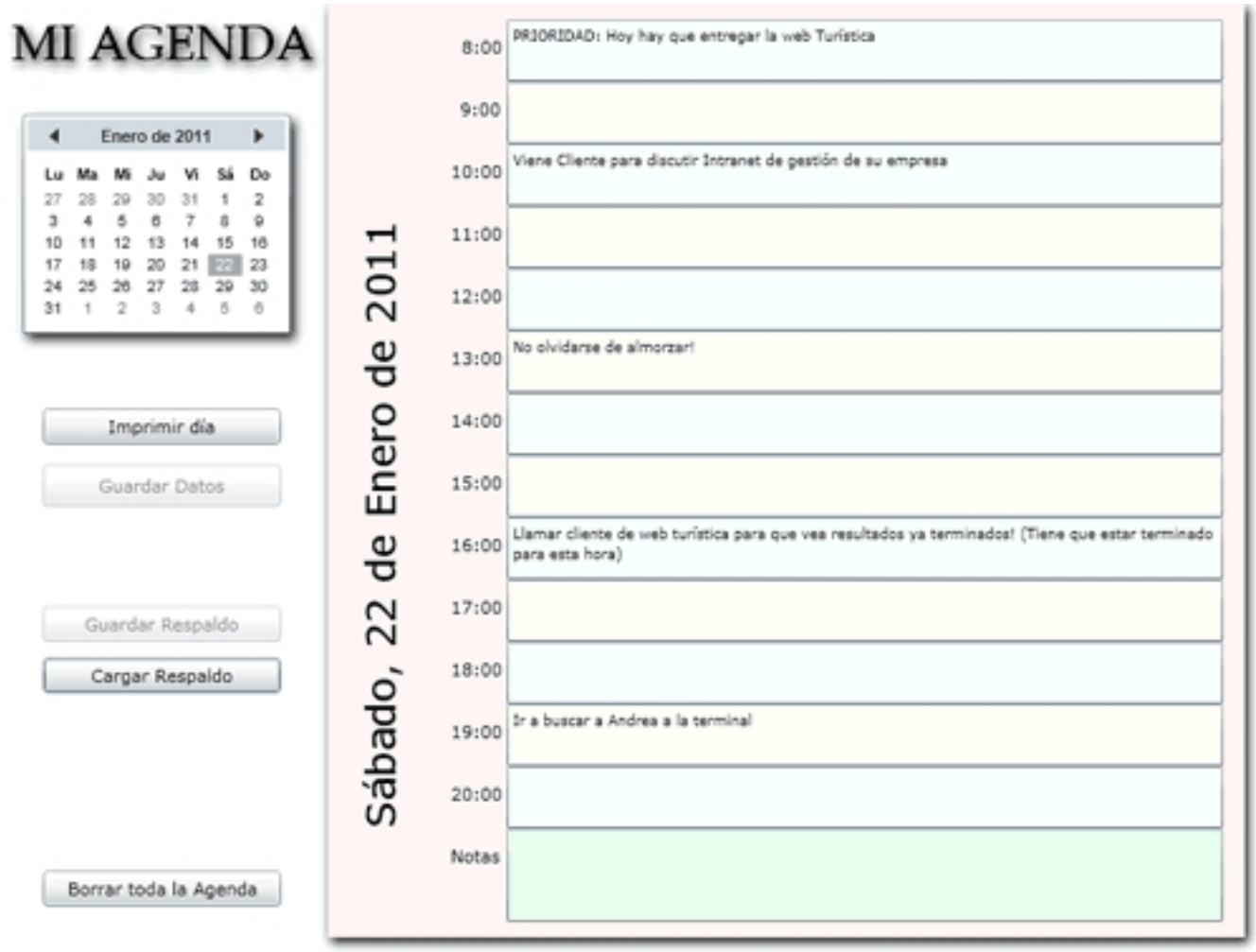

Como segundo mini-tutorial del Silverlight, quería una aplicación sencilla de agenda de actividades: La idea básica es presentar un calendario donde el usuario puede elegir un día cualquiera, y a la derecha presentar una "hoja" de agenda con las horas desde las 8:00 hasta las 20:00 y un area de NOTAS. La información se guarda internamente en el isolatedspace del equipo del usuario, en un archivo XML.

El usuario, en la medida que va agendando actividades debe poder apretar un botón para actualizar la información / cambios actividades al archivo XML del disco.El usuario tiene también de sacar una copia del XML y guardarlo en donde quiera "como backup", o cargar un xml que pueda tener guardado y hacer un "restore" de la agenda, en cualquier momento. Tambien debe poder RESETEAR la agenda (eliminar los datos), y por último imprimir la hoja que está viendo. Hasta ahi lo que me propuse.

Luego comienza el "manos a la obra" y mi primer comentario es bastante fuerte: El objeto "Calendar" que viene en el sdk del Silverlight 4 es sumamente tosco y limitado: Una de las cosas que quería lograr era que el color de fondo de los días que tienen actividades marcadas cambiara en el calendario. Lo logré a medias: no pude "domarlo". Buscando luego en internet, vi que este problema y otros peores vienen siendo denunciados por los usuarios de Silverlight, y por lo tanto surgen como respuesta objetos de calendario de terceros.

Como no tenía ganas de complicar este minitutorial agregando la necesidad de instalar un objeto de terceros, desistí de la parte de colorcitos, aunque está cualquiera de ustedes habilitado y bienvenido/a a agregarlo y postear si quieren como comentario o en el foro, vuestra versión mas bonita... Y hablando de cosas bonitas, el manejo del XML en este ejemplo es mas bien rudimentario. Hay aire para grandes mejoras y optimizaciones... esta es una "primera versión"... y al que le interese, puede retomar este trabajo y mejorarlo :)

Archivo XAML:

```
<Grid x:Name="LayoutRoot" Background="White"> <TextBlock Height="50" Margin="8,5,0,0" TextWrapping="Wrap" Text="MI AGENDA" VerticalAlignment="Top" FontSize="32" FontFamily="Book Antiqua" HorizontalAlignment="Left" Width="197">

<TextBlock.Effect>

<DropShadowEffect ShadowDepth="3" BlurRadius="10"/>

</TextBlock.Effect>
```

```
</TextBlock>

<sdk:Calendar x:Name="calendario" Margin="9,69,0,0" FontFamily="Arial"
FontSize="21.333" HorizontalAlignment="Left" Width="183"
RenderTransformOrigin="0.5,0.5" Height="155" VerticalAlignment="Top" >

<sdk:Calendar.Effect>

<DropShadowEffect BlurRadius="10" Opacity="0.85"/>

</sdk:Calendar.Effect>

</sdk:Calendar>


<Button IsEnabled="False" Name="guardar"  Content="Guardar Datos"
HorizontalAlignment="Left" Margin="28,0,0,277" Width="153" Height="27"
VerticalAlignment="Bottom"/>

<Canvas Height="600" Name="hoja" HorizontalAlignment="Left" Margin="211,0,0,0"
VerticalAlignment="Top" Width="589" Background="#FFFFF5F5">

<Canvas.Effect>

<DropShadowEffect ShadowDepth="3" BlurRadius="10"/>

</Canvas.Effect>

<sdk:Label HorizontalAlignment="Left" Margin="280,20,0,0" VerticalAlignment="Top"
Content="8:00" Canvas.Left="-194"  />

<sdk:Label HorizontalAlignment="Left" Margin="279,60,0,0" VerticalAlignment="Top"
Content="9:00" Canvas.Left="-194"  />

<sdk:Label HorizontalAlignment="Left" Margin="273,100,0,0" VerticalAlignment="Top"
Content="10:00" Canvas.Left="-194" />

<sdk:Label HorizontalAlignment="Left" Margin="273,140,0,0" VerticalAlignment="Top"
Content="11:00" Canvas.Left="-194" />

<sdk:Label HorizontalAlignment="Left" Margin="273,180,0,0" VerticalAlignment="Top"
Content="12:00" Canvas.Left="-194" />

<sdk:Label HorizontalAlignment="Left" Margin="273,220,0,0" VerticalAlignment="Top"
Content="13:00" Canvas.Left="-194" />
```

```
<sdk:Label HorizontalAlignment="Left" Margin="273,260,0,0" VerticalAlignment="Top"
Content="14:00" Canvas.Left="-194" />

<sdk:Label HorizontalAlignment="Left" Margin="273,300,0,0" □VerticalAlignment="Top"
Content="15:00" Canvas.Left="-194" />

<sdk:Label HorizontalAlignment="Left" Margin="273,340,0,0" VerticalAlignment="Top"
Content="16:00" Canvas.Left="-194" □/>

<sdk:Label HorizontalAlignment="Left" Margin="273,380,0,0" VerticalAlignment="Top"
Content="17:00" Canvas.Left="-194" />

<sdk:Label HorizontalAlignment="Left" Margin="273,420,0,0" VerticalAlignment="Top"
Content="18:00" Canvas.Left="-194" />

<sdk:Label HorizontalAlignment="Left" Margin="273,460,0,0" VerticalAlignment="Top"
Content="19:00" Canvas.Left="-194" />

<sdk:Label HorizontalAlignment="Left" Margin="273,500,0,0" VerticalAlignment="Top"
Content="20:00" Canvas.Left="-194" />

<sdk:Label HorizontalAlignment="Left" Margin="273,540,0,0" VerticalAlignment="Top"
Content="Notas" Canvas.Left="-194" />

<TextBox x:Name="hora8" Width="460" Height="40" Margin="309,10,11,0"
TextWrapping="Wrap" VerticalAlignment="Top" Background="#FFf7FFFf"
AcceptsReturn="True" FontSize="9" Canvas.Left="-194" />

<TextBox x:Name="hora9" Width="460" Height="40" Margin="309,50,11,0"
TextWrapping="Wrap" VerticalAlignment="Top" Background="#FFFFFFf7"
AcceptsReturn="True" FontSize="9" Canvas.Left="-194" />

<TextBox x:Name="hora10" Width="460" Height="40" Margin="309,90,11,0"
TextWrapping="Wrap" VerticalAlignment="Top" Background="#FFf7FFFf"
AcceptsReturn="True" FontSize="9" Canvas.Left="-194" />

<TextBox x:Name="hora11" Width="460" Height="40" Margin="309,130,11,0"
TextWrapping="Wrap" VerticalAlignment="Top" Background="#FFFFFFf7"
AcceptsReturn="True" FontSize="9" Canvas.Left="-194" />

<TextBox x:Name="hora12" Width="460" Height="40" Margin="309,170,11,0"
TextWrapping="Wrap" VerticalAlignment="Top" Background="#FFf7FFFf"
AcceptsReturn="True" FontSize="9" Canvas.Left="-194" />

<TextBox x:Name="hora13" Width="460" Height="40" Margin="309,210,11,0"
TextWrapping="Wrap" VerticalAlignment="Top" Background="#FFFFFFf7"
```

```
AcceptsReturn="True" FontSize="9" Canvas.Left="-194" />

<TextBox x:Name="hora14" Width="460" Height="40" Margin="309,250,11,0"
TextWrapping="Wrap" VerticalAlignment="Top" Background="#FFf7FFFf"
AcceptsReturn="True" FontSize="9" Canvas.Left="-194" />

<TextBox x:Name="hora15" Width="460" Height="40" Margin="309,290,11,0"
TextWrapping="Wrap" VerticalAlignment="Top" Background="#FFFFFFf7"
AcceptsReturn="True" FontSize="9" Canvas.Left="-194" />

<TextBox x:Name="hora16" Width="460" Height="40" Margin="309,330,11,0"
TextWrapping="Wrap" VerticalAlignment="Top" Background="#FFf7FFFf"
AcceptsReturn="True" FontSize="9" Canvas.Left="-194" />

<TextBox x:Name="hora17" Width="460" Height="40" Margin="309,370,11,0"
TextWrapping="Wrap" VerticalAlignment="Top" Background="#FFFFFFf7"
AcceptsReturn="True" FontSize="9" Canvas.Left="-194" />

<TextBox x:Name="hora18" Width="460" Height="40" Margin="309,410,11,0"
TextWrapping="Wrap" VerticalAlignment="Top" Background="#FFf7FFFf"
AcceptsReturn="True" FontSize="9" Canvas.Left="-194" />

<TextBox x:Name="hora19" Width="460" Height="40" Margin="309,450,11,0"
TextWrapping="Wrap" VerticalAlignment="Top" Background="#FFFFFFf7"
AcceptsReturn="True" FontSize="9" Canvas.Left="-194" />

<TextBox x:Name="hora20" Width="460" Height="40" Margin="309,490,11,0"
TextWrapping="Wrap" VerticalAlignment="Top" Background="#FFf7FFFf"
AcceptsReturn="True" FontSize="9" Canvas.Left="-194" />

<TextBox x:Name="notas" Width="460" Height="60" Margin="309,530,11,0"
TextWrapping="Wrap" VerticalAlignment="Top" Background="#FFe7ffee"
AcceptsReturn="True" FontSize="9" Canvas.Left="-194" />

<sdk:Label Canvas.Left="-215.886" Name="actual" Canvas.Top="256" Width="500"
RenderTransformOrigin="0.5,0.5" UseLayoutRounding="False"
d:LayoutRounding="Auto" Height="48" □ FontSize="26">

<sdk:Label.RenderTransform>

<CompositeTransform Rotation="-90"/>

</sdk:Label.RenderTransform>

</sdk:Label>
```

```
</Canvas>

<Button Content="Imprimir día" Height="23" Name="imprimir"
HorizontalAlignment="Left" Margin="28,260,0,0" VerticalAlignment="Top" Width="153" />

<Button Content="Borrar toda la Agenda" Height="23" HorizontalAlignment="Left"
Margin="28,557,0,0" Name="borrar" VerticalAlignment="Top" Width="153" />

<Button Content="Guardar Respaldo" IsEnabled="False"  Height="23"
HorizontalAlignment="Left" Margin="28,387,0,0" Name="respaldar"
VerticalAlignment="Top" Width="153" />

<Button Content="Cargar Respaldo" Height="23" HorizontalAlignment="Left"
Margin="28,420,0,0" Name="restorear" VerticalAlignment="Top" Width="153" />

</Grid>
```

El grid es bien sencillo, aunque un poco repetitivo en sus objetos. Para hacerlo mas divertido podría haber creado los labels y textboxes de los horarios programaticamente e insertarlos al ejecutarse el código dentro del grid ... pero a los efectos del ejemplo, una alternativa asi podemos dejarla para un próximo tutorial. El "code-behind" de esta aplicación comienza asi:

```
 Imports System.IO  Imports System.Xml.Linq  Imports System.IO.IsolatedStorage  Impor
ts System.Windows.Printing

Imports System.Windows.Controls.Primitives



Partial Public Class MainPage

Inherits UserControl
```

```
Public Sub New()

InitializeComponent()

End Sub



Private WithEvents imprime As New PrintDocument

Dim eventos As XDocument

Dim crt As String = ChrW(13) + ChrW(10)



Private Sub MainPage_Loaded(ByVal sender As Object, ByVal e As
System.Windows.RoutedEventArgs) Handles Me.Loaded

Dim fecha As Date = DateTime.Now.Date

'Primero me posiciono sobre el día de hoy:

calendario.DisplayDate = fecha

'Luego cargo los eventos

cargaragenda("miagendadeeventos.xml")

'luego muestro lo que hay para hacer en este día:

mostrarfecha(fecha)

End Sub
```

El XML "eventos" es el que contendrá todos los eventos / apuntes en mi agenda. Al cargar la aplicación, partimos de la fecha actual, posicionando el calendario en ese día, cargando el xml

desde el isolatedspace del usuario y mostrando los eventos de esta fecha. Autoexplicativo ya viendo los comentarios :) Antes de continuar, veamos como está formado el XML ... en este ejemplo (donde capturé la pantalla, arriba) el XML es asi:

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?> <!--(C)Enrique Avalle 2011--> <agenda> <A2011 > <M1>

<D22>

<hora8 value="PRIORIDAD: Hoy hay que entregar la web Turística" />

<hora10 value="Viene Cliente para discutir Intranet de gestión de su empresa" />

<hora13 value="No olvidarse de almorzar!" />

<hora16 value="Llamar cliente de web turística para que vea resultados ya terminados! (Tiene que estar terminado para esta hora)" />

<hora19 value="Ir a buscar a Andrea a la terminal" />

</D22>

</M1>

</A2011>

</agenda>
```

Los nodos y la forma general de este XML es arbitraria pero facilmente entendible: A2011 es el año 2011 .... M1 es el Mes 1, D22 es el día 22, y luego tenemos las hdistintas horas cuyos values son las anotaciones para esas horas. Continuando con el código:

```
'Limpio todos los textboxes de la hoja de la agenda: '---------- Sub limpiohoras() For Each ctr As Object In hoja.Children

Dim txt As TextBox = TryCast(ctr, TextBox)

If Not txt Is Nothing Then txt.Text = ""

Next

End Sub
```

```
'Apago textboxes de la hoja para que no procesen eventos

'-------------------------------------------------------

Sub apagohoras()

For Each ctr As Object In hoja.Children

Dim txt As TextBox = TryCast(ctr, TextBox)

If Not txt Is Nothing Then RemoveHandler txt.TextChanged, AddressOf cambiocontenido

Next

End Sub
```

```
'Prendo textboxes de la hoja para que procesen eventos

'-------------------------------------------------------

Sub prendohoras()

For Each ctr As Object In hoja.Children

Dim txt As TextBox = TryCast(ctr, TextBox)

If Not txt Is Nothing Then AddHandler txt.TextChanged, AddressOf cambiocontenido

Next

End Sub
```

Estas tres subrutinas sirven para entender varias cosas mas adelante. Primero una aclaración: Para actualizar el XML en memoria, nos valemos del event handler de "textchanged" de cada textbox. Suena a medio mucho, estar actualizando el xml letra a letra, pero por ahora es lo que al menos a mi me pareció como "a prueba de balas" para no perder

ningún texto ingresado. Y en mi portátil este método no enlentence el proceso en lo mas mínimo, por lo que decidí que queda asi. No obstante, cuando cargo el XML y promuevo las actividades del día del XML a los texboxes, no me sirve que "se actualice el XML" en cada caso donde le asigno lo que hay para esa hora, porque en definitiva SALE DEL MISMO XML y es redundante. De ahi a que haya hecho las subrutinas que "agregan" o "quitan" los handlers ... para luego poder "encender" o "apagar" la actualización en memoria, dependiendo del momento en que estoy en la aplicación. Lo vamos a ver ya en el próximo código:

```
Sub mostrarfecha(ByVal fecha As Date) 'Como muestro esta fecha directo del xml, preveo que el handler de los textboxes cuando se dispare, no reescriba el xml (al santo botón) con el mismo contenido apagohoras() ' Primero titulo hoja con el dia, mes y año de esta fecha:

actual.Content = fecha.ToLongDateString

'Segundo blanqueo las horas en pantalla

limpiohoras()

'Tercero populo horas para ese dia:

Try

Dim eldia As XElement = eventos.Element("agenda").Element("A" + fecha.Year.ToString).Element("M" + fecha.Month.ToString).Element("D" + fecha.Day.ToString)

'Aparentemente hay cosas para colorear:

For Each hora As XElement In eldia.Elements

If hora.HasAttributes Then

'Populo esta hora:

Dim eltexto As TextBox = hoja.FindName(hora.Name.ToString)

eltexto.Text = hora.Attribute("value")

End If

Next

Catch ex As Exception
```

```
'No hago nada ... no hay dias ocupados en esta agenda para este mes ...

End Try

prendohoras()

End Sub

        'Eligió otro día en el calendario:  '----------  Private Sub
calendario_SelectedDatesChanged(ByVal sender As Object, ByVal e As
System.Windows.Controls.SelectionChangedEventArgs) Handles
calendario.SelectedDatesChanged

'Levanto nuevo mes y año:

Dim fecha As Date = calendario.SelectedDate

'calendario.DisplayDate = fecha

'luego muestro lo que hay para hacer en este día:

mostrarfecha(fecha)

End Sub
```

El primer Sub promueve del XML a los textboxes, lo que hay para hacer en determinada fecha. Fijate como apago los handlers, populo los textboxes si hay contenido y al terminar reenciendo los handlers... El segundo Sub, que actúa cuando el usuario elige otro día en el calendario, toma esa nueva fecha y la muestra usando la rutina de mas arriba.

```
 'GRABO AGENDA EN EL DISCO DEL USUARIO:  '----------  Private
Sub grabaragenda(ByVal nombre As String)

Dim isostore As IsolatedStorageFile = IsolatedStorageFile.GetUserStoreForApplication

Dim isoStream As IsolatedStorageFileStream = New IsolatedStorageFileStream(nombre,
FileMode.Create, isostore)

eventos.Save(isoStream)

isoStream.Close()
```

```
isostore.Dispose()

guardar.IsEnabled = False 'Ya guardo, le dejo el botón "guardar" en grisesito, hasta que agrega datos nuevos en el calendario.

End Sub

    'CARGO AGENDA DESDE EL DISCO DEL USUARIO:  '----------------------------------------

Private Sub cargaragenda(ByVal nombre As String)

Dim isostore As IsolatedStorageFile = IsolatedStorageFile.GetUserStoreForApplication


If isostore.FileExists(nombre) Then

'cargo el archivo

Dim isoStream As IsolatedStorageFileStream = New IsolatedStorageFileStream(nombre, FileMode.Open, isostore)

eventos = XDocument.Load(isoStream)

isoStream.Close()

isostore.Dispose()

Else

isostore.Dispose()

'No existe el archivo aùn: le damos la bienvenida

MessageBox.Show("Esta aplicación es una  sencilla agenda calendario que le permite moverse" + crt + " por los días del mes y anotar tareas a realizar.Este calendario guarda su" + crt + "información en forma local, dentro de su propio PC, por lo tanto" + crt + "su información personal queda a resguardo en su propio disco duro, no" + crt + "teniendo nuestro servidor copia alguna." + crt + crt + "Esto implica que esta aplicación debe ser utilizada siempre desde" + crt + "su computadora personal y si desea utilizar esta aplicación en" + crt + " forma usual, le convendrá respaldar periódicamente la información" + crt + "(hay un botón para respaldarla).", "Bienvenido !", MessageBoxButton.OK)

'y procedemos a inicializar el archivo xml:

'----------------------------------------
```

```
resetearagenda(nombre)

End If

End Sub

    'BORRA TODO DE LA AGENDA:  '----------  Sub resetearagenda(ByVal
nombre As String)                                                    Dim agenda
As New XElement("agenda")

eventos = New XDocument(New XDeclaration("1.0", "utf-8", "yes"), New
XComment("(C)Enrique Avalle 2011"), agenda)

'y lo mando grabar

grabaragenda(nombre)

End Sub
```

Grabamos o cargamos el XML desde el isolatedspace del usuario. Si no existe el XML, entonces es la primera vez que el usuario ejecuta esta aplicación (en ese equipo) y por lo tanto le damos la "bienvenida" y generamos un XML "marco" como quien dice para "plantar bandera" en el equipo del usuario :) También tenemos una rutina para hacer "borrón y cuenta nueva" en la agenda, que basicamente genera el esqueleto o "marco" del XML y lo graba ...

```
 'APRETA BOTON DE IMPRIMIR:  '----------  Private Sub imprimir_Click(ByVal
sender As System.Object, ByVal e As System.Windows.RoutedEventArgs) Handles
imprimir.Click                                                    'Y mando imprimir:  i
mprime.Print("Agenda Calendario")

End Sub


'Rutina de imprimir del silverlight pasa por acá

Private Sub imprime_PrintPage(ByVal sender As Object, ByVal e As PrintPageEventArgs)
Handles imprime.PrintPage

e.PageVisual = hoja

End Sub
```

Sencillo código que maneja el tema de la impresión de la "hoja" de la agenda. Si nos fijamos, el control "hoja" es un CANVAS que adentro tiene el label de la fecha, los labels de las horas y los textboxes correspondientes: Solamente queremos imprimir "ese" Canvas.

```
'Esto se ejecuta como handler de cada textbox, con cada letra que el usuario escribe en la agenda, actualizando el xml en si '----------
---------- Sub cambiocontenido(ByVal sender As Object, ByVal e As System.Windows.Controls.TextChangedEventArgs)

Dim contenido As TextBox = sender 'Levanto el textbox donde viene el contenido cambiado

guardar.IsEnabled = True

respaldar.IsEnabled = True

'Levanto nuevo mes y año:

Dim fecha As Date 'Aquí vemos que fecha viene en el calendario (elegida, o si no hay una fecha elegida, la que esté mostrando)

If Not calendario.SelectedDate Is Nothing Then fecha = calendario.SelectedDate Else fecha = calendario.DisplayDate

'Actualizo este contenido en el calendario para esta fecha:

If contenido.Text <> "" Then

'-------------

'mas contenido

'-------------

If eventos.Element("agenda").Element("A" + fecha.Year.ToString) Is Nothing Then

'Nuevo año

eventos.Element("agenda").Add(New XElement("A" + fecha.Year.ToString, New XElement("M" + fecha.Month.ToString, New XElement("D" + fecha.Day.ToString, New XElement(contenido.Name, New XAttribute("value", contenido.Text))))))

ElseIf eventos.Element("agenda").Element("A" + fecha.Year.ToString).Element("M" +
```

```
fecha.Month.ToString) Is Nothing Then

'Nuevo mes

eventos.Element("agenda").Element("A" + fecha.Year.ToString).Add(New XElement("M"
+ fecha.Month.ToString, New XElement("D" + fecha.Day.ToString, New
XElement(contenido.Name, New XAttribute("value", contenido.Text)))))

ElseIf eventos.Element("agenda").Element("A" + fecha.Year.ToString).Element("M" +
fecha.Month.ToString).Element("D" + fecha.Day.ToString) Is Nothing Then

'Nuevo dia

eventos.Element("agenda").Element("A" + fecha.Year.ToString).Element("M" +
fecha.Month.ToString).Add(New XElement("D" + fecha.Day.ToString, New
XElement(contenido.Name, New XAttribute("value", contenido.Text))))

ElseIf eventos.Element("agenda").Element("A" + fecha.Year.ToString).Element("M" +
fecha.Month.ToString).Element("D" + fecha.Day.ToString).Element(contenido.Name) Is
Nothing Then

'Nueva HORA

eventos.Element("agenda").Element("A" + fecha.Year.ToString).Element("M" +
fecha.Month.ToString).Element("D" + fecha.Day.ToString).Add(New
XElement(contenido.Name, New XAttribute("value", contenido.Text)))

Else

eventos.Element("agenda").Element("A" + fecha.Year.ToString).Element("M" +
fecha.Month.ToString).Element("D" +
fecha.Day.ToString).Element(contenido.Name).Attribute("value").SetValue(contenido.Tex
t)

End If

Else

'----------------------------

'EL contenido viene en blanco:

'----------------------------

Try
```

```
eventos.Element("agenda").Element("A" + fecha.Year.ToString).Element("M" +
fecha.Month.ToString).Element("D" +
fecha.Day.ToString).Element(contenido.Name).Remove()

'Entonces vemos si ahora ese dia queda en blanco:

If eventos.Element("agenda").Element("A" + fecha.Year.ToString).Element("M" +
fecha.Month.ToString).Element("D" + fecha.Day.ToString).FirstNode Is Nothing Then

'Como el dia queda en blanco, lo borramos:

eventos.Element("agenda").Element("A" + fecha.Year.ToString).Element("M" +
fecha.Month.ToString).Element("D" + fecha.Day.ToString).Remove()

'Entonces vemos si el mes queda en blanco:

If eventos.Element("agenda").Element("A" + fecha.Year.ToString).Element("M" +
fecha.Month.ToString).FirstNode Is Nothing Then

'y si el mes queda en blanco, lo borramos tmb:

eventos.Element("agenda").Element("A" + fecha.Year.ToString).Element("M" +
fecha.Month.ToString).Remove()

'Por lo que vemos si el año queda en blanco!

If eventos.Element("agenda").Element("A" + fecha.Year.ToString).FirstNode Is Nothing
Then

'y si el año queda en blanco, lo borramos tmb:

eventos.Element("agenda").Element("A" + fecha.Year.ToString).Remove()

End If

End If

End If

Catch ex As Exception

'No habia nodo alguno en realidad para limpiar

End Try

End If
```

End Sub

Qué entrevero! esta sub es la que se llama con cada caracter que digita el usuario dentro de los textboxes. Tiene dos grandes partes: La primera donde intenta insertar el texto escrito, y va creando el "path" en el XML en la medida en que necesita, y la segunda donde va BORRANDO el contenido y hacia atrás el PATH del XML también, cosa de intentar tener siempre como resultado un XML "limpio" sin nodos carentes de "contenido" real. El punto es que funciona, aunque estoy seguro de que puede ser optimizado, sobre todo considerando que en el Silverlight 4 se ha agregado el uso de XPath y mejorado sustancialmente el manejo de XML.

```
 'APRETA GUARDAR AGENDA:  '----------  Private Sub guardar_Click(ByVal
sender As System.Object, ByVal e As System.Windows.RoutedEventArgs) Handles
guardar.Click                                                    grabaragenda("miagend
adeeventos.xml")                                                 guard
ar.IsEnabled = False

End Sub

      'Pide borrar toda la agenda:  '----------  Private Sub borrar_Click(ByVal
sender As System.Object, ByVal e As System.Windows.RoutedEventArgs) Handles
borrar.Click                                                     If
MessageBox.Show("ATENCIÓN: Va a eliminar toda la información de esta agenda." + crt
+ "Si está seguro de que quiere hacer esto, presione SI." + crt + "Si quiere mantener su
información y no borrar nada presione NO", "ATENCIÓN: Va a borrar todo!",
MessageBoxButton.OKCancel) <> MessageBoxResult.Cancel Then

'BORRA TODO:

resetearagenda("miagendadeeventos.xml")

guardar.IsEnabled = False 'no tiene sentido por ahora el botón de "guardar"

limpiohoras() 'quito textos que puedan haber en la pantalla

End If

End Sub

      'QUIERE GUARDAR / RECARGAR LOCALMENTE EL ARCHIVO XML:  '----------
----------
```

```
Private Sub respaldar_Click(ByVal sender As System.Object, ByVal e As
System.Windows.RoutedEventArgs) Handles respaldar.Click

Dim grabo As New SaveFileDialog()

grabo.Filter = "XML files (*.xml)|*.xml|All Files (*.*)|*.*"

grabo.DefaultExt = ".xml"

grabo.FilterIndex = 1

'Grabo la imagen en cuestión:

If grabo.ShowDialog Then

Dim elstream As Stream = grabo.OpenFile

eventos.Save(elstream)

elstream.Close()

respaldar.IsEnabled = False

End If

End Sub

    Private Sub restorear_Click(ByVal sender As System.Object, ByVal e As
System.Windows.RoutedEventArgs) Handles restorear.Click     Dim cargo As New
OpenFileDialog()                                                    carg
o.Filter = "XML files (*.xml)|*.xml"

cargo.FilterIndex = 1

cargo.Multiselect = False

Dim quiere As Boolean = cargo.ShowDialog

If quiere = True Then

Dim reader As StreamReader = cargo.File.OpenText

eventos = XDocument.Load(reader)

reader.Close()
```

```
Dim fecha As Date = DateTime.Now.Date

'Primero me posiciono sobre el día de hoy:

calendario.DisplayDate = fecha

'luego muestro lo que hay para hacer en este día:

mostrarfecha(fecha)

End If

End Sub
```

Por último algunas funcionalidades para completar los botones de la aplicación, incluyendo el backup y restore del XML a un archivo que pueda ser manipulado por el usuario, fuera del isolatedspace del Silverlight, asi como el handler del botón "guardar datos" que actualiza el XML del isolatedspace con el que se está trabajando en memoria, y el peligroso handler del "borrar datos" :) Con esto termino este nuevo tutorial, que puede -espero- ayudar a algún emprendedor de Silverlight a resolver algún detalle básico. En mi caso fue divertido esto, y me permitió explorar un poco mas esta herramienta.