
using System.IO.Ports;
using System.IO;

public partial class laboratorio : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        GridViewPrimerUsuario.Visible = false;
        TextBoxPrimerUsuario.Visible = false;

        SqlConnection conexion = new
        SqlConnection(SqlDataSource1.ConnectionString);
        SqlCommand consulta = new SqlCommand("delete from tanda",
        conexion);
        SqlCommand consulta2 = new SqlCommand("insert into
        tanda(username) values('')", conexion);
        SqlConnection conexion2 = new
        SqlConnection(SqlDataSource2.ConnectionString);
        SqlConnection conexion3 = new
        SqlConnection(SqlDataSource4.ConnectionString);
        SqlCommand consulta3 = new SqlCommand("insert into
        nodos_trayectoria(circunferencia,radio,plano,altura,zona,pinza,tiempo,use
        rname) values('','','','','','','','" + User.Identity.Name.ToString() +
        "')", conexion2);

        if (num_filas_tanda() == 0)
        {
            //si no hay filas en la tanda añadimos una en blanco para que
            sea visible la tabla
            conexion.Open();
            consulta2.ExecuteNonQuery();
            conexion.Close();
            GridView2.DataBind();
        }

        if ((num_filas_tanda() >
        0)&&(GridView2.Rows[0].Cells[0].Text.Trim()!
        == "")&&(GridView2.Rows[0].Cells[0].Text.Trim()!="&nbsp;"))
        {
            TextBoxPrimerUsuario.Text = GridView2.Rows[0].Cells[0].Text;
            GridView2.Rows[0].Cells[0].BackColor = Color.GreenYellow;
        }
        if ((DLcircunferencia.SelectedValue.ToString() == "C3") &&
        (DLplano.SelectedValue.ToString() == "P1"))
        {

```

```

        RequiredFieldValidator5.Enabled = true;
    }
    else
    {
        RequiredFieldValidator5.Enabled = false;
    }
    if (DLzona.SelectedValue.ToString() == "TR")
    {

        RequiredFieldValidator3.Enabled = false;
        RequiredFieldValidator4.Enabled = false;
    }
    else
    {
        RequiredFieldValidator3.Enabled = false;
        RequiredFieldValidator4.Enabled = false;
        RequiredFieldValidator5.Enabled = false;
    }
    if (!IsPostBack)//si es la primera vez que se carga la página:
    {
        TextBoxPrimerUsuario.Text = GridView2.Rows[0].Cells[0].Text;
        Session.Add("idusuario", User.Identity.Name.ToString());//es
necesario añadir esta variable de sesión para poder indicar en el código
ASP del gridview que se cargue con un select haciendo referencia al
username que será la variable de sesión que hemos añadido

    }
    else//si es un postback
    {
        GridViewPrimerUsuario.DataBind();
    }
}

protected void Button1_Click(object sender, EventArgs e)
{
    string comando = obtener_comando();
    if ((Page.IsValid) && (comando != null))//Sólo introducimos el
nuevo nodo en el caso de que pase la validación correctamente y de que
exista dicha posición
    {
        insertar_nodo();
        if (num_apariciones() > 0)
        {
nodo, añadimos
            //al boton 2 la propiedad para que salga el mensaje de
advertencia
            Button2.Attributes.Add("onclick", "alert('No puedes cojer
tanda estando aún en espera')");
        }
        if (num_filas_nodos(User.Identity.Name.ToString())<2)
        {
            //si hay menos de dos nodos además de la posición de reposo
            Button2.Attributes.Add("onclick", "alert('Debes
intrudocir almenos dos nodos')");
        }
    }
    if (obtener_comando() == null)

```

```

        {
            Button1.Attributes.Add("onclick", "alert('Posición no
válida')");
        }
    }
    protected void GridView2_RowCreated(object sender,
GridViewRowEventArgs e)
    {

    }
    protected void TextBox1_TextChanged(object sender, EventArgs e)
    {

    }
    protected void Button2_Click(object sender, EventArgs e)
    {
        if ((num_apariciones() == 0) &&
(num_filas_nodos(User.Identity.Name.ToString()) >= 2))
        {
            //Insertamos el usuario en la tanda de espera sólo en el caso de
que no esté ya en la tanda y de que haya 2 nodos como mínimo en la cola
de nodos:
            insertar_usuario_tanda();//insertamos el usuario
            GridView2.Rows[0].Cells[0].BackColor = Color.GreenYellow;
        }

        if (num_apariciones() > 0){
            Button2.Attributes.Add("onclick", "alert('No puedes cojer
tanda estando aún en espera')");
        }
    }
    protected void Button3_Click(object sender, EventArgs e)
    {
        SqlConnection conexion = new
SqlConnection(SqlDataSource2.ConnectionString);
        SqlCommand comando_delete = new SqlCommand("delete from
nodos_trayectoria where [username]='" + User.Identity.Name.ToString() +
"", conexion);
        SqlCommand comando_insert = new SqlCommand("insert into
nodos_trayectoria(circunferencia,radio,plano,altura,zona,pinza,tiempo,use
rname) values('','','','','','','','" + User.Identity.Name.ToString() +
"')", conexion);
        try
        {
            conexion.Open();
            comando_delete.ExecuteNonQuery();//borramos la lista de nodos
del usuario logueado
            comando_insert.ExecuteNonQuery();//añadimos una fila en
blanco para que se vea la lista de nodos con una fila vacía
            conexion.Close();
            GridViewnodos.DataBind();
            GridViewPrimerUsuario.DataBind();
        }
        catch
        {

        }
    }
}

```

```

        protected void borrar_cola_nodos()
        {
            //borra la cola de nodos perteneciente al primer usuario de la tanda
            SqlConnection conexion = new
            SqlConnection(SqlDataSource2.ConnectionString);
            SqlCommand comando_delete = new SqlCommand("delete from
            nodos_trayectoria where [username]='" + GridView2.Rows[0].Cells[0].Text +
            "'", conexion);
            SqlCommand comando_insert = new SqlCommand("insert into
            nodos_trayectoria(circunferencia,radio,plano,altura,zona,pinza,tiempo,use
            rname) values('','','','','','','','" + GridView2.Rows[0].Cells[0].Text +
            "')", conexion);
            try
            {
                conexion.Open();
                comando_delete.ExecuteNonQuery();//borramos la lista de nodos
                comando_insert.ExecuteNonQuery();//añadimos una fila en
                //del primer usuario de la tanda
                //blanco para que se vea la lista de nodos con una fila vacía
                conexion.Close();
                GridViewnodos.DataBind();
                GridViewPrimerUsuario.DataBind();
            }
            catch
            {
            }
        }

        protected void insertar_usuario_tanda()
        {
            SqlConnection conexion = new
            SqlConnection(SqlDataSource1.ConnectionString);
            SqlCommand comando_insert = new SqlCommand("insert into
            tanda(username) values('" + User.Identity.Name.ToString() + "')",
            conexion);
            SqlCommand comando_delete = new SqlCommand("delete from tanda
            where [username]=''", conexion);
            try
            {
                conexion.Open();
                comando_insert.ExecuteNonQuery();//insertamos el usuario
                comando_delete.ExecuteNonQuery();//eliminamos las filas en
                //blanco que puedan aparecer por error
                conexion.Close();
                GridView2.DataBind();
                TextBoxPrimerUsuario.Text =
                GridView2.Rows[0].Cells[0].Text;//esto sirve para que el
                //gridviewprimerusuario se cargue con los nodos del usuario indicado por el
                //textboxprimerusuario
                GridViewPrimerUsuario.DataBind();//cargamos el gridview
                //temporal con los nodos del primer usuario de la tanda
            }
            catch
            {
                TextBox1.Text = "ERROR";
            }
        }
    }

```

```

protected int num_filas_nodos(string texto)
{
    SqlConnection conexion = new
SqlConnection(SqlDataSource2.ConnectionString);
    SqlCommand comando_count = new SqlCommand("select
count(circunferencia) from nodos_trayectoria where [username]='" + texto
+ "'", conexion);
    conexion.Open();
    int numfilasnodoss = (int)comando_count.ExecuteScalar();//numero
de filas de la cola de nodos para el usuario en cuestion
    conexion.Close();

    return numfilasnodoss;
}

protected int num_filas_nodos_novacias(string texto)
{//devuelve el número de filas no vacías del usuario que indique el
texto.
    SqlConnection conexion = new
SqlConnection(SqlDataSource2.ConnectionString);
    SqlCommand comando_count = new SqlCommand("select
count(circunferencia) from nodos_trayectoria where [username]='" + texto
+ "'AND NOT([circunferencia]='" + null + "'AND [radio]='" + null + "'AND
[plano]='" + null + "'AND [altura]='" + null + "'AND [zona]='" + null +
"'AND [pinza]='" + null + "'AND [tiempo]='" + null + "'", conexion);
    conexion.Open();
    int numfilasnodoss = (int)comando_count.ExecuteScalar();//numero
de filas de la cola de nodos para el usuario en cuestion
    conexion.Close();

    return numfilasnodoss;
}

protected int num_apariciones()
{
    SqlConnection conexion = new
SqlConnection(SqlDataSource1.ConnectionString);
    SqlCommand comando_count= new SqlCommand("select count(username)
from tanda where [username]='" + User.Identity.Name.ToString() + "'",
conexion);
    conexion.Open();
    int apariciones = (int)comando_count.ExecuteScalar();//num de
veces que aparece el usuario en la tanda
    conexion.Close();
    return apariciones;
}

protected int num_filas_tanda()
{
    SqlConnection conexion = new
SqlConnection(SqlDataSource1.ConnectionString);
    SqlCommand comando_count = new SqlCommand("select count(username)
from tanda", conexion);
    conexion.Open();
    int numfilastanda = (int)comando_count.ExecuteScalar();//num de
filas de la tanda de usuarios
    conexion.Close();
}

```

```

        return numfilastanda;
    }

    protected void insertar_nodo()
    {
        SqlConnection conexion = new
        SqlConnection(SqlDataSource2.ConnectionString);
        SqlCommand comando_insert = new SqlCommand("insert into
        nodos_trayectoria(circunferencia,radio,plano,altura,zona,pinza,tiempo,use
        rname) values('" + DLcircunferencia.SelectedValue.ToString() + "','" +
        DLradio.SelectedValue.ToString() + "','" +
        DLplano.SelectedValue.ToString() + "','" +
        DLaltura.SelectedValue.ToString() + "','" +
        DLzona.SelectedValue.ToString() + "','" +
        DLpinza.SelectedValue.ToString() + "','" +
        (Convert.ToInt32(TextBox1.Text)).ToString() + "','" +
        User.Identity.Name.ToString() + "')", conexion);
        SqlCommand comando_delete = new SqlCommand("delete from
        nodos_trayectoria where [circunferencia]=''", conexion);
        SqlCommand comando_insert2 = new SqlCommand("insert into
        nodos_trayectoria(circunferencia,radio,plano,altura,zona,pinza,tiempo,use
        rname) values('" + DLcircunferencia.SelectedValue.ToString() + "','" +
        DLradio.SelectedValue.ToString() + "','" +
        DLplano.SelectedValue.ToString() + "','" +
        DLaltura.SelectedValue.ToString() + "','" + null + "','" +
        DLpinza.SelectedValue.ToString() + "','" +
        (Convert.ToInt32(TextBox1.Text)).ToString() + "','" +
        User.Identity.Name.ToString() + "')", conexion);
        SqlCommand comando_insert3 = new SqlCommand("insert into
        nodos_trayectoria(circunferencia,radio,plano,altura,zona,pinza,tiempo,use
        rname) values('" + DLcircunferencia.SelectedValue.ToString() + "','" +
        DLradio.SelectedValue.ToString() + "','" + null + "','" + null + "','" +
        DLzona.SelectedValue.ToString() + "','" +
        DLpinza.SelectedValue.ToString() + "','" +
        (Convert.ToInt32(TextBox1.Text)).ToString() + "','" +
        User.Identity.Name.ToString() + "')", conexion);
        try
        {
            conexion.Open();
            if (DLzona.SelectedValue.ToString() == "0")
            {
                comando_insert2.ExecuteNonQuery();//insertamos un nuevo
                nodo pero con valor NULL en el campo [Zona]
            }
            else if ((DLplano.SelectedValue.ToString() == "0") &&
            (DLaltura.SelectedValue.ToString() == "0"))
            {
                comando_insert3.ExecuteNonQuery();//insertamos un nuevo
                nodo pero con valor NULL en los campos [plano] y [altura]
            }
            else
            {
                comando_insert.ExecuteNonQuery();//insertamos un nuevo
                nodo con todos los valores seleccionados de los dropdownlists
            }
            comando_delete.ExecuteNonQuery();//eliminamos las filas en
            blanco que puedan aparecer por error
        }
    }
}

```

```

        conexion.Close();
        GridViewnodos.DataBind();
        GridViewPrimerUsuario.DataBind();
    }
    catch
    {
    }
}

protected void Timer1_Tick(object sender, EventArgs e)
{
    if(Application["OCUPADO"].ToString() == "NO")
    {
        if(num_filas_tanda() > 0)
        {
            if ((GridView2.Rows[0].Cells[0].Text.Trim() !=
"&nbsp;") && (GridView2.Rows[0].Cells[0].Text.Trim() != ""))
                { //Si hay usuarios en la tanda empezamos la ejecución del
código del primer usuario:
                    Application["OCUPADO"] = "SI";
                    Thread hilo = new Thread(new
ThreadStart(ejecutar_comandos));
                    hilo.Start();
                }
        }
    }
}

protected void ejecutar_comandos()
{ //ponemos esto en un thread aparte para que el sleep no paralice
toda la web
    string primerusuario =
GridView2.Rows[0].Cells[0].Text.ToString();
    SqlConnection conexion = new
SqlConnection(SqlDataSource1.ConnectionString);
    SqlCommand comando_delete = new SqlCommand("delete from tanda
where [username]='" + primerusuario + "'", conexion);
    string[] vcomandos = new
string[num_filas_nodos(GridView2.Rows[0].Cells[0].Text) + 1];
    int[] vtiempos = new
int[num_filas_nodos(GridView2.Rows[0].Cells[0].Text) + 1];
    try
    {
        rellenar_vectores(vcomandos, vtiempos); //rellenamos el vector
de comandos y el de tiempos
        enviar_splines(vcomandos, vtiempos);
        borrarCola_nodos(); //borramos la cola de nodos sólo cuando
no necesitemos nunca más esos datos
        Application["OCUPADO"] = "NO";
        conexion.Open();
        comando_delete.ExecuteNonQuery(); //borramos el primer
elemento de la tanda
        conexion.Close();
        GridView2.DataBind();
        GridViewPrimerUsuario.DataBind();
    }
}

```

```

    }
    catch
    {
    }

}

protected void rellenar_vectores(string[] v_comando,int[] v_tiempo)
{
    //recorremos el gridview de los nodos y buscamos el comando sc-32
    equivalente a cada nodo
    //y rellenamos el vector de tiempos:
    v_tiempo[0] = 0;
    v_comando[0] = equivalente("reposo");
    int filasnodos =
num_filas_nodos(GridView2.Rows[0].Cells[0].Text);
    for (int i=1; i <= filasnodos; i++)
    {
        v_comando[i]=obtener_comando(i-1);
        v_tiempo[i] = v_tiempo[i - 1] +
int.Parse(GridViewPrimerUsuario.Rows[i - 1].Cells[6].Text);
    }
}

protected string equivalente(string cadena)
{
    SqlConnection conexion = new
SqlConnection(SqlDataSource3.ConnectionString);
    SqlCommand comando = new SqlCommand("select [comando_sc32] from
equivalencias where botones='"+cadena+"'", conexion);
    conexion.Open();
    string resultado = (string)comando.ExecuteScalar();
    conexion.Close();
    return resultado;
}

protected string obtener_comando()
{
    //obtiene el comando sc-32 equivalente al nodo que indican los
    dropdownlists, pero sin indicar la pinza
    string nodo,comando = "";

    if (DLzona.SelectedValue.ToString() == "0")
    {
        nodo = DLcircunferencia.SelectedValue.ToString() +
DLradio.SelectedValue.ToString() + DLplano.SelectedValue.ToString() +
DLaltura.SelectedValue.ToString();
    }
    else if ((DLplano.SelectedValue.ToString() == "0") &&
(DLaltura.SelectedValue.ToString() == "0"))
    {
        nodo = DLcircunferencia.SelectedValue.ToString() +
DLradio.SelectedValue.ToString() + DLzona.SelectedValue.ToString();
    }
    else
    {

```

```

        nodo = DLcircunferencia.Selected.Value.ToString() +
        DLradio.Selected.Value.ToString() + DLplano.Selected.Value.ToString() +
        DLaltura.Selected.Value.ToString() + DLzona.Selected.Value.ToString();
    }

    comando = equivalente(nodo);

    return comando;
}

protected string obtener_comando(int i)
{
    //obtiene el comando sc32 equivalente al nodo de la fila indicada
    del gridViewprimerusuario
    string nodo, pinza, comando = "";

    nodo = GridViewPrimerUsuario.Rows[i].Cells[0].Text +
    GridViewPrimerUsuario.Rows[i].Cells[1].Text +
    GridViewPrimerUsuario.Rows[i].Cells[2].Text +
    GridViewPrimerUsuario.Rows[i].Cells[3].Text +
    GridViewPrimerUsuario.Rows[i].Cells[4].Text;
    string temp = nodo.Replace("&nbsp;", "");
    pinza = GridViewPrimerUsuario.Rows[i].Cells[5].Text;
    comando = equivalente(temp) + equivalente(pinza);
    return comando;
}

protected void enviar_splines(string[] v_comandos, int[] v_tiempos)
{
    //creamos la instancia del puerto serie con la configuración
    adecuada para el circuito ssc-32:
    SerialPort puerto = new SerialPort("COM1", 115200, Parity.None,
    8, StopBits.One);

    //creamos un vector de q's para cada servo:
    int[] qservo0 = new int[v_tiempos.Length];
    int[] qservo1 = new int[v_tiempos.Length];
    int[] qservo2 = new int[v_tiempos.Length];
    int[] qservo3 = new int[v_tiempos.Length];
    int[] qservo4 = new int[v_tiempos.Length];
    int[] qservo5 = new int[v_tiempos.Length];

    //rellenamos dichos vectores con las posiciones de cada
    servo, obteniéndolas del vector de comandos:
    for (int i = 0; i < v_comandos.Length; i++)
    {
        string comando = v_comandos[i]; //obtenemos el comando
        //introducimos cada posición del comando en su servo
        correspondiente:
        string[] posiciones_servos = comando.Split('#'); //ahora
        tendremos el vector posiciones_servos, donde la posición [0] es la del
        servo 0, la[1] del servo 1, etc.

        //a cada posición de cada servo le borramos la subcadena
        "0P", "1P", etc. y la colocamos en la posición correspondiente del vector
        de posiciones de cada servo en cada instante de tiempo:
        qservo0[i] = int.Parse(posiciones_servos[1].Remove(0, 2));
    }
}

```

```

        if (posiciones_servos.Length > 2)
        {
            qservo1[i] = int.Parse(posiciones_servos[2].Remove(0,
2));
            qservo2[i] = int.Parse(posiciones_servos[3].Remove(0,
2));
            qservo3[i] = int.Parse(posiciones_servos[4].Remove(0,
2));
            qservo4[i] = int.Parse(posiciones_servos[6].Remove(0,
2));
            qservo5[i] = int.Parse(posiciones_servos[5].Remove(0,
2));
        }
    }

    int n = v_tiempos.Length;
    double[] zservo0 = new double[n + 1];
    double[] hservo0 = new double[n];
    double[] zservo1 = new double[n + 1];
    double[] hservo1 = new double[n];
    double[] zservo2 = new double[n + 1];
    double[] hservo2 = new double[n];
    double[] zservo3 = new double[n + 1];
    double[] hservo3 = new double[n];
    double[] zservo4 = new double[n + 1];
    double[] hservo4 = new double[n];
    double[] zservo5 = new double[n + 1];
    double[] hservo5 = new double[n];
    double spline_i0, spline_i1, spline_i2, spline_i3, spline_i4,
spline_i5 = 0;

    //creamos las velocidades:
    velocidades(n, v_tiempos, qservo0, zservo0, hservo0); //esto
rellena los vectores z y h
    velocidades(n, v_tiempos, qservo1, zservo1, hservo1);
    velocidades(n, v_tiempos, qservo2, zservo2, hservo2);
    velocidades(n, v_tiempos, qservo3, zservo3, hservo3);
    velocidades(n, v_tiempos, qservo4, zservo4, hservo4);
    velocidades(n, v_tiempos, qservo5, zservo5, hservo5);

    for (int i = v_tiempos[0] ; i < v_tiempos[n-1] ; i+=10 )
    { //para cada servo calculamos el spline cúbico perteneciente al
instante i (es decir cada 10 ms)
        spline_i0 = splinecubico(n, v_tiempos, qservo0, zservo0,
hservo0, i);
        spline_i1 = splinecubico(n, v_tiempos, qservo1, zservo1,
hservo1, i);
        spline_i2 = splinecubico(n, v_tiempos, qservo2, zservo2,
hservo2, i);
        spline_i3 = splinecubico(n, v_tiempos, qservo3, zservo3,
hservo3, i);
        spline_i4 = splinecubico(n, v_tiempos, qservo4, zservo4,
hservo4, i);
        spline_i5 = splinecubico(n, v_tiempos, qservo5, zservo5,
hservo5, i);
        //reconstruimos el comando que indica las posiciones de los 5
servos en el instante i:

```

```

        string comando = "#0P" + ((int)spline_i0).ToString() + "#1P"
+ ((int)spline_i1).ToString() + "#2P" + ((int)spline_i2).ToString() +
"#3P" + ((int)spline_i3).ToString() + "#4P" + ((int)spline_i4).ToString()
+ "#5P" + ((int)spline_i5).ToString();
        try
        {
            puerto.Open();
            puerto.WriteLine(comando + "T10" +
Convert.ToChar(13)); //enviamos al COM1 el comando indicándole tiempo 10
ms
            puerto.Close();
            Thread.Sleep(10); //hacemos un sleep de 10 ms
        }

        catch
        {
        }

    }

    //Tras finalizar la trayectoria con splines, enviamos al robot a
la posición de reposo:
    Thread.Sleep(5000); //mantenemos 5 segundos la última posición de
la trayectoria antes de mandarlo a la posición de reposo
    puerto.Open();
    puerto.WriteLine(equivalente("reposo")+"T5000"+
Convert.ToChar(13));
    puerto.Close();
    Thread.Sleep(5000);
}

protected void velocidades(int n, int[] t, int[] q, double[] z,
double[] h)
{
    //Calcula las incógnitas zi del sistema de ecuaciones, es decir,
    //las velocidades en los puntos de paso

    double[] b = new double[n];
    double[] u = new double[n - 1];
    double[] v = new double[n - 1];

    for (int i = 0; i < n - 1; i++)
    {
        h[i] = t[i + 1] - t[i];
        b[i] = 6 * (q[i + 1] - q[i]) / h[i];
    }

    u[1] = 2 * (h[0] + h[1]);
    v[1] = b[1] - b[0];
    for (int i = 2; i < n - 1; i++)
    {
        u[i] = 2 * (h[i] + h[i - 1]) - (double)Math.Pow((h[i - 1]),
2) / u[i - 1];
        v[i] = b[i] - b[i - 1] - h[i - 1] * v[i - 1] / u[i - 1];
    }

    z[n] = 0;
    for (int i = n - 2; i > 0; i--)
    {
        //calculamos las velocidades:
        z[i] = (v[i] - h[i] * z[i + 1]) / u[i];
    }
}

```

```

    }
    z[0] = 0;
}

protected double splinecubico(int n, int[] t, int[] q, double[] z,
double[] h, double x)
{
    //calcula el spline perteneciente al intervalo
    //de tiempo al que pertenezca "x"
    double Ai;
    double Bi;
    double Ci;
    double Sx = 0;

    for (int i = 0; i < n - 1; i++)
    { //busca a que intervalo de tiempo pertenece x y entonces
calcula el spline evaluado en x
        if ((t[i] < x) && (x < t[i + 1]))
        {
            Ai = (z[i + 1] - z[i]) / (6 * h[i]);
            Bi = z[i] / 2;
            Ci = -(h[i] * z[i + 1]) / 6 - (h[i] * z[i]) / 3 + (q[i +
1] - q[i]) / h[i];

            Sx = q[i] + (x - t[i]) * (Ci + (x - t[i]) * (Bi + (x -
t[i]) * Ai));
        }
        else if (x == t[i])
        {
            Sx = q[i];
        }
        else if (x == t[i + 1])
        {
            Sx = q[i + 1];
        }
        else
        {
        }
    }
    return Sx;
}

protected void CustomValidator1_ServerValidate(object source,
ServerValidateEventArgs args)
{
    if (int.Parse(args.Value) % 10 == 0)
    {
        args.IsValid = true;
    }
    else
    {
        args.IsValid = false;
        Button1.Attributes.Add("onclick", "alert('El tiempo debe ser
múltiplo de 10')");
    }
}

protected void botonsubir_Click(object sender, EventArgs e)

```

```

{
    if (num_apariciones() > 0)
    {
        botonsubir.Attributes.Add("onclick", "alert('No puedes cojer
tanda estando aún en espera')");
    }
    else if ((num_apariciones() ==
0)&&(num_filas_nodos_novacias(User.Identity.Name.ToString())==0))//sólo
subimos el archivo en el caso de no estar ya en la tanda y que no haya
nodos en la cola de nodos del usuario logeado:
    {
        if (num_filas_nodos_novacias(User.Identity.Name.ToString()) >
0)
        {
            Labell1.Text = "Antes de subir el archivo, borra la cola
de nodos.";
        }
        string path = @"C:\";
        if (FileUpload1.HasFile)
        {
            var info = new FileInfo(path + FileUpload1.FileName);
            if (info.Extension == ".txt")
            {
                try
                {
                    if (File.Exists(path +
User.Identity.Name.ToString() + ".txt"))
                    {
                        File.Delete(path +
User.Identity.Name.ToString() + ".txt");
                    }
                    FileUpload1.SaveAs(path + FileUpload1.FileName);
                    File.Move(path + FileUpload1.FileName, path +
User.Identity.Name.ToString() + ".txt");//hacemos una copia del archivo
original renombrándolo con el nombre del usuario logeado
                    File.Delete(path +
FileUpload1.FileName);//borramos el archivo original
                }
                catch
                {
                    Labell1.Text = "Error al subir el archivo.
Inténtelo de nuevo.";
                }
                if (validar_texto(path +
User.Identity.Name.ToString() + ".txt") == true)
                {
                    //si pasa la validación:
                    //insertamos en la tanda de usuarios el usuario
que ha enviado el archivo:
                    if ((num_apariciones() == 0) &&
(num_lineas_fichero(path + User.Identity.Name.ToString() + ".txt") >= 3))
                    {
                        //Insertamos el usuario en la tanda de espera
sólo en el caso de que no esté ya en la tanda y de que haya 2 nodos como
mínimo en el fichero:
                        insertar_usuario_tanda();//insertamos el
usuario
                        GridView2.Rows[0].Cells[0].BackColor =
Color.GreenYellow;
                    }
                }
            }
        }
    }
}

```

```

        Label1.Text = "Archivo subido con éxito";
        transcribir_fichero(path +
User.Identity.Name.ToString() + ".txt");//pasamos los datos del fichero a
la tabla de nodos correspondiente al usuario logeado
    }
    else if (num_lineas_fichero(path +
User.Identity.Name.ToString() + ".txt") < 3)
    {
        Label1.Text = "Error: el archivo ha de
contener como mínimo 2 nodos";
        File.Delete(path +
User.Identity.Name.ToString() + ".txt");//eliminamos el archivo
    }

    }
    else
    {
        //si no pasa la validación:
        Label1.Text = "Error: posiciones no válidas o
sintaxis incorrecta";
        File.Delete(path + User.Identity.Name.ToString()
+ ".txt");//eliminamos el archivo
    }
    if (File.Exists(path + FileUpload1.FileName))
    {
        File.Delete(path +
FileUpload1.FileName);//borramos el archivo original
    }

    }
    else
    {
        Label1.Text = "ERROR: El archivo debe ser .txt";
    }
}
else
{
    Label1.Text = "No hay ningún archivo seleccionado";
}
}
}

protected bool validar_texto(string path)
{
    //valida el texto comprobando que todas las posiciones existen
    bool resultado = true;
    StreamReader sr = new StreamReader(path);
    SqlConnection conexion = new
SqlConnection(SqlDataSource3.ConnectionString);
    string linea = sr.ReadLine();
    while ((linea != "END;") && (resultado == true))
    {
        if ((linea == "") || (linea == null))
        {
            sr.Close();
            return false;
        }
        string[] partes = linea.Split(',');//separamos el comando en
dos partes,entendiendo el caracter ',' como separador
    }
}

```

```

        string posicion = "";
        string tiempo = "";
        if (partes.Length > 1)
        {
            posicion = partes[0];
            tiempo = partes[1].Replace(";", ""); //eliminamos el
caracter ";"
        }
        else
        {
            sr.Close();
            return false;
        }
        if (posicion.StartsWith("Pos"))
        {
            posicion=posicion.Replace("Pos", ""); //eliminamos la
cabecera "Pos"
        }
        else if (posicion.StartsWith("Gripp")) {
            posicion="Abierta";
        }
        else if (posicion.StartsWith("UNGripp")) {
            posicion="Cerrada";
        }
        //BUSCAR en la tabla equivalencias si existe la posicion, y
si no es así retornamos false:
        SqlCommand comando_count = new SqlCommand("select
count(botones)from equivalencias where [botones]='" + posicion + "'",
conexion);
        conexion.Open();
        int contador = (int)comando_count.ExecuteScalar();
        conexion.Close();
        resultado = (contador > 0) && (int.Parse(tiempo)
%10==0) && (int.Parse(tiempo)>=500) && (int.Parse(tiempo)<=3000); //devolverem
os true cuando exista la posición en la tabla equivalencias y cuando el
tiempo sea múltiplo de 10 y esté entre 500 y 3000.
        linea = sr.ReadLine();
    }
    sr.Close();
    return resultado;
}

private int num_lineas_fichero(string path)
{
    int contador = 0;
    StreamReader sr = new StreamReader(path);
    while (sr.ReadLine() != "END;")
    {
        contador ++;
    }
    sr.Close();
    return contador;
}

private void transcribir_fichero(string path)
{ //coloca los nodos del fichero en la tabla nodos_trayectoria
correspondiente al usuario logeado

```

```

        StreamReader sr = new StreamReader(path);
        string linea = sr.ReadLine();
        string pinza = "Abierta";

        string circunferencia = null;
        string radio=null;
        string plano= null;
        string altura = null;
        string zona = null;
        while (linea != "END;")
        {
            SqlConnection conexion = new
SqlConnection(SqlDataSource2.ConnectionString);
            SqlCommand comando_delete = new SqlCommand("delete from
nodos_trayectoria where
[circunferencia]=''AND[radio]=''AND[plano]=''AND[altura]=''AND[zona]=''AND
D[pinza]=''AND[tiempo]=''", conexion);
            string[] partes = linea.Split(',');//separamos el comando en
dos partes,entendiendo el caracter ',' como separador
            string posicion = partes[0];
            string tiempo = partes[1].Replace(";", "");//eliminamos el
caracter ";"

            if (posicion.StartsWith("Pos"))
            {
                posicion = posicion.Replace("Pos", "");//eliminamos la
cabecera "Pos"

                if ((posicion.EndsWith("Z1")) || (posicion.EndsWith("Z2"))){
                    //-colocar en la tabla nodos_trayectoria los
substrings equivalentes a la Circunferencia,radio,plano,altura y zona.
                    //-colocar también el tiempo en la tabla.
                    //-colocar en el campo pinza el valor actual del
string pinza.

                    circunferencia = posicion.Substring(0, 2);
                    radio = posicion.Substring(2, 2);
                    plano = posicion.Substring(4, 2);
                    altura = posicion.Substring(6, 1);
                    zona = posicion.Substring(7, 2);
                    SqlCommand comando_insertar=new SqlCommand("insert
into
nodos_trayectoria(circunferencia,radio,plano,altura,zona,pinza,tiempo,use
rname) values('" + circunferencia + "','" + radio + "','" + plano + "','"
+ altura + "','" + zona + "','" + pinza + "','" + tiempo + "','" +
User.Identity.Name.ToString() + "')", conexion);
                    conexion.Open();
                    comando_insertar.ExecuteNonQuery();
                    conexion.Close();
                }
                else if(posicion.EndsWith("TR")){
                    //-colocar en la tabla nodos_trayectoria la
circunferencia, radio y zona, y colocar valores NULL en el resto
                    //-colocar en el campo pinza de nodos_trayectoria el
valor actual del string pinza.
                    //-colocar el tiempo en nodos_trayectoria.
                    circunferencia = posicion.Substring(0, 2);
                    radio = posicion.Substring(2, 2);

```

```

        plano = null;
        altura = null;
        zona = posicion.Substring(4, 2);
        SqlCommand comando_insertar = new SqlCommand("insert
into
nodos_trayectoria(circunferencia,radio,plano,altura,zona,pinza,tiempo,use
rname) values('" + circunferencia + "','" + radio + "','" + plano + "','"
+ altura + "','" + zona + "','" + pinza + "','" + tiempo + "','" +
User.Identity.Name.ToString() + "')", conexion);
        conexion.Open();
        comando_insertar.ExecuteNonQuery();
        conexion.Close();
    }
    else if ((posicion.EndsWith("H")) ||
(posicion.EndsWith("L")))
    {
        //-colocar en nodos_trayectoria la
circunferencia,radio,plano y altura, y null en el campo zona.
        //-colocar en nodos_trayectoria el tiempo.
        //-colocar en nodos_trayectoria el valor del string
pinza en el campo pinza.
        circunferencia = posicion.Substring(0, 2);
        radio = posicion.Substring(2, 2);
        plano = posicion.Substring(4, 2);
        altura = posicion.Substring(6, 1);
        zona = null;
        SqlCommand comando_insertar = new SqlCommand("insert
into
nodos_trayectoria(circunferencia,radio,plano,altura,zona,pinza,tiempo,use
rname) values('" + circunferencia + "','" + radio + "','" + plano + "','"
+ altura + "','" + zona + "','" + pinza + "','" + tiempo + "','" +
User.Identity.Name.ToString() + "')", conexion);
        conexion.Open();
        comando_insertar.ExecuteNonQuery();
        conexion.Close();
    }
}
else if(posicion.StartsWith("Gripp")){
    pinza = "Cerrada";
    //-colocar valores null en todos los campos de
nodos_trayectoria excepto en el campo pinza donde colocamos el string
pinza
    //-colocar el tiempo en nodos_trayectoria.

    SqlCommand comando_insertar = new SqlCommand("insert into
nodos_trayectoria(circunferencia,radio,plano,altura,zona,pinza,tiempo,use
rname) values('" + circunferencia + "','" + radio + "','" + plano + "','"
+ altura + "','" + zona + "','" + pinza + "','" + tiempo + "','" +
User.Identity.Name.ToString() + "')", conexion);
    conexion.Open();
    comando_insertar.ExecuteNonQuery();
    conexion.Close();
}
else if (posicion.StartsWith("UNGripp"))
{
    pinza = "Abierta";

```

```

        //-colocar valores null en todos los campos de
        nodos_trayectoria excepto en el campo pinza donde colocamos el string
        pinza

        //-colocar el tiempo en nodos_trayectoria.

        SqlCommand comando_insertar = new SqlCommand("insert into
        nodos_trayectoria(circunferencia,radio,plano,altura,zona,pinza,tiempo,use
        rname) values('" + circunferencia + "','" + radio + "','" + plano + "','"
        + altura + "','" + zona + "','" + pinza + "','" + tiempo + "','" +
        User.Identity.Name.ToString() + "','', conexion);
        conexion.Open();
        comando_insertar.ExecuteNonQuery();
        conexion.Close();
    }
    conexion.Open();
    comando_delete.ExecuteNonQuery();//eliminamos las filas en
    blanco que puedan aparecer por error
    conexion.Close();
    GridViewnodos.DataBind();
    GridViewPrimerUsuario.DataBind();

    linea = sr.ReadLine();
}

sr.Close();
}

protected void ImageButton1_Click(object sender, ImageClickEventArgs
e)
{
    string filename = "manual.pdf";
    descargar_archivo(filename);
}

protected void ImageButton4_Click(object sender, ImageClickEventArgs
e)
{
    string filename = "verificador.exe";
    descargar_archivo(filename);
}

protected void descargar_archivo(string filename)
{
    String dlDir = @"documentos/";
    String path = Server.MapPath(dlDir + filename);

    System.IO.FileInfo toDownload = new System.IO.FileInfo(path);

    if (toDownload.Exists)
    {
        Response.Clear();
        Response.AddHeader("Content-Disposition", "attachment;
filename=" + toDownload.Name);
        Response.AddHeader("Content-Length",
toDownload.Length.ToString());
        Response.ContentType = "application/octet-stream";
        Response.WriteFile(dlDir + filename);
    }
}

```

```
        Response.End();  
    }  
}  
}
```



Universitat
Autònoma
de Barcelona



**PROYECTO N° 2017. IMPLEMENTACIÓN DE MOVIMIENTOS SUAVES EN
ROBOTS INDUSTRIALES MEDIANTE SPLINES CÚBICOS. APLICACIÓN A UN
LABORATORIO REMOTO DE ROBÓTICA**

ANEXO 7

Anexo del Proyecto Fin de Carrera
De Ingeniería en Informàtica

Realizado por

Marcos Mohedano Gómez

y dirigido por

Asier Ibeas Hernández

Bellaterra, Junio 2010



Universitat
Autònoma
de Barcelona



escola
d'enginyeria

**Laboratorio remoto de Robótica:
Implementación de movimientos suaves
en robots industriales mediante *splines*
cúbicos.**

Manual de usuario

ÍNDICE

1. PÁGINA WEB.....	1
2. ROBOT.....	4
3. ENVIAR TRAYECTORIAS.....	8
3.1. A TRAVÉS DE UN ARCHIVO DE TEXTO.....	8
3.2. DE FORMA MANUAL.....	10

1. PÁGINA WEB

Los alumnos de la asignatura Robótica y Automatización Industrial pueden interactuar con un robot manipulador desde cualquier punto con conexión a internet gracias a la web del laboratorio remoto de robótica.

El usuario, una vez registrado y autenticado en la web, podrá enviar al servidor una serie de coordenadas en el espacio cartesiano que indican los puntos por donde desea que pase el efector final del robot. Cuando el robot se encuentre disponible, ejecutará una trayectoria que pase por dichos puntos (a los cuales a partir de ahora nos referiremos como *nodos*) mediante la interpolación por *splines* cúbicos. De esta manera los movimientos del robot serán lo más suaves posibles. El alumno podrá observar en la interfaz web los movimientos del robot en directo gracias al *streaming* de vídeo capturado por dos cámaras que enfocan al robot desde distintos ángulos.

La URL del laboratorio remoto es la siguiente:

`http://[ip_servidor]/laboratorio.aspx`

Donde [ip_servidor] es la dirección IP del servidor donde se aloja la web.

Nota: Si se desea acceder a la web desde fuera de la red de la UAB, consultar en:

<http://www.uab.es/servlet/Satellite?cid=1096478607689&pagename=i-UAB%2FPage%2FTemplateGenericHeaderSite>

Una vez el alumno ha introducido la URL en el navegador, aparece la página de Login/Creación de nueva cuenta, que tiene el siguiente aspecto (ver figura 1):

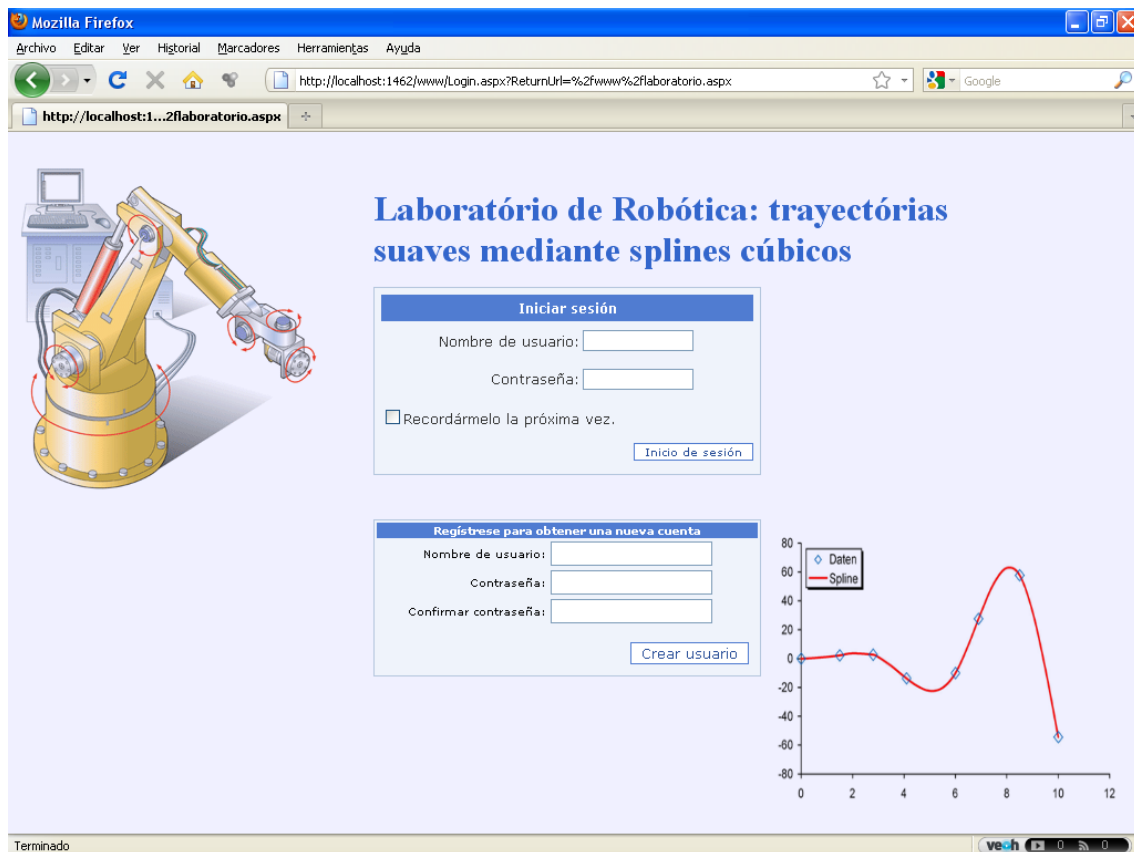


Figura 1 - Página de Login

Si se trata de un nuevo usuario, hay que rellenar los campos del cuadro de registro, indicando el nombre de usuario y dos veces la contraseña deseada. Si el usuario ya tenía cuenta, se logea en el cuadro de inicio de sesión.

Una vez iniciada la sesión en una cuenta de usuario, aparece la siguiente página (ver figura 2):

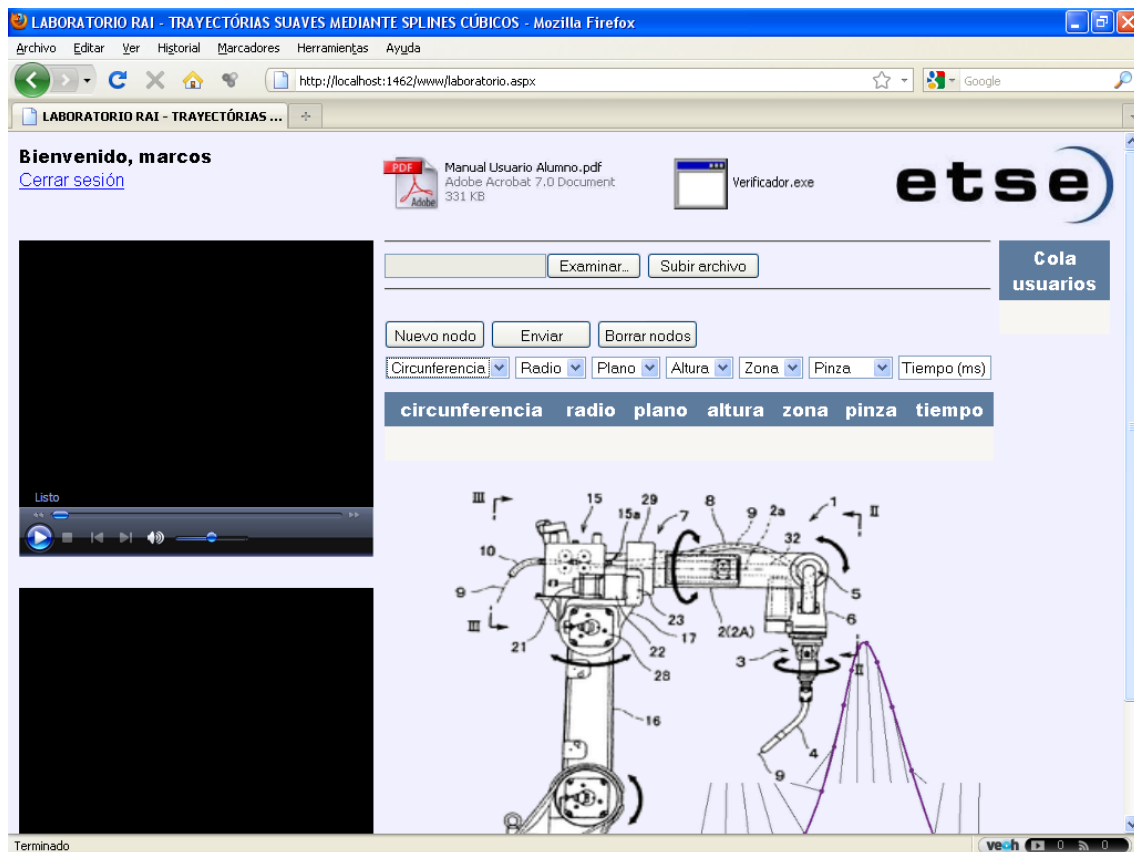


Figura 2 – Página del laboratorio

Es en esta página donde el usuario puede enviar los nodos (hay dos métodos de envío como se verá en el siguiente capítulo) al servidor y esperar a que su *nick* se encuentre en primera posición de la cola de usuarios que se observa a la derecha de la imagen. Entonces podrá ver a través de las dos ventanas del reproductor de *Windows Media Player* incrustadas a la izquierda de la interfaz, en directo, cómo el robot efectúa la trayectoria que pasa por los nodos introducidos.

2. ROBOT

Se dispone del brazo robot *Lynx6*, de la marca *Lynxmotion*. Este se ha montado sobre una plataforma en la cual se han señalado una serie de semicircunferencias y radios que definen parte del espacio de trabajo del manipulador.

Se han definido una serie de puntos característicos dentro del espacio de trabajo del robot. Podemos hacer referencia a dichos puntos mediante el uso de unas etiquetas predefinidas.

En la siguiente figura tenemos una visión vertical del esquema de puntos característicos (Ver Figura 3):

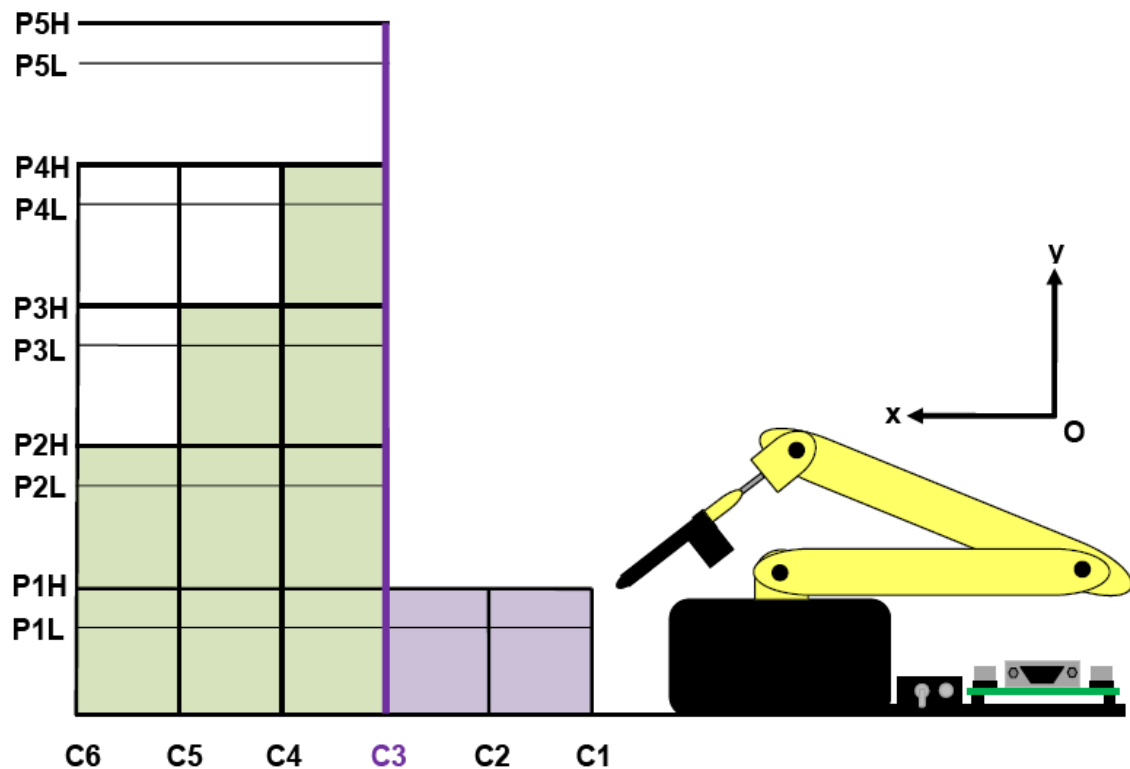


Figura 3 –Vista vertical del espacio de trabajo del robot

Y en la figura 4 se muestra la vista horizontal de la distribución de los puntos característicos (ver figura 4).

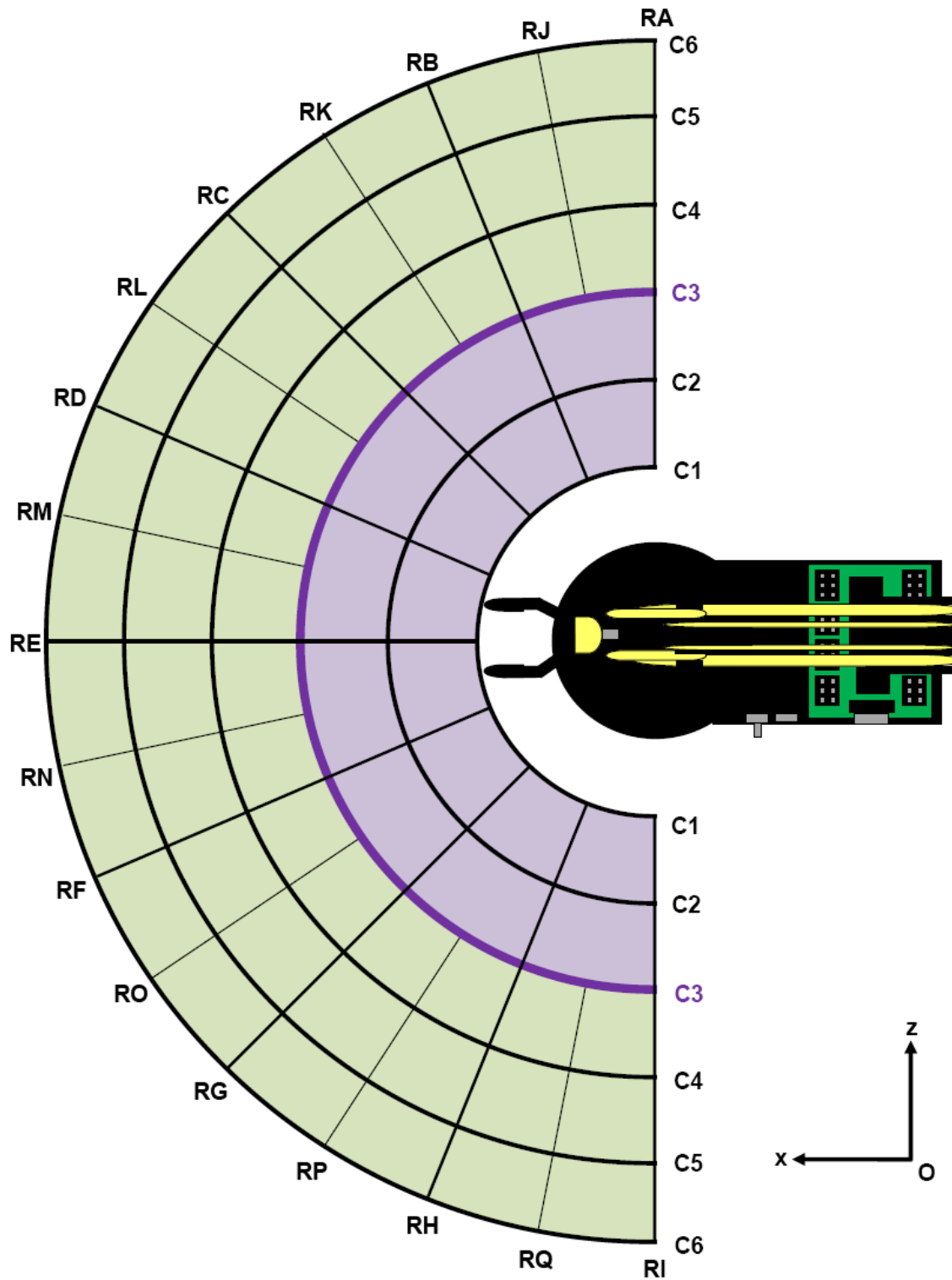


Figura 4 – Vista Horizontal

Existen dos zonas de trabajo distintas dentro del espacio de trabajo del robot, en las cuales el robot adquiere una configuración (inclinación) distinta para situarse sobre los puntos característicos y así poder coger los objetos cerrando la pinza.

La “Zona 1” está identificada sobre el esquema mediante el color lila, y en esta zona están contenidos los radios “RA, RB, RC, RD, RE, RF, RG, RH y RI”, las semicircunferencias “C1, C2 y C3” y el Plano 1 con su parte baja y alta “P1L y P1H”. La configuración que adquiere el robot para situarse sobre los puntos característicos en esta zona, es como la que se observa en la siguiente figura (ver figura 5):

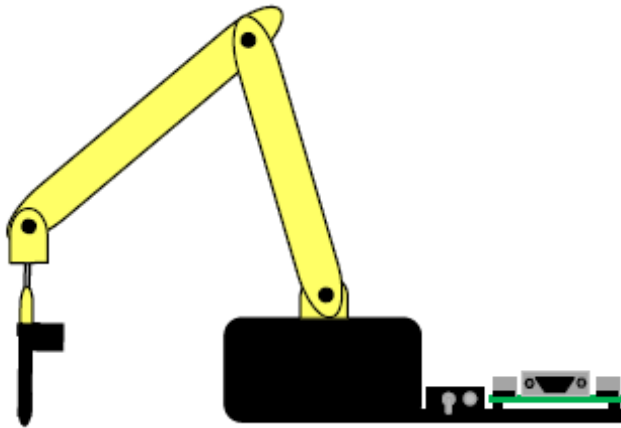


Figura 5 – Configuración “Zona 1”

La “Zona 2” está identificada sobre el esquema mediante el color verde, y en esta zona están contenidos los radios “RA, RB, RC, RD, RE, RF, RG, RH, RI, RJ, RK, RL, RM, RN, RO, RP y RQ”, las semicircunferencias “C3, C4, C5 y C6” y los Planos del 1 al 5, aunque no todos los planos son accesibles desde todas las semicircunferencias, la distribución es la siguiente; “P1L y P1H: C3, C4, C5 y C6”, “P2L y P2H: C3, C4, C5 y C6”, “P3L y P3H: C3, C4 y C5”, “P4L y P4H: C3 y C4” y “P5L y P5H: C3”. Esta limitación de la altura viene dada por las propias limitaciones físicas del robot, las cuales no permiten alcanzar los puntos que no se han incluido, ya que se hayan fuera de su espacio de trabajo. La configuración que adquiere el robot para situarse sobre los puntos característicos en esta zona, es como la que se observa en la siguiente figura (ver figura 6):

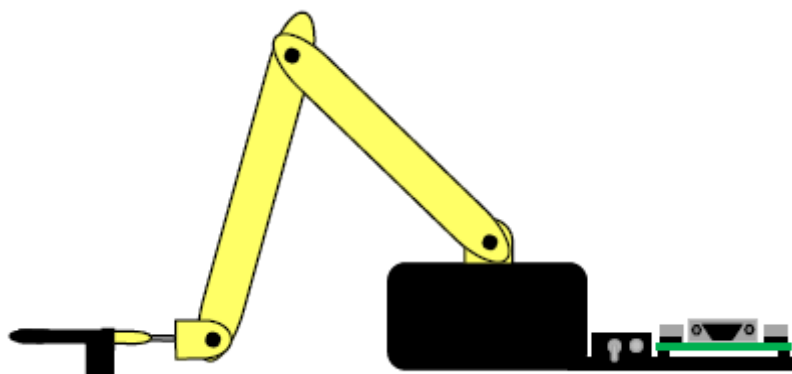


Figura 6- Configuración “Zona 2”

3. ENVIAR TRAYECTORIAS

Hay dos formas de introducir los nodos para que el robot ejecute la trayectoria que pase por estos:

La primera, es **mediante un archivo de texto** en el que cada línea representa un nodo referente a uno de los puntos característicos del espacio de trabajo del robot descritos en el capítulo anterior. El archivo deberá tener el formato *.txt* y se utilizará una sintaxis específica mediante un lenguaje de etiquetas.

La segunda, **seleccionándolos manualmente** mediante la interfaz web.

3.1. A TRAVÉS DE UN ARCHIVO DE TEXTO

Cada una de las instrucciones que representan un nodo se escribirá en una sola línea del archivo, y la última línea siempre debe contener la instrucción “END;”. El texto debe contener como mínimo 4 líneas (incluyendo la de la instrucción “END;”) ya que el sistema sólo funciona con un número de nodos mayor que 2.

El formato de las instrucciones será el siguiente:

<etiqueta de posición>,<tiempo de ejecución>;

Donde <etiqueta de posición> indica las coordenadas del punto característico del espacio de trabajo del robot donde se encuentra el nodo, y <tiempo de ejecución> es el tiempo en milisegundos que se desea que el manipulador tarde en realizar el movimiento desde el nodo anterior. El tiempo no debe ser inferior a los 500ms ni superior a los 1000ms.

La etiqueta de posición tiene el siguiente formato:

Pos<semicircunferencia><radio><plano><altura><zona>

Donde el campo <zona> sólo se utiliza para identificar con que configuración se sitúa el robot sobre los puntos contenidos en el plano “PIH” y “PIL” de la

semicircunferencia “C3”, ya que ésta es el nexo de unión entre la zona de trabajo 1 y la zona de trabajo 2 y sirve para poder llevar objetos de una zona a la otra sin alterar la inclinación del objeto.

Veamos algunos ejemplos de instrucciones completas válidas:

- **PosC1RAP1L,600;**

Coloca el efector final del robot en la semicircunferencia C1, radio RA, plano P1 y altura L en un tiempo de 600 ms.

- **PosC3RAP1LZ1,800;**

Coloca el efector final del robot en la semicircunferencia C3, radio RA, plano P1, altura L y configuración “Zona1” en un tiempo de 800 ms.

- **PosC3REP1HZ2,1000;**

Coloca el efector final del robot en la semicircunferencia C3, radio RE, plano P1, altura H y configuración “Zona2” en un tiempo de 1000 ms.

- **PosC3RATR,900;**

Esta es una **instrucción de transición**. Se utiliza para llevar al robot a una posición elevada cuando queremos cambiar su configuración entre la zona 1 y la zona 2, y así evitar que colisione con el objeto que está manipulando. Como se puede observar, en las instrucciones de transición no hay que indicar los campos <plano> ni <altura>.

Una vez editado el archivo de texto correctamente, lo enviamos al servidor mediante los controles de la interfaz mostrados en la siguiente figura (ver figura 7):

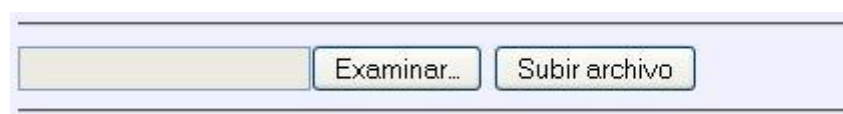


Figura 7 – Enviar archivo de texto

3.2. DE FORMA MANUAL

El usuario introduce cada nodo mediante una serie de listas desplegables de la interfaz que permiten seleccionar la semicircunferencia, radio, plano, altura, zona y tiempo de ejecución (ver figura 8). Una vez seleccionadas las coordenadas, presiona el botón “Nuevo nodo” para añadirlo a la lista. Si el nodo introducido corresponde a una posición válida, se añade a la lista de nodos del usuario. Finalmente pulsamos el botón “Enviar” para aceptar la lista de nodos introducidos y esperar a la ejecución de la trayectoria por parte del robot.

circunferencia	radio	plano	altura	zona	tiempo
C1	RA	P1	H	Abierta	600
C2	RA	P1	H	Abierta	600
C2	RC	P1	H	Abierta	600

Figura 8 –Enviar nodos manualmente



Universitat
Autònoma
de Barcelona



**PROYECTO N° 2017. IMPLEMENTACIÓN DE MOVIMIENTOS SUAVES EN
ROBOTS INDUSTRIALES MEDIANTE SPLINES CÚBICOS. APLICACIÓN A UN
LABORATORIO REMOTO DE ROBÓTICA**

ANEXO 8

Anexo del Proyecto Fin de Carrera
De Ingeniería en Informàtica

Realizado por

Marcos Mohedano Gómez

y dirigido por

Asier Ibeas Hernández

Bellaterra, Junio 2010



Universitat
Autònoma
de Barcelona



escola
d'enginyeria

**Laboratorio remoto de Robótica:
Implementación de movimientos suaves
en robots industriales mediante *splines*
cúbicos.**

**Gestión de cuentas de
usuario**

ÍNDICE

1. INTRODUCCIÓN	1
2. CREAR CUENTA	2
3. ELIMINAR CUENTA	4

1. INTRODUCCIÓN

Este tutorial va dirigido al administrador del servidor encargado del mantenimiento de la web del laboratorio remoto de robótica correspondiente al actual proyecto. Se mostrará cómo administrar cuentas de usuario en el sistema, concretamente para crear y eliminar cuentas de usuario.

Para ello necesitamos la herramienta de desarrollo web con la que se ha realizado la programación de la web, **Visual Web Developer Express Edition**. En los próximos capítulos se explicará cómo utilizar esta herramienta para la administración de cuentas.

2. CREAR CUENTA

Una vez en marcha la aplicación Visual Web Developer y seleccionado nuestro sitio web, se realizarán los siguientes pasos:

1. En el menú, seleccionar la opción “Sitio web”, y a continuación “Configuración de ASP.NET” como se indica en la siguiente figura (ver figura 1).

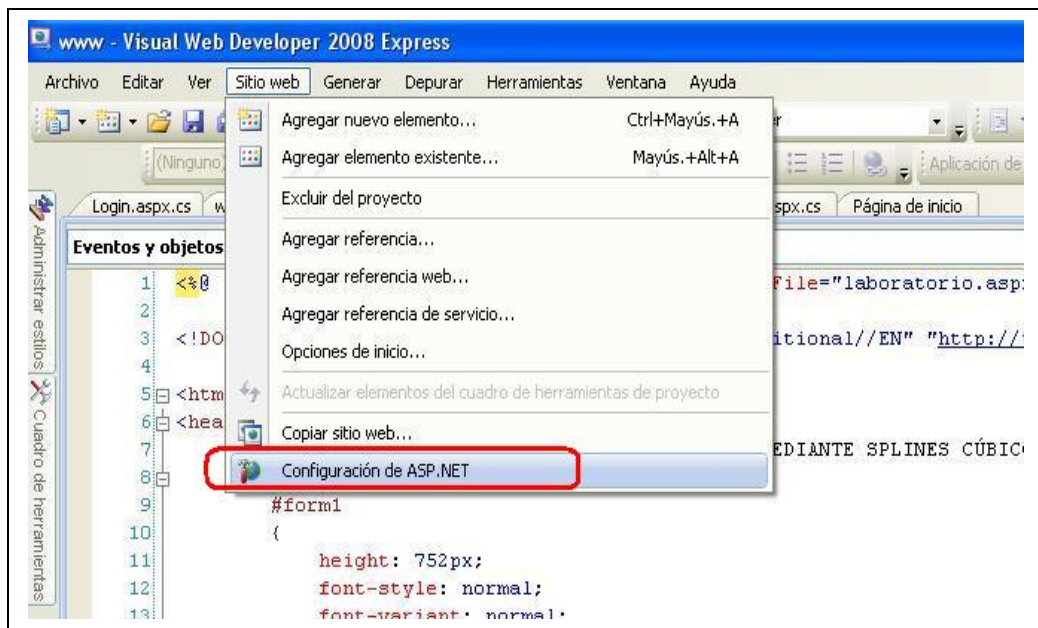


Figura 1- Menú “Sitio Web”

2. Se abrirá una ventana del navegador web predeterminado en la que aparece una interfaz con varias pestañas. Seleccionar la pestaña “Seguridad” (ver figura 2).
3. Hacer click en el vínculo “crear usuario” que hay debajo del título “USUARIOS” (ver figura 2).



Figura 2 – Administración del sitio web

4. Aparecerán una serie de cuadros de texto para escribir el nombre del nuevo usuario, la que será su contraseña y la dirección de correo electrónico. Una vez introducidos los datos pulsar el botón “Crear usuario” y ya estará lista la nueva cuenta (ver figura 3).

Figura 3 – Crear usuario

3. ELIMINAR CUENTA

Estos son los pasos a seguir para eliminar una cuenta de usuario ya existente:

1. Repetir los pasos “1” y “2” del capítulo anterior, para situarnos en la pestaña “Seguridad” del menú de configuración de ASP.NET.
2. Hacer click sobre el vínculo “Administrar usuarios”(ver figura 4).



Figura 4 - Administración del sitio web

3. Se abrirá una página donde aparece una lista con los usuarios existentes. Seleccionar el vínculo “Eliminar” correspondiente a la cuenta de usuario que se desea borrar (ver figura 5).

Activo	Nombre de usuario			
☒	manolo	Editar usuario	Eliminar usuario	Editar funciones
☒	marcos	Editar usuario	Eliminar usuario	Editar funciones
☒	pepito	Editar usuario	Eliminar usuario	Editar funciones

[Crear nuevo usuario](#)

Figura 5 – Eliminar usuario