

UNIVERSIDAD DE SONORA
DIVISIÓN DE INGENIERÍA
Departamento de Ingeniería Industrial

**“Actualización y Mantenimiento de un Framework
Empresarial para el Control de Reportes y Plantilla
Laboral”**

Reporte de Prácticas Profesionales

CARLOS ULISES CHÁVEZ MERCADO
Expediente 210206621

INGENIERO EN SISTEMAS DE INFORMACIÓN

Director: Dr. Mario Barceló Valenzuela

HERMOSILLO, SONORA.

MARZO 2016

Índice General

1. Introducción	3
1.1 Descripción de la Empresa	4
1.2 Descripción del Proyecto	6
1.3 Justificación del Proyecto	7
1.4 Objetivos del Proyecto	8
2. Problemas planteados a resolver	9
2.1 Alcances en la solución de problemas.....	10
2.2 Limitaciones en la solución de problemas	11
3. Fundamentos teóricos de las herramientas y conocimientos aplicados	12
4. Procedimientos empleados y actividades desarrolladas	14
5. Análisis de la Experiencia Adquirida.....	40
5.1 Análisis General del Proyecto.....	40
5.2 Análisis de los Objetivos de las Prácticas	40
5.3 Análisis de las Actividades Realizadas	41
5.4 Análisis de la Metodología Utilizada	41
6. Conclusiones y Recomendaciones.....	43
7. Referencias Bibliográficas y Virtuales	44

Índice de Figuras

Figura 1. Fachada de PC HOME	4
Figura 4.1. Estructura Básica del Tablero de Control	14
Figura 4.2. Columnas del Tablero de Control	15
Figura 4.3. Estructura de Control	16
Figura 4.4. Tablero de Control Nivel Trabajadores	17
Figura 4.5. Listado de los Trabajadores.....	17
Figura 4.6. Plantilla Laboral JE.....	18
Figura 4.7. Metodo CargaDatos – Variables	Figura 4.8. Metodo CargaDatos – Puesto 20
Figura 4.9. Plantilla Laboral – DropDownLists.....	20
Figura 4.10. SelectedIndexChanged	21
Figura 4.11. Datos de Asignación	22
Figura 4.12. ClsPlantilla	23
Figura 4.13. Propiedades de ClsPlantilla	24
Figura 4.14. Métodos de Procedimientos Almacenados	25
Figura 4.15. Estructura Plantilla Base.....	27
Figura 4.16. Sección Cambios en DropDownList.....	27
Figura 4.17. Sección Carga de DropDownList	28
Figura 4.18. Sección Carga de Datos Generales.....	28
Figura 4.19. Sección Actualización de Controles.....	28
Figura 4.20. Sección Liberación de Prospecto.....	29
Figura 4.21. Ajuste de Porcentajes y Manejo de Tiempos.....	31
Figura 4.22. Formato de Tablero.....	31
Figura 4.23. División de Secciones	32
Figura 4.24. Carga de Datos	33
Figura 4.25. Consulta Split de Comas.....	34
Figura 4.26. Plantilla ENYRE	35
Figura 4.27. Recepción de Parámetros	39
Figura 4.28. Vínculos de Niveles.....	39
Figura 4.29. Recepción de Parámetros	39

1. Introducción

Las Prácticas Profesionales se realizaron en la empresa PC HOME S.A. de C.V., en cumplimiento de la Normatividad Institucional y los Planes de Estudio del Programa de Ingeniería en Sistemas de Información de la Universidad de Sonora.

La empresa actualmente cuenta con un sistema propio que su utiliza mayormente para generar reportes que ayuden a la toma de decisiones estratégicas. Desafortunadamente este sistema no cuenta con una función para monitorear efectivamente los equipos de trabajo y su personal en general. Estas tareas, se realizan por medio de hojas de cálculo sin ningún formato estándar, lo que dificulta bastante el control adecuado y la organización de las actividades de los empleados.

Por lo anteriormente expuesto, el trabajo realizado tiene el propósito de brindarle soluciones a la empresa haciendo un uso adecuado de las Tecnologías de la Información y la Comunicación, en los diferentes problemas que se puedan presentar en las actividades y eventos de la empresa, las necesidades abarcan varias fases bajo la consideración de que la empresa no cuenta actualmente con un sistema para coordinar las tareas de sus empleados.

1.1 Descripción de la Empresa

Debido a que la mayoría de la información de la empresa es de carácter confidencial incluso el trabajo realizado en éstas prácticas, por ello solo se mencionan los aspectos necesarios para la realización del reporte. La empresa cuenta con un número aproximado de 50,000 empleados y su función específica (relacionada a este reporte) es la administración de su personal, en una serie de eventos que incluyen tanto la asistencia a los mismos, como la movilidad de personas externas a la empresa (figura 1.1).



Figura 1. Fachada de PC HOME

La empresa maneja todo a través de dos equipos de trabajo. Uno de esos equipos son consultores externos a la empresa y son quienes implementaron el framework existente. Ellos se encargan de administrar la base de datos adecuadamente, el servidor y cualquier error que resulte en los módulos que hayan elaborado. También se encargan de organizar cualquier nuevo proyecto que incluya el desarrollo de varios módulos nuevos.

El otro equipo es el departamento de sistemas de la empresa en el cual desarrollé mis actividades. Ellos se encargan de actualizar los módulos existentes para cubrir cualquier nueva necesidad tecnológica, crear nuevos módulos para incrementar la funcionalidad del framework o para mostrar un nuevo reporte. Otra de sus tareas es la actualización de la base de datos, cada cierto tiempo llega un Excel o un PDF con información pertinente para alimentar los módulos al cual le deben extraer la información, limpiarla y acomodarla ya sea en una nueva tabla o en una tabla existente.

1.2 Descripción del Proyecto

Las necesidades tecnológicas de la empresa se estipularon considerando un futuro cercano con un máximo de seis meses. En este lapso, se deben adecuar todos los procesos al nuevo sistema y posteriormente crear una serie de módulos que le brinden a la empresa, las herramientas necesarias para las actividades posteriores a esos seis meses.

El proyecto se divide en tres etapas principales:

1. Creación de módulos y tableros de control para monitorear un evento importante dentro de la empresa, el cual es clave para la adecuación de la empresa al nuevo sistema.
2. Creación de módulos para la transición de procesos al sistema y la migración de la información existente a la base de datos del nuevo sistema.
3. Creación de módulos y tableros de control para monitorear el evento más importante en el cual se pondrá a prueba el sistema completo y se dará fin al proceso de transición de la empresa.

La tarea específica del proyecto que me fue asignada, es la elaboración de los tableros de control y de los módulos que coordinen la plantilla laboral de la empresa.

Estos tableros deben mostrar la cantidad de empleados del departamento en cuestión que deben asistir a los eventos, y si la tarea que se les asignó ya fue completada, enfocándose en lo general con la opción de navegar a través de los distintos niveles de la empresa ya sea por Región, Zona, Ciudad, Líder o Puesto.

Los módulos tienen como objetivo almacenar la información personal del trabajador dentro de la base de datos, así como el departamento al que pertenecen, quien es la persona que está a cargo de ellos, su puesto y cualquier otra información pertinente a las actividades que debe realizar.

El desglose completo de las actividades que se tuvieron que realizar para cumplir con esta tarea se pueden observar en los Objetivos del Proyecto

1.3 Justificación del Proyecto

Desafortunadamente aún existen muchas empresas que prefieren manejar toda su información por medio de hojas de cálculo o documentos físicos, los cuales tienden a ser poco confiables. El uso de estas herramientas dificulta el acceso a información específica en el momento y reduce considerablemente la velocidad en la que se puede manejar dicha información, esto nulifica completamente el proceso de toma de decisiones.

El uso de las Tecnologías de la Información y la Comunicación se ha vuelto indispensable para lograr la estandarización de los procesos mediante la utilización de sistemas apoyados en Software. De acuerdo a los lineamientos del “Software Engineering Institute” (SEI), plasmados en el Modelo de Capacidad de Madurez “Capability Maturity Model” (CMM 2002) [1] [2] donde nos indica que la madurez de las empresas y específicamente en lo relativo a Software: Las empresas son maduras en la medida que puedan controlar sus procesos y hacerlos confiables y predecibles.

De acuerdo a lo anterior, este proyecto se enfocó en la implementación de Tecnologías de la Información y la Comunicación, por medio de un sistema para lograr un manejo mejor de la información de la empresa PC HOME, para monitorear adecuadamente a todos sus empleados y para asegurarse de que sus líderes y gerentes, estén cumpliendo adecuadamente con las responsabilidades a su cargo.

La finalidad es dotar a la empresa de una herramienta tecnológica, que le permita mejorar la participación de sus empleados, tanto gerentes como subordinados, utilizando adecuadamente los recursos informáticos y tecnológicos existentes, mejorar la forma en la que su información es almacenada y ayudar en la toma de decisiones estratégicas por medio de reportes basados en información confiable y actualizada.

1.4 Objetivos del Proyecto

1. Desarrollar un Tablero de Control para el primer evento de la empresa en el cual se puede monitorear la asistencia del personal entre los distintos niveles jerárquicos de la empresa.
2. Desarrollar un módulo para dar de alta y modificar la información del personal de la empresa para el departamento de JE.
3. Desarrollar un módulo para dar de alta y modificar la información del personal de la empresa para el departamento de ESTER.
4. Desarrollar una Plantilla Base para la creación de futuras plantillas laborales.
5. Desarrollar un Tablero de Control para monitorear la asistencia y la realización de las tareas asignadas a los empleados que fueron dados de alta a través de la Plantilla Laboral ESTER.
6. Desarrollar un módulo para dar de alta y modificar la información del personal de la empresa para el departamento de ENYRE.
7. Desarrollar un módulo para dar de alta y modificar la información del personal de la empresa para el departamento de 11-15.
8. Dividir la Plantilla Laboral JE en dos nuevas Plantillas Laborales “J” y “E”.
9. Desarrollar un Tablero de Control para monitorear la asistencia y la realización de las tareas asignadas a los empleados que fueron dados de alta a través de la Plantilla Laboral ENYRE.
10. Desarrollar un Tablero de Control para monitorear la asistencia y la realización de las tareas asignadas a los empleados que fueron dados de alta a través de la Plantilla Laboral 11-15.

2. Problemas planteados a resolver

La empresa PC HOME actualmente cuenta con pequeño framework recién implementado, el cual solo tiene pocas herramientas a su disposición. La función principal es la de soportar todas las necesidades tecnológicas de la empresa, sin embargo, solo está siendo utilizado para generar reportes para los altos mandos. Se están organizando dos eventos futuros de suma importancia para los objetivos de la empresa y es necesario que el framework, cuente con las funciones necesarias para ayudar en la organización tanto antes del evento como en el mismo día.

Otra problemática es la forma en la que se maneja la información. La empresa cuenta con un departamento de encuestas y servicio al cliente donde los trabajadores capturan la información y la suben al sistema. El problema principal es que, muchas veces se dan de alta clientes que ya se encuentran en el sistema por algún error de escritura o de búsqueda de parte de los trabajadores y, los campos que llenan también están sujetos al criterio personal del trabajador en lugar de un formato especial.

Por último la problemática más preocupante es el almacenamiento de la información. El framework cuenta con una base de datos en donde se almacena toda la información, desafortunadamente esta no se creó anticipándose a las futuras necesidades tecnológicas de la empresa. Esto ocasiona que las consultas sean largas y lentas, lo que a su vez disminuye el rendimiento del servidor y ralentiza todos los procesos de la empresa. Además complica la implementación de nuevos módulos que requieren de información que no está estructurada y los cuales, para funcionar correctamente, deben empujar la mala estructura de la base de datos lo que sólo empeora la situación actual.

Esto requiere que se haga un análisis completo de la situación actual y de las necesidades futuras especialmente las que son fundamentales para ambos eventos. Se requiere también una evaluación de las correcciones que se pueden hacer rápidamente y de los lineamientos que se deben seguir para evitar caer en los mismos problemas. También es recomendado estandarizar el proceso de captura por parte de los trabajadores y establecer un formato para el manejo adecuado de la base de datos.

2.1 Alcances en la solución de problemas

Es posible brindarle a la empresa un sistema que pueda cumplir con todas sus necesidades tecnológicas. Implementando nuevos módulos al ya existente framework y con ello, podemos asegurarnos de que todos los procesos de la empresa se alinean adecuadamente a las tecnologías de la información con las que cuenta.

El principal impacto que podemos tener dentro de la empresa es la optimización de sus procesos, ya sea en el adecuado manejo y almacenamiento de su información o en la optimización de tiempos dentro del sistema mismo. Esto se puede conseguir mejorando la estructura actual de la base de datos y a su vez las consultas existentes, mejorando el tiempo de respuesta.

Otro impacto favorable es el de brindarle a los trabajadores una herramienta que reduzca los errores que estos pueden cometer a la hora de capturar información ya que esto compromete su veracidad y la toma de decisiones adecuadas.

Se eliminará el uso de hojas de cálculo para manejar la información de la empresa. Con la finalización del proceso de mejoramiento del framework el uso de hojas de cálculo para almacenar información y distribuirla en los departamentos se volverá obsoleto.

2.2 Limitaciones en la solución de problemas

Debido a que la mayor parte de la problemática dentro de la empresa depende de sus trabajadores y dado que el objetivo principal de nuestra labor es asegurarnos de que el personal asista y cumpla con sus funciones dentro de los eventos, es importante recalcar que el éxito de este proyecto no depende tanto de la estructura del sistema como tal, sino más bien del uso adecuado que le den los trabajadores.

Esto dificulta un poco el potencial éxito que se pueda tener por realizar un sistema adecuado y que se ajuste perfectamente a lo que la empresa requiera. Es necesario asegurarse de que los empleados reciban una capacitación completa sobre el uso del sistema y los estándares necesarios para su correcto funcionamiento.

Por otra parte una limitación preocupante, es la de mantener el mismo formato de los módulos actuales en los módulos futuros, debido a que los trabajadores ya están acostumbrados a la vista y funcionalidad del framework, cualquier modificación podría mermar su eficacia y terminar perjudicando el trabajo.

Por último la base de datos es la limitación principal, la falta de visión por parte de sus desarrolladores dificultará cualquier trabajo realizado sobre la misma y ralentizará la implementación de nuevos módulos en el sistema. Sin mencionar el tiempo que se deberá tomar para mejorarla y mejorar los módulos que funcionan en base a la vieja estructura.

3. Fundamentos teóricos de las herramientas y conocimientos aplicados

- **Desarrollo Web** se refiere en términos generales a las tareas relacionadas con el desarrollo de Sitios Web para ser alojados por intranet o Internet. El proceso de desarrollo Web incluye el diseño Web, desarrollo de contenidos Web, scripting cliente/servidor y la configuración de seguridad de red, entre otras tareas [3].
- **Visual Basic .NET** es un lenguaje de programación orientado a objetos que se puede considerar una evolución de Visual Basic implementada sobre el framework .NET. La gran mayoría de programadores de VB.NET utilizan el entorno de desarrollo integrado Microsoft Visual Studio en alguna de sus versiones aunque existen otras alternativas, como SharpDevelop [4].
- **ASP (Active Server Pages)** es una tecnología de Microsoft del tipo "lado del servidor" para páginas web generadas dinámicamente, que ha sido comercializada como un anexo a Internet Information Services (IIS). Está limitada a funcionar solo en Microsoft Windows, pues requiere el servidor IIS [5].
- **CSS** Hojas de Estilo en Cascada, en inglés Cascading Style Sheets, es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos [6].
- **SQL** es un lenguaje de computación que sirve para trabajar con conjuntos de datos y las relaciones entre ellos. Los programas de bases de datos relacionales, como Microsoft Office Access, usan SQL para trabajar con datos. A diferencia de muchos lenguajes de computación, SQL no es difícil de leer y entender, incluso para un usuario inexperto. Al igual que muchos lenguajes de computación, SQL es un estándar internacional que es reconocido por organismos de estándares, como ISO y ANSI [7].

- **Visual Studio** es un conjunto completo de herramientas de desarrollo para la generación de aplicaciones web ASP.NET, Servicios Web XML, aplicaciones de escritorio y aplicaciones móviles. Visual Basic, Visual C# y Visual C++ utilizan todos el mismo entorno de desarrollo integrado (IDE), que habilita el uso compartido de herramientas y hace más sencilla la creación de soluciones en varios lenguajes. Asimismo, dichos lenguajes utilizan las funciones de .NET Framework, las cuales ofrecen acceso a tecnologías clave para simplificar el desarrollo de aplicaciones web ASP y Servicios Web XML [8].
- **SQL Server Management Studio (SSMS)** es un entorno integrado para obtener acceso, configurar, administrar y desarrollar todos los componentes de SQL Server. SSMS combina un amplio grupo de herramientas gráficas con una serie de editores de script enriquecidos que permiten a desarrolladores y administradores de todos los niveles obtener acceso SQL Server.

SSMS combina las características del Administrador corporativo, el Analizador de consultas y Analysis Manager, herramientas incluidas en versiones anteriores de SQL Server, en un único entorno. Además, SSMS funciona con todos los componentes de SQL Server, como Reporting Services y Integration Services. De este modo, los desarrolladores pueden disfrutar de una experiencia familiar y los administradores de bases de datos disponen de una herramienta única y completa que combina herramientas gráficas fáciles de usar con funciones avanzadas de scripting [9].

4. Procedimientos empleados y actividades desarrolladas

Debido a políticas de la empresa, no se muestran imágenes del sistema en ejecución, por lo que las imágenes que se mostrarán, serán de la estructura de la página en general solamente (figura 4.1). También se agregará código cuando sea necesario, sin embargo se podrán llegar a tapar líneas de código para evitar revelar información importante y/o de carácter confidencial.

La primera actividad que se realizó fue desarrollar un Tablero de Control para el primer evento de la empresa, en el cual se pudiera monitorear la asistencia del personal en distintas regiones y niveles jerárquicos. Una característica importante era el poder contactar a los líderes de departamento o gerentes, a través de mensajes a su teléfono, para informarles qué trabajadores no estaban cumpliendo con sus tareas o no habían asistido al evento.

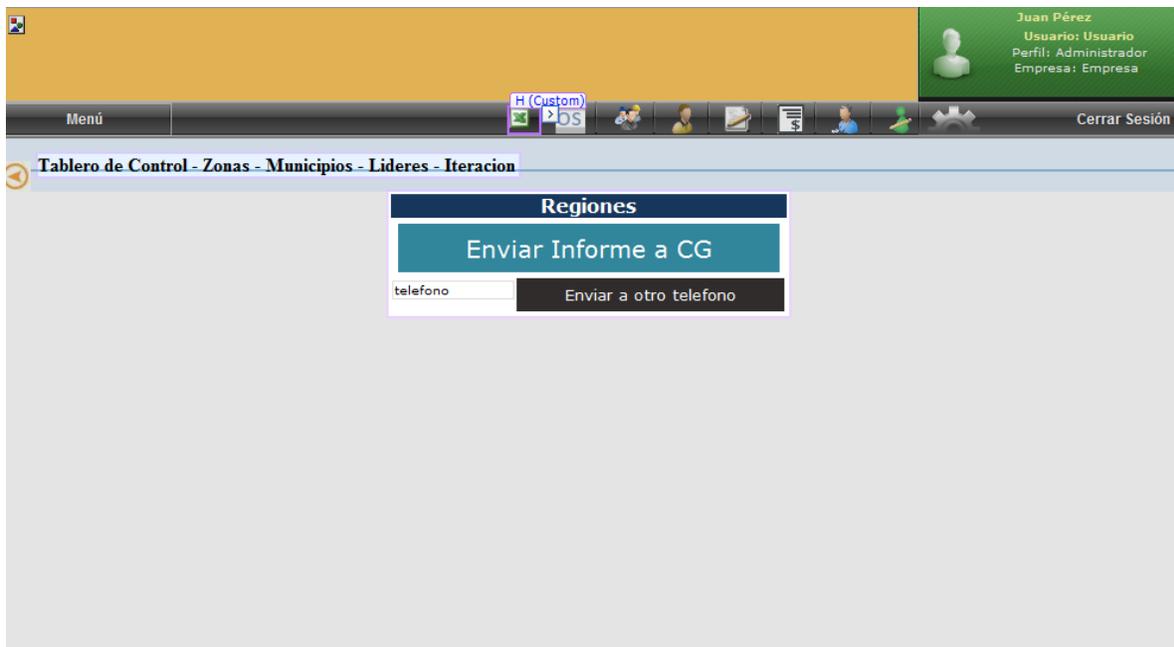


Figura 4.1. Estructura Básica del Tablero de Control

En la figura 4.1 se pueden observar algunas de las características principales del Tablero de Control como la opción de mandar avisar a los Coordinadores Generales de la situación actual del tablero o a algún otro teléfono, también se puede observar a grandes rasgos cómo funciona la navegación entre niveles. La tabla se llena al momento de entrar a la página por lo que no es posible apreciarla en esta figura.

Ya que la empresa necesitaba ver la información en distintos niveles, los cuales podían llegar a ser hasta 5, se decidió realizar el proyecto de tal forma que todo el proceso de cambio en el tablero se manejara en una sola página.

Esto representó un problema ya que en algunos casos la información que debía mostrarse en cada nivel tenía una estructura diferente, especialmente en los cambios de lugares a puestos. Para solucionar esta problemática se optó por generar columnas estándar con la información que será constante en todos los niveles y también generar columnas proxy las cuales cambiaran dependiendo del nivel.

A continuación se puede observar la distribución de las columnas en ASP (figura 4.2):

```

<Columns>
  <asp:TemplateField>
    <HeaderTemplate></HeaderTemplate>
    <ItemTemplate>
      <asp:HyperLink ID="HplSuperGrupo" runat="server" font-size="14px" Text="<%=# Eval("SuperGrupo")%>"></asp:HyperLink>
    </ItemTemplate>
  </asp:TemplateField>

  <asp:TemplateField>
    <HeaderTemplate>Participacion</HeaderTemplate>
    <ItemTemplate>
      <asp:HyperLink ID="HplParticipacion" runat="server" Text="<%=# Eval("Participacion", "{0:N0}")%>"></asp:HyperLink>
    </ItemTemplate>
  </asp:TemplateField>

  <asp:TemplateField>
    <HeaderTemplate>Avance</HeaderTemplate>
    <ItemTemplate>
      <asp:HyperLink ID="HplAvance" runat="server" Text="<%=# Eval("Avance")%>"></asp:HyperLink>
    </ItemTemplate>
  </asp:TemplateField>

  <asp:TemplateField>
    <HeaderTemplate>Militancia</HeaderTemplate>
    <ItemTemplate>
      <asp:HyperLink ID="HplMilitantes" runat="server" Text="<%=# Eval("Militantes", "{0:N0}")%>"></asp:HyperLink>
    </ItemTemplate>
  </asp:TemplateField>

  <asp:TemplateField>
    <HeaderTemplate>JGM</HeaderTemplate>
    <ItemTemplate>
      <asp:HyperLink ID="HplJGM" runat="server" Text="<%=# Eval("JGM", "{0:N0}")%>"></asp:HyperLink>
    </ItemTemplate>
  </asp:TemplateField>

  <asp:TemplateField>
    <HeaderTemplate>Contienda</HeaderTemplate>
    <ItemTemplate>
      <asp:HyperLink ID="HplContienda" runat="server" Text="<%=# Eval("Contienda")%>"></asp:HyperLink>
    </ItemTemplate>
  </asp:TemplateField>

  <asp:TemplateField>
    <HeaderTemplate>Sin Participar</HeaderTemplate>
    <ItemTemplate>
      <asp:HyperLink ID="HplSinParticipar" runat="server" Text="<%=# Eval("SinParticipar", "{0:N0}")%>"></asp:HyperLink>
    </ItemTemplate>
  </asp:TemplateField>

  <asp:TemplateField>
    <HeaderTemplate></HeaderTemplate>
    <ItemTemplate>
      <asp:LinkButton ID="btnEnviarMensaje" CssClass="btn-default btn" runat="server" text="Enviar Mensaje" visible="true"/>
      <asp:Hyperlink ID="Telefono" runat="server" Text="<%=# Eval("Telefono")%>" style="text-align:center;" visible="true">Telefono</asp:HyperLink>
    </ItemTemplate>
  </asp:TemplateField>

```

Figura 4.2. Columnas del Tablero de Control

Como se puede observar en la figura algunas de las columnas no tienen texto por defecto, como en el caso de la primera (*ID=HplSuperGrupo*), esta columna es un ejemplo de una columna proxy. El resto de las columnas mantienen su título en cada nivel ya que estas contienen información básica (asistencia, porcentaje de asistencia, total de trabajadores, faltantes, etc). Esta información se ajusta automáticamente al nivel en el que se encuentra, por ejemplo si nos encontramos en el nivel de la Región Norte los números abarcan todos los trabajadores de la Región Norte mientras que si nos encontramos en el municipio de Hermosillo los datos solo reflejan el total de Hermosillo. También se asignaron permisos para que los líderes de cada nivel solo pudieran ver el tablero a partir de su nivel.

Debido a que se decidió manejar cada nivel dentro de una misma página optamos por utilizar una estructura de control para cada nivel independiente. El proceso es simple, al cargarse la página se identifica el nivel utilizando el rol del usuario actual o los parámetros enviados por un nivel anterior. Una vez identificado el nivel se ejecuta una consulta a la base de datos específica para ese nivel y se cargan los datos. Al cargarse los datos se generan los títulos adecuados (columnas proxy), se llenan las filas y se crean las ligas a los otros niveles con los parámetros necesarios para que la página identifique cual es el nivel que se debe cargar ya sea inferior o superior (figura 4.3). Se tuvo especial cuidado en la navegación entre niveles para poder facilitar el trabajo de los encargados del tablero.

```
Select Case tipoUsuario
  Case "1052", "1"
    pagina = "1053"
    superGrupo = "REGION " & cvePrimaria
    Dim zona As String = e.Row.DBCampo("SuperGrupo").ToString()
    If e.Row.RowIndex = 0 Then
      e.Row.FCHyperLink("HplSuperGrupo").Text = "TOTAL"
      e.Row.FCHyperLink("HplAvance").Text = Math.Round((participacion / militantes) * 100, 2) & "%"
      If participacion > 0 Then
        e.Row.FCHyperLink("HplContienda").Text = Math.Round((JGM / participacion) * 100, 2) & "%"
      Else
        e.Row.FCHyperLink("HplContienda").Text = "0%"
      End If
    Else
      e.Row.FCHyperLink("HplSuperGrupo").Text = "REGION " & zona
      e.Row.FCHyperLink("HplSuperGrupo").Font.Size = 16
      e.Row.FCHyperLink("HplSuperGrupo").NavigateUrl = String.Format("/M2015/TableroControl.aspx?Tipo=" & pagina & "&Usuario=" & Usuario & "&CvePrimaria=" & cvePrimaria & "&SuperGrupo=" & superGrupo)
      e.Row.FCLiteral("LtlIr").Text = "<a href="" />M2015/ListadoMilitantes.aspx?CveRegion=" & cvePrimaria & "">Ir</a>"
    End If
  Case "1053"
    pagina = "1054"
    Fila.BackColor = System.Drawing.Color.FromArgb(204, 153, 0)
    GridView1.BackColor = System.Drawing.Color.FromArgb(255, 255, 204)
    cvePrincipal = e.Row.DBCampo("cvePrincipal").ToEntero()
    If e.Row.RowIndex = 0 Then
      e.Row.FCHyperLink("HplSuperGrupo").Text = "TOTAL"
      e.Row.FCHyperLink("HplAvance").Text = Math.Round((participacion / militantes) * 100, 2) & "%"
      If participacion > 0 Then
        e.Row.FCHyperLink("HplContienda").Text = Math.Round((JGM / participacion) * 100, 2) & "%"
      Else
        e.Row.FCHyperLink("HplContienda").Text = "0%"
      End If
    Else
      superGrupo = titulo & "/" & e.Row.DBCampo("SuperGrupo").ToString()
      e.Row.FCHyperLink("HplSuperGrupo").NavigateUrl = String.Format("/M2015/TableroControl.aspx?Tipo=" & pagina & "&Usuario=" & Usuario & "&CvePrimaria=" & cvePrimaria & "&SuperGrupo=" & superGrupo & "&Zona=" & zona)
      btnEnviarCG.Visible = True
      label3.NavigateUrl = String.Format("/M2015/TableroControl.aspx")
    End If
End Select
```

Figura 4.3. Estructura de Control

Como parte del tablero de control se creó una vista especial para el nivel específico de los trabajadores a cargo de algún líder. Este nivel se enfoca en mostrar los datos del trabajador incluyendo si ya asistió o no al evento. También se mantiene la opción de mandar un mensaje a los superiores de estos trabajadores y la opción de filtrar a los trabajadores por asistencia.

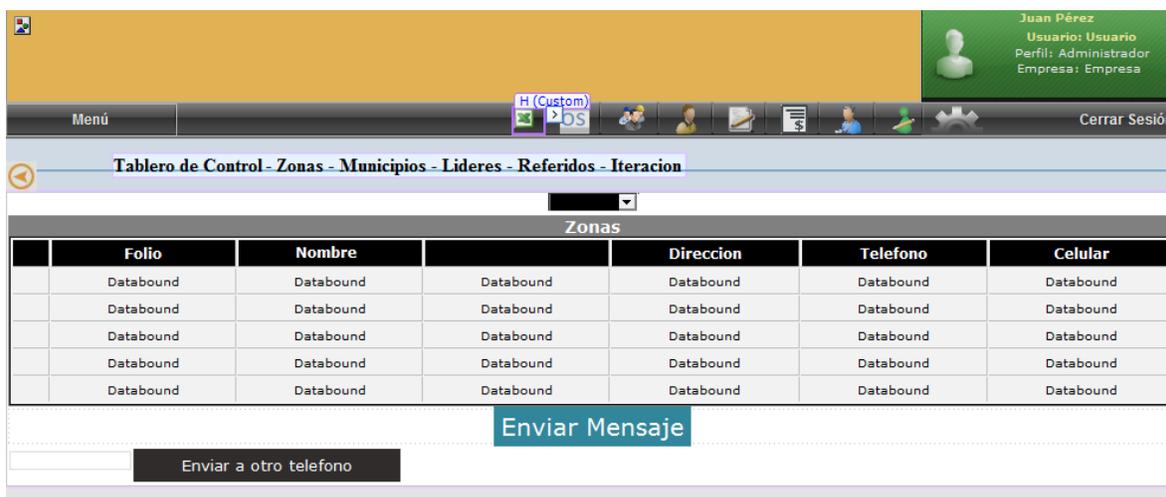


Figura 4.4. Tablero de Control Nivel Trabajadores

Por último se creó una vista más para mostrar todos los trabajadores de la empresa con el simple objetivo de filtrar por asistencia y mostrar los líderes responsables de ellos.

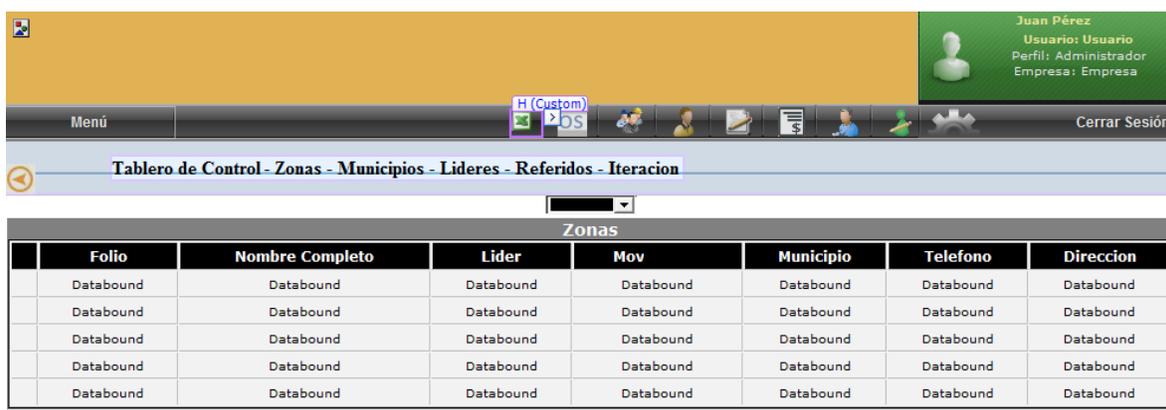


Figura 4.5. Listado de los Trabajadores

La segunda actividad que se nos pidió realizar fue la de crear una plantilla laboral que pudiera remplazar la actual. Esta fue la primera de muchas plantillas laborales y marcó la pauta y los cimientos para las siguientes. La plantilla laboral con la que la empresa estaba trabajando, aunque funcional, dejaba mucho que desear. Su funcionalidad base era dar de alta datos personales de los trabajadores, sin embargo, no tenía la capacidad de separar a los trabajadores por departamento y todos los departamentos tenían acceso a ella. Esto representaba un problema ya que cualquier departamento podía dar de alta a trabajadores con puestos de otros departamentos.

Se decidió mantener el mismo formato de la plantilla laboral actual para la nueva plantilla, de esta manera el cambio no tendría gran impacto en los capturistas de información. La plantilla laboral quedó de la siguiente manera:

Usuario: [redacted]
Perfil: JE_Admin

Menú Cerrar Sesión

Plantilla Laboral JE

Datos Generales

Apellido Paterno: (*) Apellido Materno: (*) Nombre(s): (*)
Fecha Nacimiento: Sexo: Masculino ▼ IFE: (*)
Teléfono: Celular: Correo:

Calle: Número: No. Interior:
Entre Calle: y
Colonia: Código Postal:
Seccion: ▼ Municipio: ▼

Datos de asignación

Fecha de Ingreso: (*) Superior:
Puesto: ▼

Figura 4.6. Plantilla Laboral JE

Como se puede observar la plantilla laboral (figura 4.6) tiene una funcionalidad relativamente básica. La sección **Datos Generales** muestra la información general del empleado y nos permite modificarla. Esta información es la que se dio de alta al ingresar al empleado por primera vez a la base de datos, estos campos se utilizan en su mayoría para ver y modificar la información existente, no para agregarla. La sección **Datos de Asignación** es la que se encarga de ubicar al trabajador por departamento y posteriormente asignarle un puesto y un superior.

El proceso para dar de alta el personal por departamentos es el siguiente:

1. Dar de alta al usuario dentro de la empresa (esto implica que el trabajador exista en la base de datos y cuente con una `CveEmpleado`).
2. Ingresar a la plantilla laboral, la cual como primera opción solicita el nombre del empleado o la `CveEmpleado` para realizar la búsqueda en la base de datos. (Por cuestiones de seguridad no se nos permitió mostrar esa imagen)
3. Una vez que se carga el usuario (figura 4.6) los datos de asignación proceden a cargarse. Si el usuario ya estaba registrado y está siendo consultado para cambio sus datos se cargan automáticamente, de lo contrario los datos aparecen vacíos.
4. Por último se procede a asignarle una fecha de alta, su puesto y su superior.

Conforme se fue trabajando en el proyecto la complejidad de la plantilla laboral creció considerablemente, sin embargo para poder darle seguimiento al objetivo número 4 (Crear una plantilla base) en lugar de mostrar el producto final se está mostrando el proceso completo.

La magia de la plantilla laboral ocurre en el código, la plantilla cambia completamente dependiendo del puesto que se está cargando. Puede ir desde mostrar todos los municipios del estado en forma de un `CheckList` hasta hacer un camino de `DropDownLists` dependientes unos de otros.

Para poder tener este nivel de flexibilidad se requiere tener un control estricto de los datos que se cargan y se guardan. Para esto se crearon una serie de métodos que reducen en gran medida los potenciales errores que pudieran llegar a surgir. El primero de estos métodos es el `Método CargaDatos`. Este método se encarga de asignar todos los valores básicos a las variables, de esta manera nos aseguramos que estén todos los valores necesarios para ejecutar los métodos siguientes. Este método se puede apreciar en la figura 4.7.

Como se aprecia en la figura las variables no solo almacenan la información, sino que a su vez cargan la información en la plantilla, así podemos observar inmediatamente si hubo algún error al cargar los datos. La segunda parte del método es la que identifica el puesto y carga el resto de la plantilla dependiendo de cuál sea. Este proceso se puede ver en la figura 4.8.

```

Private Sub CargaDatos()
    If ObjProspectos.DtResultados.Rows.Count > 0 Then

        TxtCveProspecto.Text = ObjProspectos.CveProspecto.ToString
        TxtApellidoPaterno.Text = ObjProspectos.ApellidoPaterno
        TxtApellidoMaterno.Text = ObjProspectos.ApellidoMaterno
        TxtNombres.Text = ObjProspectos.Nombres
        TxtFechaNacimiento.Text = ObjProspectos.FechaNacimiento.ToString

        TxtCalle.Text = ObjProspectos.Calle
        TxtNumero.Text = ObjProspectos.Numero
        TxtNumeroInterior.Text = ObjProspectos.NumeroInterior
        TxtEntreCalle1.Text = ObjProspectos.EntreCalle1
        TxtEntreCalle2.Text = ObjProspectos.EntreCalle2
        TxtIFE.Text = ObjProspectos.IFE

        DdlSexo.SelectedValue = ObjProspectos.Sexo.ToString
        TxtColonia.Text = ObjProspectos.Colonia
        TxtCodigoPostal.Text = ObjProspectos.CodigoPostal.ToString
        CargaMunicipios(DdlMunicipios, 26)
        DdlMunicipios.SelectedValue = ObjProspectos.CveMunicipio.ToString

        TxtTelefono.Text = ObjProspectos.Telefono
        TxtCelular.Text = ObjProspectos.Celular
        TxtCorreo.Text = ObjProspectos.Correo
        TxtEmpresaNombre.Text = ObjProspectos.EmpresaNombre
        TxtEmpresaGiro.Text = ObjProspectos.EmpresaGiro
        TxtEmpresaSector.Text = ObjProspectos.EmpresaSector
        TxtEmpresaPagina.Text = ObjProspectos.EmpresaPagina
        TxtEmpresaCalle.Text = ObjProspectos.EmpresaCalle
        TxtEmpresaNumero.Text = ObjProspectos.EmpresaNumero
        TxtEmpresaColonia.Text = ObjProspectos.EmpresaColonia
        TxtEmpresaCodigoPostal.Text = ObjProspectos.EmpresaCodigoPostal

        DdlEmpresaPaises.SelectedValue = ObjProspectos.CveEmpresaPais.ToString
        CargaEstados(DdlEmpresaEstados, DdlEmpresaPaises.SelectedValue.ToEntero)
        DdlEmpresaEstados.SelectedValue = ObjProspectos.CveEmpresaEstado.ToString
        CargaMunicipios(DdlEmpresaMunicipios, DdlEmpresaEstados.SelectedValue.ToEntero)
        DdlEmpresaMunicipios.SelectedValue = ObjProspectos.CveEmpresaMunicipio.ToString

        TxtEmpresaTelefono.Text = ObjProspectos.EmpresaTelefono
        TxtEmpresaFax.Text = ObjProspectos.EmpresaFax
        TxtEmpresaCorreo.Text = ObjProspectos.EmpresaCorreo
        TxtEmpresaEspecialidad.Text = ObjProspectos.EmpresaEspecialidad
    
```

Figura 4.7. Metodo CargaDatos – Variables

```

Select Case DdlEquipo.SelectedValue.ToEntero
    Case 1
        CargaPuesto(DdlEquipo.SelectedValue.ToEntero)
    Case 2
        CargaPuesto(DdlEquipo.SelectedValue.ToEntero)
    Case 3
        CargaPuesto(DdlEquipo.SelectedValue.ToEntero)
    Case 4
        CargaPuesto(DdlEquipo.SelectedValue.ToEntero)
End Select

Select Case DdlPuesto.SelectedValue.ToEntero
    Case 1
        DdlDistrito.Enabled = False
        DdlZona.Enabled = False
        DdlSeccion.Enabled = False
        DdlSeccion.Enabled = False

    Case 4
        'CargaDistrito()
        CargaDistritoInver2(ObjProspectos.CveCasillaAsignada.ToEntero)
        CargaZona(DdlDistrito.SelectedValue.ToEntero)
        DdlZona.SelectedValue = ObjProspectos.CveCasillaAsignada.ToString
        DdlSeccion.Enabled = False
        DdlSeccion.Enabled = False

    Case 5
        'CargaDistrito()
        CargaDistritoInver(ObjProspectos.CveCasillaAsignada.ToEntero)
        CargaZonaInver(ObjProspectos.CveCasillaAsignada.ToEntero)
        CargaSeccion(DdlZona.SelectedValue.ToEntero)
        CargaSeccionInver(ObjProspectos.CveCasillaAsignada.ToEntero)
        DdlZona.SelectedValue = ObjProspectos.CveCasillaAsignada.ToString
        DdlSeccion.Enabled = False
        DdlSeccion.Enabled = False

    Case 6
        'CargaDistrito()
        CargaDistritoInver(ObjProspectos.CveCasillaAsignada.ToEntero)
        CargaZonaInver(ObjProspectos.CveCasillaAsignada.ToEntero)
        CargaSeccion(DdlZona.SelectedValue.ToEntero)
        CargaSeccionInver(ObjProspectos.CveCasillaAsignada.ToEntero)
        DdlZona.SelectedValue = ObjProspectos.CveCasillaAsignada.ToString
        DdlSeccion.Enabled = False
        DdlSeccion.Enabled = False

    Case 9
        CargaDistritoInver2(ObjProspectos.CveCasillaAsignada.ToEntero)
        CargaZona(DdlDistrito.SelectedValue.ToEntero)
        DdlZona.SelectedValue = ObjProspectos.CveCasillaAsignada.ToString
        DdlSeccion.Enabled = False
        DdlSeccion.Enabled = False
    
```

Figura 4.8. Metodo CargaDatos – Puesto

Aunque posteriormente se creó una plantilla para cada departamento (J y E) al inicio se nos pidió que la plantilla abarcara ambos departamentos ya que compartían funciones muy similares. Para identificar el departamento se requería de un DropDownList que permitiera seleccionar el departamento y posteriormente cargar los puestos específicos de ese departamento. Estos DropDownList se muestran en la figura 4.9.

```

<td>Equipo Trabajo:</td>
<td>
    <asp:DropDownList ID="DdlEquipo" AutoPostBack="true" DataTextField="EquipoTrabajo" DataValueField="CveEquipoTrabajo" Width="70px" runat="server"></asp:DropDownList>(*)
    <asp:UpdatePanel ID="UpdateEquipo" runat="server" UpdateMode="Conditional" RenderMode="Inline">
        <ContentTemplate>
        </ContentTemplate>
    </asp:UpdatePanel>
</td>
<td>Puesto:</td>
<td>
    <asp:DropDownList ID="DdlPuesto" AutoPostBack="true" DataTextField="Puesto" DataValueField="CvePuesto" Width="200px" runat="server"></asp:DropDownList>(*)
    <asp:UpdatePanel ID="UpdatePuesto" runat="server" UpdateMode="Conditional" RenderMode="Inline">
        <ContentTemplate>
        </ContentTemplate>
    </asp:UpdatePanel>
</td>
</td>

```

Figura 4.9. Plantilla Laboral – DropDownLists

La lógica detrás de la adaptación de la plantilla y su versatilidad recae dentro de los métodos que detectan los cambios en los DropDownList (SelectedIndexChanged). Este método no solo se encarga de cambiar lo visual, también llena DropDownLists, CheckBoxLists y TextBoxes y a su vez al agregar información a estos dispara los métodos SelectedIndexChanged, generando una cadena de cambios y adaptaciones en la página. Este proceso se puede observar un poco en la figura 4.10.

```
Protected Sub DdlEquipo_SelectedIndexChanged(sender As Object, e As EventArgs) Handles DdlEquipo.SelectedIndexChanged
    Select Case DdlEquipo.SelectedValue.ToEntero
        Case 1
            DdlPuesto.Enabled = True
            CargaPuesto(DdlEquipo.SelectedValue.ToEntero)
        Case 2
            DdlPuesto.Enabled = True
            CargaPuesto(DdlEquipo.SelectedValue.ToEntero)
        Case 3
            DdlPuesto.Enabled = True
            CargaPuesto(DdlEquipo.SelectedValue.ToEntero)
        Case 4
            DdlPuesto.Enabled = True
            CargaPuesto(DdlEquipo.SelectedValue.ToEntero)
    End Select
End Sub

Protected Sub DdlPuesto_SelectedIndexChanged(sender As Object, e As EventArgs) Handles DdlPuesto.SelectedIndexChanged
    Select Case DdlPuesto.SelectedValue.ToEntero
        Case 1
            DdlDistrito.Enabled = False
            DdlZona.Enabled = False
            DdlSeccion.Enabled = False
            DdlCasilla.Enabled = False
        Case 2
            DdlDistrito.Enabled = True
            DdlZona.Enabled = False
            DdlSeccion.Enabled = False
            DdlCasilla.Enabled = False
        Case 4
            DdlDistrito.Enabled = True
            DdlZona.Enabled = True
            DdlSeccion.Enabled = False
            DdlCasilla.Enabled = False
        Case 11
            DdlDistrito.Enabled = True
            DdlZona.Enabled = True
            DdlSeccion.Enabled = False
            DdlCasilla.Enabled = False
        Case 5
            DdlDistrito.Enabled = True
            DdlZona.Enabled = True
            DdlSeccion.Enabled = True
            DdlCasilla.Enabled = True
        Case 6
            DdlDistrito.Enabled = True
            DdlZona.Enabled = True
            DdlSeccion.Enabled = True
            DdlCasilla.Enabled = True
    End Select
End Sub
```

Figura 4.10. SelectedIndexChanged

En la figura 4.10 podemos ver como en DdlEquipo_SelectedIndexChanged se llama al método cargar puesto y se le envía como parámetro el valor del Ddl (DropDownList para futura referencia). El método cargar puesto corre una consulta enviando el valor del Equipo de Trabajo y obteniendo como resultado los puestos pertenecientes a ese equipo. Una vez obtenidos los resultados, estos se cargan al Ddl del puesto y se activa el SelectedIndexChanged de este. La sintaxis del método CargarPuesto es la siguiente:

```
Private Sub CargaPuesto(CveEquipo As Integer)
    Dim Puesto As New ClsProsJE
    If Puesto.GetPuesto(CveEquipo) Then
        DdlPuesto.DataSource = Puesto.DtResultados
        DdlPuesto.DataBind()
    End If
    Puesto.Dispose()
    ' ddl.DataSource = GlobalDTGralPuestos(CveEstado) :
    ddl.DataBind()
End Sub
```

De esta manera se pueden llenar y cargar automáticamente todos los campos necesarios para alimentar la plantilla y obtener algo parecido a la figura 4.11 pero a mayor escala y con una infinidad de estructuras y datos.

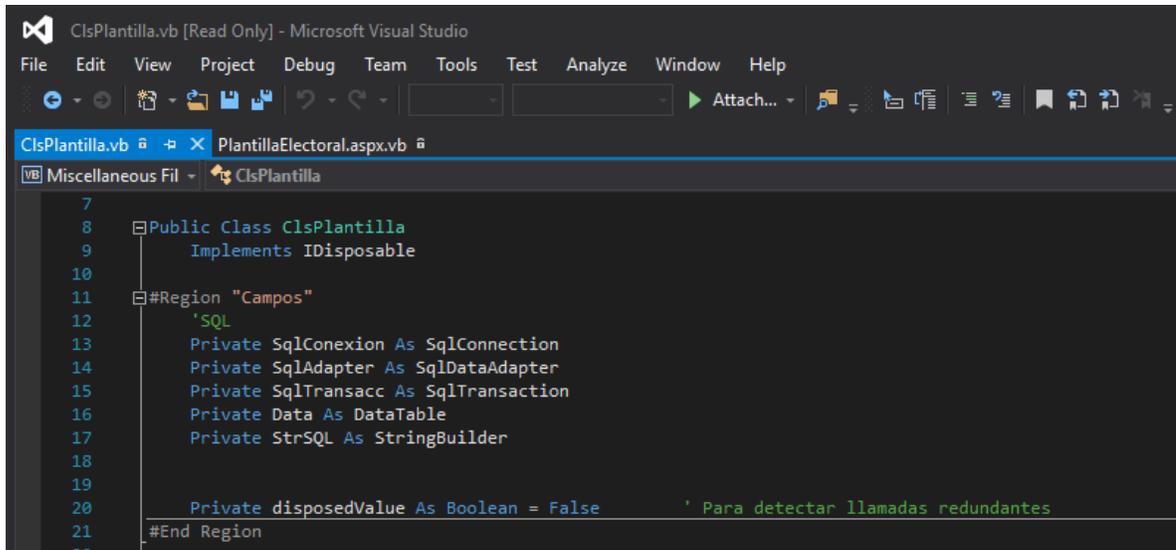
The image shows a form titled "Datos de asignación" with the following fields:

- Fecha de Ingreso: A text input field with a calendar icon and an asterisk (*) indicating it is required.
- Puesto: A dropdown menu with "Responsable" selected.
- Superior: A dropdown menu with "LOPEZ NORIEGA ALEJ" selected.
- Zona: A dropdown menu with "NORTE" selected.

Figura 4.11. Datos de Asignación

Este proceso es la base de los futuros desarrollos de Plantillas Laborales y estos continúan creciendo en complejidad y en capacidad de adaptación a los cambios. En este caso esta funcionalidad fue suficiente para cumplir con las necesidades del departamento JE. Su implementación fue un éxito y esto dio paso a que se nos solicitara el objetivo número 3, la plantilla laboral para el departamento de ESTER.

El caso de la plantilla laboral de ESTER fue muy similar a la plantilla de JE, la única diferencia siendo los puestos que se manejaron. Como la estructura era la misma se decidió utilizar la plantilla que ya se tenía para el departamento de JE y enfocarse en la sección de los datos de asignación. Se crearon los procedimientos almacenados pertinentes y sus respectivas modificaciones a las clases. Como ambas plantillas son prácticamente iguales utilizare este apartado para mostrar la clase ClsPlantilla (figura 4.12), la cual se encarga de manejar todos los métodos para las plantillas laborales y su conexión a la bases de datos.



```
7
8 Public Class ClsPlantilla
9     Implements IDisposable
10
11 #Region "Campos"
12     'SQL
13     Private SqlConexion As SqlConnection
14     Private SqlAdapter As SqlDataAdapter
15     Private SqlTransacc As SqlTransaction
16     Private Data As DataTable
17     Private StrSQL As StringBuilder
18
19
20     Private disposedValue As Boolean = False ' Para detectar llamadas redundantes
21 #End Region
22
```

Figura 4.12. ClsPlantilla

Como en todo sistema las clases son la parte más fundamental, en este framework no es diferente. La clase se encarga de almacenar todas las variables necesarias, los métodos que le asignan valor a esas variables, y las llamadas a los procedimientos almacenados en nuestra base de datos. Lo primero que podemos apreciar es la definición de las variables, debido a la forma en la que .NET maneja su destrucción de objetos cualquier variable que no sea declarada como una propiedad es destruida inmediatamente al cargarse la página. Para tener acceso a datos una vez cargada la página utilizamos las propiedades, las cuales se mantienen activas hasta que la página se cierra. Para mantener una limpieza de código se utilizaron comentarios y definición de regiones para separar las distintas secciones de la clase.

Un ejemplo de las propiedades se muestra a continuación (figura 4.13).

```
Public Property CveProspectoPadreJE As Integer
Public Property CvePuesto As Integer
Public Property FechaIngJE As Date
Public Property CveProspecto1 As Integer
Public Property Status As Integer
Public Property CveTipo As Integer
Public Property Tipo As Integer
Public Property CveUsuarioSistema As Integer
Public Property Referido As String
Public Property TelReferido As String
Public Property NuevaCveprospecto As Integer
Public Property SeccionPredominante As Integer
'***** Propiedades EnyRe *****
Public Property Zona As Integer
Public Property MunicipioPuesto As Integer
Public Property SeccionPuesto As Integer
Public Property TipoCasilla As Integer
Public Property Casilla As Integer
Public Property SeleccionMultiple As String

'***** Propeidades J [REDACTED] *****
Public Property SeccionJE As Integer

'***** Propiedades E [REDACTED] *****
Public Property DistritoJE As String
Public Property [REDACTED]JE As String
Public Property SeccionE As String
Public Property Taller As Integer
Public Property Integracion As Integer
Public Property Estrategia As Integer
Public Property Ruta As String

End Region
```

Figura 4.13. Propiedades de ClsPlantilla

Los comentarios dividen las variables específicas al tipo de plantilla, las variables que no se encuentran en ninguna división corresponden a las variables generales que toda plantilla comparte.

La clase maneja los procedimientos almacenados que se utilizan para las plantillas laborales. El framework ya cuenta con una clase que se encarga de la conexión a la base de datos de la cual creamos el objeto con el que llamamos los Procedimientos Almacenados.

La forma en la que se llaman los procedimientos almacenados se muestra a continuación (figura 4.14).

```
Public Function GetPuestos(CvePuesto As Integer, CveProspecto As Integer, CveSuperior As Integer) As Boolean
    Dim Retorno As Boolean = False
    DataAdapter("PA_Esterr_LlenarPuesto")
    With SqlDataAdapter.SelectCommand.Parameters
        .Add("@CvePuesto", SqlDbType.Int).Value = CvePuesto
        .Add("@CveProspecto", SqlDbType.Int).Value = CveProspecto
        .Add("@CveSuperior", SqlDbType.Int).Value = CveSuperior
    End With
    Try
        SqlDataAdapter.Fill(Data)
        Retorno = True
    Catch ex As Exception
        OnError("Ocurrió un error al obtener el Catálogo de los Puestos.", ex, SqlDataAdapter.SelectCommand.Parameters.ParametrosInfo())
    End Try

    Return Retorno
End Function

Public Function GetSuperiores(CvePuesto As Integer) As Boolean
    Dim Retorno As Boolean = False
    DataAdapter("PA_Esterr_GetSuperior")
    With SqlDataAdapter.SelectCommand.Parameters
        .Add("@CvePuesto", SqlDbType.Int).Value = CvePuesto
    End With
    Try
        SqlDataAdapter.Fill(Data)
        Retorno = True
    Catch ex As Exception
        OnError("Ocurrió un error al obtener el Catálogo de Superiores.", ex, SqlDataAdapter.SelectCommand.Parameters.ParametrosInfo())
    End Try

    Return Retorno
End Function
```

Figura 4.14. Métodos de Procedimientos Almacenados

El proceso es sencillo, se crea un método para cada procedimiento almacenado. Las variables que requiere el procedimiento se reciben como parámetros y se insertan a través de la siguiente sintaxis:

```
SqlAdapter.SelectCommand.Parameters.Add(
    "@CvePuesto", SqlDbType.Int).Value = CvePuesto
```

El texto entre comillas con el arroba al inicio es el nombre de la variable que va a recibir el valor, a su lado se le indica el tipo de la variable (en este caso un entero) y por último se le manda el valor a través del parámetro (CvePuesto).

Gracias a la definición de esta clase, para acceder a los resultados de las consultas solo es necesario crear un objeto ClsPlantilla y llamar los métodos correspondientes. Posteriormente solo queda asignar estos resultados a algún tipo de control como un dropdownlist o un checkboxlist de la siguiente manera:

```
<asp:DropDownList ID="DdlPuesto" AutoPostBack="true" DataTextField="Puesto"
DataValueField="CvePuesto" Width="140px" runat="server"></asp:DropDownList>
```

```
Dim ObjPl As New ClsPlantilla
If ObjPl.GetPuesto() Then
    DdlPuesto.DataSource = ObjPl.DtResultados
    DdlPuesto.DataBind() End If
```

En el código de asp le indicamos que el texto a mostrarse en el ddl lo tome de la columna "Puesto" y que su valor sea la columna "CvePuesto" de esta manera al obtener el resultado de la consulta el ddl toma automáticamente los valores de estas columnas sin la necesidad de escribir código extra. Esto se vuelve muy importante ya que los valores que almacenamos en la base de datos son en su mayoría valores de controles asp.

Con las plantillas laborales de JE y ESTER terminadas se inició la planeación de futuras implementaciones de plantillas a más departamentos. Ya que todos estos desarrollos tenían una naturaleza urgente, hubo poco tiempo para planear a futuro sus desarrollos, se recomendó a los superiores que se le diera el tiempo apropiado para hacer un análisis adecuado y una posible plantilla base para acelerar el desarrollo de futuras plantillas. Desafortunadamente no se me permitió dedicarle tiempo a esa tarea y ese desarrollo quedo en espera. Con la decisión de implementar 3 plantillas laborales más se nos permitió retomar la plantilla base e iniciar su análisis.

La tarea fue bastante sencilla, para empezar las plantillas no cambiarían mucho ya que su función es la misma. Esto nos permitió mantener el mismo código de asp reduciendo simplemente los datos de la sección de asignación. Los únicos elementos que se mantuvieron fueron los de Fecha de Asignación, Puesto y Superior. La funcionalidad extra simplemente seria agregada en esa sección en particular y no se requeriría de ninguna modificación.

Lo complicado del proceso fue analizar correctamente el código actual y desechar lo que no se estuviera usando. Se empezó por limpiar todos los datos de asignación y sus respectivos métodos, teniendo mucho cuidado de no alterar funcionalidades generales. El objetivo era obtener una plantilla base que se pudiera ejecutar inmediatamente y que fuera 100% funcional, claro que esta plantilla no se utilizaría para ningún departamento pero utilizando esta metodología nos aseguramos de no llevarnos algún error o bug que pudiera comprometer su funcionamiento más adelante.

Una vez limpiada y probada la nueva plantilla se decidió darle una estructura más adecuada para que el código de la plantilla fuera más fácil de mantener y modificar de ser necesario, además de abrir camino para la parte más importante de este proceso la cual mencionaré más adelante. La estructura se dividió en 7 partes divididas a través de regiones las cuales fueron *Propiedades*, *Carga de datos Generales*, *Carga de los Controles*, *Búsqueda de Prospecto*, *Liberación de Prospecto*, *Funcionalidad de los Controles* y *Actualización de los controles*. Una imagen de la estructura final se puede observar a continuación (figura 4.15).

```

Option Explicit On
Option Strict On

Imports Extensiones
Imports System.Runtime.InteropServices
Imports System.IO
Imports System.Data

Partial Class Esterr_AsignarPuesto
    Inherits ClsPaginas

    Propiedades
    Carga de Datos Generales
    Carga DropDownList
    BUSQUEDA PROSPECTOS
    Liberar Prospecto
    Cambia valores de dropdownlist
    ActualizarPaneles
End Class

```

Figura 4.15. Estructura Plantilla Base

Por último y la parte más esencial del desarrollo de la plantilla base, se optó por especificar la ubicación exacta en la que se debe insertar el código para cada nueva plantilla. De esta manera se asegura la continuidad de la estructura y la velocidad de desarrollo para futuras plantillas. El análisis previo nos permitió identificar las secciones adecuadas para cada nuevo desarrollo sin alterar el código ya existente. Aunado a esto se puede saber exactamente que código es nuevo y se facilita la detección y corrección de errores. Esta modalidad se manejó de la siguiente manera (figuras 4.16 a 4.20):

```

#Region "Cambia valores de dropdownlist"

    Protected Sub DdlPuesto_SelectedIndexChanged(sender As Object, e As EventArgs) Handles DdlPuesto.SelectedIndexChanged
    End Sub

    Protected Sub DdlSuperior_SelectedIndexChanged(sender As Object, e As EventArgs) Handles DdlSuperior.SelectedIndexChanged
    End Sub

    ***** Agregar Cambio en Nuevos DropDownListAqui *****
    *****

#End Region

```

Figura 4.16. Sección Cambios en DropDownList

```

#Region "Carga DropDownList"
    Private Sub CargaEstados(ddl As DropDownList, CvePais As Integer)
        ddl.DataSource = GlobalDTGralEstados(CvePais) : ddl.DataBind()
    End Sub

    Private Sub CargaMunicipios(ddl As DropDownList, CveEstado As Integer)
        ddl.DataSource = GlobalDTGralMunicipios(CveEstado) : ddl.DataBind()
    End Sub

    Private Sub CargaSeccion(ddl As DropDownList)
        ddl.DataSource = GlobalDTGralSecciones : ddl.DataBind()
    End Sub

    '***** Agregar Nuevos DropDownList Aqui *****

    '*****

#End Region

```

Figura 4.17. Sección Carga de DropDownList

```

Request("CveProspecto") IsNot Nothing AndAlso IsNumeric(Request("CveProspecto")) AndAlso Request("CveProspecto").ToEntero
ViewState("CveProspecto") = Request("CveProspecto")

CargaDatos()

cveProspectito = Request("CveProspecto").ToEntero

'***** Cargamos los datos aqui *****

'*****

cargarprospecto()

```

Figura 4.18. Sección Carga de Datos Generales

```

#Region "ActualizarPaneles"
    Protected Sub ActualizarPaneles()

        '***** Agregamos todos los update panels aqui *****

        UPSuperior.Update()
        UPPuesto.Update()

    End Sub
#End Region

```

Figura 4.19. Sección Actualización de Controles

```

ObjPlantilla.Numero = TxtNumero.Text.Trim
ObjPlantilla.NumeroInterior = TxtNumeroInterior.Text.Trim
ObjPlantilla.EntreCalle1 = TxtEntreCalle1.Text.Trim
ObjPlantilla.EntreCalle2 = TxtEntreCalle2.Text.Trim
ObjPlantilla.Colonia = TxtColonia.Text.Trim
ObjPlantilla.CodigoPostal = TxtCodigoPostal.ToEntero
ObjPlantilla.FechaIngJE = TxtFechaIngJE.Text.ToDate
ObjPlantilla.Seccion = DdlSeccion.SelectedValue.ToEntero
ObjPlantilla.CveMunicipio = DdlMunicipios.SelectedValue.ToEntero
ObjPlantilla.CveUsuario = UsuarioActual.CveUsuario.ToEntero

***** Agregar Guardado de Datos de Asignacion *****

*****

ObjPlantilla.GuardarProspecto()
ObjPlantilla.Dispose()

```

Figura 4.20. Sección Liberación de Prospecto

Con la plantilla base hecha de esta manera crear una nueva plantilla solo es cuestión de agregar código nuevo en la sección pertinente y probar que todo funcione como debe de ser. La plantilla base se entregó y fue el cimiento de las 3 siguientes plantillas laborales, además de ayudar en la división de la plantilla JE.

Con las plantillas laborales más fuertes terminadas (JE y ESTER) se empezó con el desarrollo de las herramientas que serían fundamentales a futuro. Al final todo mi trabajo en esta empresa se reduce a la información que se muestra en los tableros de las respectivas plantillas laborales y aunque las plantillas laborales fueron un desarrollo importante para la mejora de los procesos de la empresa, su objetivo principal es el de recopilar la información necesaria para que su desempeño pueda ser fácilmente monitoreado y permita a las autoridades pertinentes tomar decisiones con información en tiempo real.

Teniendo esto en mente se me pidió realizar un tablero de control como el que se hizo en el primer desarrollo pero que solo mostrara información del departamento de ESTER. Su funcionalidad es bastante sencilla, el tablero tiene como objetivo el mostrar en tiempo real el total de personas registradas en el departamento de ESTER que hayan terminado su tarea asignada. El tablero se divide en cinco secciones: el avance actual y el porcentaje de la meta cumplido, el avance actual y el porcentaje de la meta equivalente al 30%, el avance actual y el porcentaje de la meta equivalente al 60%, el avance actual y el porcentaje de la meta equivalente al 90%, el avance actual y el porcentaje de la meta equivalente al 100%.

Además de esto, el avance entre secciones solo debía mostrar los valores antes de cierto límite de tiempo. Por ejemplo para el 30% se mostraba el avance hasta las 10am después de esta hora los números se quedaban estáticos y el avance continuaba en las secciones posteriores, a las 12 de la tarde se detenía la sección del 60%, a las 2 de la tarde, se detenía la sección del 90% y a las 4 de la tarde se detenía la sección del 100% dando por terminado el evento y el control del tablero. Los valores de la primera sección siempre se mantienen activos y son para ver el avance actual y el porcentaje de la meta general que se ha cumplido.

Para el evento de ESTER, debido a que no se esperaba más que el 50% de participación, se impuso una meta. Esta meta fue la que se dividió en los porcentajes de las secciones del tablero y lo que se tenía como objetivo era que antes del tiempo límite de cada sección se alcanzara dicha meta. El tablero fue un éxito y se comportó como se esperaba se apreciaba perfectamente como subían los números y como se alcanzó o no la meta, además de cómo se congelaban las cifras de las secciones al cumplirse su tiempo límite.

Para este desarrollo se siguió el mismo proceso del primer tablero. Se creó un procedimiento almacenado para cada división, fuera sección, distrito, municipio, etc. El tablero, igual que en el anterior, debía mostrar cifras a través de una serie de niveles. Para cumplir con esta funcionalidad se creó una página para cada división la cual manda los parámetros del nivel anterior para cargar los datos correspondientes.

Se crearon un total de 5 paginas (General, municipio, distrito, sección y listado) con una funcionalidad muy básica. Casi todo el trabajo se realizó a nivel consulta dejando el código para darle formato y manejar la navegación. Un ejemplo de cómo

se maneja el formato para el tablero se puede apreciar a continuación (figuras 4.21 a 4.23):

```
Dim avance As Double = e.Row.FCHyperLink("lblAvance").Text
Dim meta As Double = e.Row.FCHyperLink("lblMetas").Text
Dim vivienda As Double = e.Row.FCHyperLink("lblViviendas").Text
Dim participacion10 As Integer = e.Row.DBCampo("Participacion10")
Dim participacion12 As Integer = e.Row.DBCampo("Participacion12")
Dim participacion14 As Integer = e.Row.DBCampo("Participacion14")

e.Row.FCHyperLink("lblMetas1").Text = Math.Floor(meta * 0.3).ToString("N0")
e.Row.FCHyperLink("lblMetas2").Text = Math.Floor(meta * 0.6).ToString("N0")
e.Row.FCHyperLink("lblMetas3").Text = Math.Floor(meta * 0.9).ToString("N0")
e.Row.FCHyperLink("lblMetas4").Text = meta.ToString("N0")

e.Row.FCHyperLink("lblAvance").Text = avance & "%"
e.Row.FCHyperLink("lblAvance1").Text = Math.Round((participacion10 / e.Row.FCHyperLink("lblMetas1").Text.ToEntero) * 100, 1) & "%"
e.Row.FCHyperLink("lblAvance2").Text = Math.Round((participacion12 / e.Row.FCHyperLink("lblMetas2").Text.ToEntero) * 100, 1) & "%"
e.Row.FCHyperLink("lblAvance3").Text = Math.Round((participacion14 / e.Row.FCHyperLink("lblMetas3").Text.ToEntero) * 100, 1) & "%"
e.Row.FCHyperLink("lblAvance4").Text = avance & "%"
```

Figura 4.21. Ajuste de Porcentajes y Manejo de Tiempos

```
Dim zona As Integer = e.Row.DBCampo("ZonaEster")
e.Row.FCHyperLink("lblEster").NavigateUrl = String.Format("/DIAX/TableroControlEsterrMunicipios.", zona)
e.Row.FCHyperLink("lblPart").NavigateUrl = String.Format("/DIAX/TableroControlEsterrListado.aspx", zona)

e.Row.Cells(0).BackColor = Drawing.Color.White
e.Row.Cells(1).BackColor = Drawing.Color.White
e.Row.Cells(2).BackColor = Drawing.Color.White
e.Row.Cells(3).BackColor = Drawing.Color.White
e.Row.Cells(4).BackColor = Drawing.Color.White
e.Row.Cells(5).BackColor = System.Drawing.Color.FromArgb(218, 238, 243)
e.Row.Cells(6).BackColor = System.Drawing.Color.FromArgb(218, 238, 243)
e.Row.Cells(7).BackColor = System.Drawing.Color.FromArgb(183, 222, 232)
e.Row.Cells(8).BackColor = System.Drawing.Color.FromArgb(183, 222, 232)
e.Row.Cells(9).BackColor = System.Drawing.Color.FromArgb(146, 205, 220)
e.Row.Cells(10).BackColor = System.Drawing.Color.FromArgb(146, 205, 220)
e.Row.Cells(11).BackColor = System.Drawing.Color.FromArgb(22, 54, 92)
e.Row.Cells(12).BackColor = System.Drawing.Color.FromArgb(22, 54, 92)
e.Row.Cells(11).ForeColor = Drawing.Color.White
e.Row.Cells(12).ForeColor = Drawing.Color.White
```

Figura 4.22. Formato de Tablero

```
Cell_Header = New TableCell()
Cell_Header.Text = "Meta General"
Cell_Header.HorizontalAlign = HorizontalAlign.Center
Cell_Header.ColumnSpan = 3
Cell_Header.BackColor = System.Drawing.Color.FromArgb(0, 176, 240)
HeaderRow.Cells.Add(Cell_Header)

Cell_Header = New TableCell()
Cell_Header.Text = "Primer Corte 10:00"
Cell_Header.HorizontalAlign = HorizontalAlign.Center
Cell_Header.ColumnSpan = 2
Cell_Header.BackColor = System.Drawing.Color.FromArgb(183, 222, 232)
HeaderRow.Cells.Add(Cell_Header)

Cell_Header = New TableCell()
Cell_Header.Text = "Segundo Corte 12:00"
Cell_Header.HorizontalAlign = HorizontalAlign.Center
Cell_Header.ColumnSpan = 2
Cell_Header.BackColor = System.Drawing.Color.FromArgb(146, 205, 220)
HeaderRow.Cells.Add(Cell_Header)

Cell_Header = New TableCell()
Cell_Header.Text = "Tercer Corte 14:00"
Cell_Header.HorizontalAlign = HorizontalAlign.Center
Cell_Header.ColumnSpan = 2
Cell_Header.BackColor = System.Drawing.Color.FromArgb(49, 134, 155)
Cell_Header.ForeColor = Drawing.Color.White
HeaderRow.Cells.Add(Cell_Header)

Cell_Header = New TableCell()
Cell_Header.Text = "Cuarto Corte 16:00"
Cell_Header.HorizontalAlign = HorizontalAlign.Center
Cell_Header.ColumnSpan = 2
```

Figura 4.23. División de Secciones

Una vez terminado el tablero, el cual tenía la misma importancia que su plantilla, se pudo continuar con las plantillas de importancia no muy urgente para los departamentos más pequeños de la empresa. En este caso se nos pidió que iniciáramos con el desarrollo de la plantilla del departamento de 11-15.

Con la plantilla base terminada el desarrollo de la plantilla 11-15 fue relativamente rápido y sencillo. De hecho la plantilla es muy similar a la plantilla de ESTER, ambas plantillas se encargan de la distribución de puestos divididos por áreas geográficas más que por puestos como en el caso de la plantilla JE. Como se puede apreciar en la sección del tablero de control, ESTER se basa en el control del personal por municipios, distritos, secciones, etc. Sin embargo existen 15 zonas particulares que ESTER no cubre, las cuales se manejan a través de esta nueva plantilla.

Teniendo esto en mente es necesario aclarar que la razón por la que estas 15 zonas no se manejan en ESTER y también por la que no usamos la misma plantilla de ESTER para cubrir estas 15 zonas es porque su asignación tiende a ser más específica y requiere de un control mucho más estricto. Afortunadamente la estructura de la plantilla base nos permitió hacer un acomodo mucho más limpio y menos propenso a errores.

Nuestra primera tarea fue definir los datos de asignación, incluyendo pero no limitado a la creación de una tabla nueva para el control de personal de 11-15, los procedimientos almacenados para la creación, actualización y lectura de los registros, el añadir las nuevas propiedades y métodos a la clase `ClPlantilla` y por último el agregar las propiedades y controles nuevos a la plantilla 11-15. Con estas tareas realizadas resta simplemente agregar el código pertinente a cada sección en la plantilla base. Por ejemplo en la carga de datos se agregó lo siguiente (figura 4.24):

```

***** Cargamos los datos aqui *****

'Cargamos los datos del prospecto
Dim ObjPL As New ClsPlantilla

Try
    ObjPL.CargaProspectoOnce15(cveProspectito)
    CvePuesto = ObjPL.DtResultados.Rows(0).Item("CvePuesto").ToString.ToEntero
    FechaIngJE = ObjPL.DtResultados.Rows(0).Item("Fecha").ToString
    Superior = ObjPL.DtResultados.Rows(0).Item("CveProspectoPadreJE").ToString.ToEntero
    Zona = ObjPL.DtResultados.Rows(0).Item("Zona").ToString.ToEntero
    Municipio = ObjPL.DtResultados.Rows(0).Item("Municipio").ToString.ToEntero

Catch ex As Exception
    'OnError("Ocurrió un error al cargar los datos del prospecto", ex)
End Try

ObjPL.Dispose()

```

Figura 4.24. Carga de Datos

La figura 4.24 muestra claramente la mayoría de las tareas mencionadas anteriormente en ejecución. Podemos apreciar la definición de un objeto de la clase ClsPlantilla, un llamado al método `CargaProspectoOnce15()` para la obtención de los datos del prospecto, la asignación de los resultados de la consulta a las nuevas propiedades y como todo se insertó en la sección que definimos en la plantilla base “*Carga los datos aqui*”. Con la carga de los datos terminada solo resta agregar las demás partes como la carga de los ddl y la funcionalidad a ejecutar cuando la selección de las mismas cambie.

Con la plantilla base como cimiento el desarrollo de la plantilla 11-15 fue considerablemente más rápido, claro que también la experiencia obtenida después de realizar las otras plantillas ayudó bastante. Con estas ventajas se pudo terminar la plantilla de 11-15 en solo 2 días de trabajo incluyendo las pruebas, lo que tranquilizó a los superiores de la empresa y nos permitió continuar con el resto de las plantillas.

La siguiente plantilla con la que se trabajó fue la del departamento ENYRE. Esta plantilla es muy similar a la plantilla JE en el aspecto en que se centran principalmente en los puestos y en tareas específicas. El reto principal de esta plantilla fue el de agregar una funcionalidad nueva a los datos de asignación. Se nos pidió que se pudieran dar de alta varios municipios por prospecto. Esto representó un problema en primera instancia por la cantidad de información que se muestra debido a que son muchos municipios, y en segunda porque al cargar los datos del prospecto se deben cargar también sus selecciones.

Después de un análisis riguroso se optó por guardar la selección de varios municipios en una sola columna, esto debido a que toda la información de los prospectos debía manejarse en una sola tabla (políticas de la empresa). El guardado fue sencillo se almaceno una cadena con los valores de la selección separados por coma. Sin embargo esos datos se utilizarían después para algunos reportes por lo que se le dio prioridad a poder obtener estos datos correctamente.

Lo primero que se hizo para abordar esta problemática fue crear una consulta que nos permitiera obtener los valores de la columna de seleccionmultiple sin las comas. Esta consulta convierte cada valor con coma en tipo XML con la etiqueta <M> la que indica que es un valor de fila único. Por ejemplo teniendo el valor 1,2,3,4,5 la consulta regresa este valor: <M>1</M><M>2</M><M>3</M><M>4</M><M>5</M>. Una vez teniendo este resultado solo es cuestión de separar cada nodo en un una fila.

La consulta y su resultado se muestran a continuación para el ejemplo 1,2,3,4,5 (figura 4.25):

```

1 SELECT Split.a.value('.', 'VARCHAR(100)') AS SeleccionMultiple
2 FROM
3 (SELECT CAST ('<M>' + REPLACE(SeleccionMultiple, ',', '</M><M>') + '</M>' AS XML) AS SeleccionMultiple
4 FROM Clientes WHERE CveCliente = 3) AS A CROSS APPLY SeleccionMultiple.nodes ('/M') AS Split(a)

```

SeleccionMultiple
1
2
3
4
5

Figura 4.25. Consulta Split de Comas

Con la consulta terminada pudimos continuar con el desarrollo de la plantilla. En nuestro caso mostrar los valores es sencillo, sólo es cuestión de separar la cadena por comas e iterar a través del checkboxlist buscando los valores obtenidos y marcándolos como true. Esto se logra con el siguiente código:

```
Dim parts As String() = SeleccionMultiple.Split(New Char() {"",","})
Dim part As String

For Each part In parts
    Dim currentCheckBox As ListItem
    currentCheckBox = CheckBoxListCasilla.Items.FindByValue(part)
    If currentCheckBox IsNot Nothing Then
        currentCheckBox.Selected = True
    End If
Next
```

Al final la plantilla quedó de la siguiente manera (figura 4.26):

The screenshot shows a web browser window with the URL 'localhost:83/Esterr/PlantillaEnyRe.aspx'. The form contains the following sections:

- Personal Data:**
 - Apellido Paterno: LOPEZ (*)
 - Apellido Materno: AMEZQUITA (*)
 - Nombre(s): ALEXIS (*)
 - Fecha Nacimiento: 11/02/1992
 - Sexo: Femenino
 - IFE: LPAMAL92021126M800 (*)
 - Teléfono: [Empty]
 - Celular: [Empty]
 - Correo: [Empty]
- Address:**
 - Calle: C PASEO APASIBLE
 - Número: 7
 - No. Interior: [Empty]
 - Entre Calle: [Empty]
 - Colonia: COL LAS QUINTAS GALICIA
 - Código Postal: [Empty]
 - Seccion: 501
 - Municipio: Hermosillo
- Datos de asignación:**
 - Fecha de Ingreso: [Empty] (*)
 - Puesto: Responsable
 - Superior: LOPEZ NORIEGA ALEJ
 - Zona: NORTE
- Selección Múltiple:** A grid of 50 municipalities, each with an unchecked checkbox. The municipalities listed are: ACONCHI, AGUA PRIETA, ALAMOS, ALTAR, ARIVECHI, ARIZPE, ATIL, BACADEHUACHI, BACANORA, BACERAC, BACOACHI, BACUM, BANAMICHI, BAVIACORA, BAVISPE, BENITO JUAREZ, BENJAMIN HILL, CABORCA, CAJEME, CANANEA, CARBO, CUCURPE, CUMPAS, DIVISADEROS, EMPALME, ETCHOJOA, FRONTERAS, GRAL. PLUTARCO ELIAS CALLES, GRANADOS, GUAYMAS, HERMOSILLO, HUACHINERA, HUASABAS, HUATABAMPO, HUEPAC, IMURIS, LA COLORADA, MAGDALENA, MAZATAN, MOCTEZUMA, NACO, NACORI CHICO, NACOZARI DE GARCIA, NAVOJOA, NOGALES, ONAVAS, OPODEPE, OQUITOA, PITTIQUITO, PUERTO PEÑASCO, QUIRIEGO, RAYON, ROSARIO, SAHUARIPA, SAN FELIPE DE JESUS, SAN IGNACIO RIO MUERTO, SAN JAVIER, SAN LUIS RIO COLORADO, SAN MIGUEL DE HORCASITAS, SAN PEDRO DE LA CUEVA, SANTA ANA, SANTA CRUZ, SARIC, SOYOPA, SUAQUI GRANDE, TEPACHE, TRINCHERAS, TUBUTAMA, URES, VILLA HIDALGO, VILLA PESQUEIRA, YECORA.

At the bottom of the form are two buttons: 'Guardar' and 'Cancelar'.

Figura 4.26. Plantilla ENYRE

Con la evolución de las plantillas y el modelo de negocio de la empresa y la funcionalidad nueva que se puede obtener de las mismas, se amplió el panorama de lo que se podía obtener y las soluciones que se podían dar con las nuevas herramientas. Esto llevó a la inevitable decisión de no solo mejorar si no separar completamente el departamento JE en dos nuevos. Con este cambio se volvió necesario también el separar la plantilla de JE en dos para cubrir con las necesidades independientes de los dos nuevos departamentos.

Lo primero que se hizo fue redefinir la estructura de los departamentos, no solo cambiaron los puestos, sino también las tareas y la forma en la que estas se manejaban por personal. Con estos cambios hechos se me entregó la nueva estructura de ambos departamentos y con ella pude iniciar el análisis para separar adecuadamente la plantilla actual de JE.

El primer cambio que se hizo fue a nivel base de datos, se realizaron los cambios pertinentes en la tabla de puestos, se liberaron todos los prospectos con puestos y tareas obsoletas y se agregaron las columnas nuevas a la tabla de prospectos. Los reportes existentes también cambiaron para asegurarse de empatar con la nueva estructura del departamento.

Con estos cambios terminados nos pasamos nuevamente a la plantilla base y creamos una nueva copia. Inicé con la plantilla más sencilla, la J. Esta plantilla era la más similar a la que teníamos anteriormente, mantuvo la mayoría de los puestos y también sus tareas, además de la mayoría de sus superiores. Al igual que con las plantillas anteriores el proceso más extenso fue la modificación de la clase y todo lo relacionado a la base de datos. Al final todo recae en poner el código correspondiente en los apartados de la plantilla.

La nueva funcionalidad de la plantilla ENYRE que se desarrolló anteriormente abrió las puertas a formas más eficaces de distribuir y controlar la información. Como era de esperarse la funcionalidad de la selección de múltiples opciones se aplicó también a estas últimas dos plantillas y una vez más apareció un nuevo reto. La empresa me solicitó que ahora no hubiera solo una lista de checkbox sino dos, y que esta última dependiera directamente de las selecciones de la primera. Específicamente se quería que se pudieran seleccionar múltiples municipios y que al presionar el botón de cargar se mostraran todos los distritos correspondientes a los municipios seleccionados, permitiendo de igual manera la selección de múltiples de estos. Esta petición nos llevó también a cambiar la estructura actual de la tabla, ya que antes la selección múltiple se almacenaba en una columna general, al incluirse varias selecciones múltiples se optó por cambiar las columnas y su tipo para que en lugar de almacenar los valores enteros almacenaran cadenas.

Con esta problemática resuelta se pudo terminar el desarrollo de la plantilla J. La plantilla E no tuvo alguna problemática nueva, aunque si se implementó el uso de listas de checkbox, a estas alturas no era una funcionalidad nueva y mucho menos problemática. Esta plantilla se limitó a imitar la funcionalidad de camino que tenía la plantilla JE, donde dependiendo de las selecciones se fueran mostrando nuevos ddl y diferentes opciones hasta llegar al último nivel cuando fuera el caso. El trabajo a la par con los reportes que se mencionó anteriormente fue de vital importancia ya que así se mantenía un registro de los empleados que ya habían sido dados de alta, las tareas que se habían asignado y de esta manera era más fácil ver que faltaba por ser asignado y a quien.

La implementación satisfactoria de estas dos plantillas concluyó el ciclo de captura de mi tiempo en la empresa y dio inicio a la etapa de preparación para el evento final. Con un mes de anticipación se empezó a planear las herramientas que nos ayudarían a control todo el día del evento y los tableros que se ocuparían. Con esto en mente el tiempo que no se estuviera trabajando en los nuevos tableros fue utilizado para dar soporte a las plantillas y resolver dudas que surgieran con los capturistas.

De la misma manera que se solicitó un tablero de control para monitorear el personal y las actividades del departamento de ESTER, se nos solicitó un tablero para las plantillas de ENYRE y 11-15. Su funcionalidad es exactamente la misma que la de ESTER, se debe mostrar el avance y el porcentaje cubierto de la meta en distintas horas del día. Al cumplir la hora se detiene el avance de esa sección y se pasa a la siguiente. También se divide en niveles moviéndose entre municipios, secciones, etc.

El desarrollo de estos dos tableros se centró específicamente en la base de datos, ya que, como se mencionó anteriormente, el código se utiliza en su mayoría para darle formato al tablero y para controlar la navegación entre niveles. La consulta en la base de datos fue sencilla, se reunían todos los prospectos por nivel, dependiendo del nivel, y se agrupaban por dicho nivel haciendo un count para obtener la cifra exacta. Para poder ver la participación se utilizó la columna "*participo*" la cual almacenaba un valor de 0 o 1 para indicar si el prospecto había cumplido con su tarea. También fue importante almacenar la hora en la que ese valor se convirtió en uno.

Con esto en mente la consulta simplemente dividía los prospectos por agrupador y a su vez por sección donde solo se contaba la participación si esta cumplía con el parámetro de hora. Además se utilizaron una serie de unions para obtener los totales del nivel padre en los casos en que la consulta fuera un nivel hijo. Estos unions incrementaban su número entre más bajo fuera el nivel. Estos unions se convirtieron en una pieza clave del tablero ya que no volvían a consultar los valores del nivel padre sino más bien sirvieron de cascarón para que en código se pudiera reemplazar fácilmente la cantidad consultada.

De esta manera se redujo considerablemente el tiempo de consulta y la carga al servidor. Los valores de los niveles padres se pasaban como parámetro en el link del padre el cual recibía y colocaba el hijo. Y se consiguió el poder tener varios tableros abiertos al mismo tiempo (el cual era el objetivo ya que el tablero debía verse en los distintos centros de control).

En las imágenes siguientes se puede ver como se mandan las variables de nivel padre a hijo y como el hijo las acomoda en su lugar correspondiente, además de varios ejemplos de la estructura de los tableros (figuras 4.27 y 4.28):

```

If Request.QueryString("Zona") IsNot Nothing Then
    zona = Request.QueryString("Zona").ToString.ToEntero
End If

If Request.QueryString("metaEsterr") IsNot Nothing Then
    metaOnce15 = Request.QueryString("metaEsterr").ToString.ToEntero
End If

If Request.QueryString("viviendaEsterr") IsNot Nothing Then
    viviendaOnce15 = Request.QueryString("viviendaEsterr").ToString.ToEntero
End If

```

Figura 4.27. Recepción de Parámetros

```

.NavigateUrl = String.Format("/DIAX/TableroControlOnce15Secciones.aspx?Municipio=" & municipio
.NavigateUrl = String.Format("/DIAX/TableroControlOnce15Listado.aspx?Municipio=" & municipio &

```

Figura 4.28. Vínculos de Niveles

Por último se creó el nivel del listado la cual puede ser tanto una página independiente como el último nivel del tablero (figura 4.29). Esta página nos muestra el listado de personas en el nivel especificado ya sea a través de un vínculo o a través de la asignación de los filtros en la parte superior. Este listado funciona igual que el listado del primer tablero de control que se realizó.

Folio	Nombre	Municipio	Sección
Databound	Databound	Databound	Databound
Databound	Databound	Databound	Databound
Databound	Databound	Databound	Databound
Databound	Databound	Databound	Databound
Databound	Databound	Databound	Databound
Databound	Databound	Databound	Databound
Databound	Databound	Databound	Databound
Databound	Databound	Databound	Databound
Databound	Databound	Databound	Databound
Databound	Databound	Databound	Databound
Databound	Databound	Databound	Databound
Databound	Databound	Databound	Databound
Databound	Databound	Databound	Databound
Databound	Databound	Databound	Databound
Databound	Databound	Databound	Databound
Databound	Databound	Databound	Databound

Figura 4.29. Recepción de Parámetros

5. Análisis de la Experiencia Adquirida

En este apartado se presentan la valoración de los aprendizajes y lecciones aprendidas durante mi estancia profesional en la empresa PC Home, tanto en términos del proyecto en general, los objetivos que se plantearon para su realización, las actividades desarrolladas y la metodología que se utilizó.

5.1 Análisis General del Proyecto

Trabajar para la empresa PC Home fue diferente a lo que esperaba, ya que, no le daban importancia suficiente a las tecnologías de información. Con esto me refiero a que la empresa, a pesar de entender lo importante que estas tecnologías son para sus clientes, no aplicaba el mismo pensamiento en sus propios procesos. Esta mentalidad dificultó un poco el poder disfrutar de la experiencia de trabajar en una empresa desarrollada con procesos tecnológicos. En su lugar tuve que dedicarme a mejorar procesos arcaicos y lidiar con clientes que no entienden porque el cambio era necesario.

Por otro lado esto me permitió experimentar el lado amargo de mi carrera y el que quiéralo yo o no tendría que vivir eventualmente. Gracias a este proyecto ahora estoy más preparado, para lidiar con empresas en proceso de transición hacia una metodología más tecnológica y también puedo tratar con clientes reacios al cambio adecuadamente.

5.2 Análisis de los Objetivos de las Prácticas

Mi objetivo principal al entrar a PC Home era el de aprender cómo funciona una empresa desde adentro y como manejan adecuadamente toda su información. Mi interés principal era la base de datos, ya que nunca antes había visto una que tuviera tanta información. Debo admitir que al inicio sentía algo de confianza en mis habilidades de base de datos, sin embargo después de leer una de las consultas existentes me di cuenta que no en realidad no sabía nada.

A partir de ese momento establecí como objetivo personal el aprender tanto como pudiera de SQL y poder hacer cualquier consulta que me propusiera. Fuera de eso, mi trabajo en el framework fue de importancia menor para mí ya que no está hecho en un lenguaje que me guste mucho, ni que se ajuste a mis objetivos personales a futuro. Desafortunadamente en lo que a programación respecta, no puedo decir que

haya mejorado mis habilidades ya que en su mayoría todo se centró en la sintaxis más que en la lógica.

5.3 Análisis de las Actividades Realizadas

Ver mi estancia en la empresa PC Home desde el presente hacia el pasado me permite darme cuenta de cómo cada actividad fue esencial para la realización de la siguiente. Las primeras actividades que me tocó completar aunque importantes más que retar mis habilidades fueron una inducción hacia el sistema y el modelo de negocio de la empresa. Las primeras consultas me dieron las bases para ir mejorando mis conocimientos de SQL y para desarrollar más mi lógica de base datos.

El primer tablero de control me ayudo a familiarizarme con las tablas más importantes, las cuales estarían directamente relacionadas a las plantillas laborales. En lo que a plantillas respecta en general las actividades me resultaron un poco frustrantes al inicio. Trabajar en base al código de otra persona con una lógica y estilo diferentes al mío siempre es complicado, pero en este caso particular me tocó trabajar sobre el código de alguien que tuvo que ser removido del departamento de sistemas por sus malas prácticas de programación y falta de conocimientos. Claro que con la plantilla base terminada el desarrollo del resto de las plantillas no solo dejó de ser frustrante, se convirtió en algo que disfrute mucho.

Por último los tableros de control para la visualización de las actividades del personal, me dieron una sensación de satisfacción y realización personal cuando los vi por primera vez en acción. Ver mi desarrollo en pantallas gigantes ayudando a los gerentes a tomar decisiones importantes me lleno de emoción y me hizo disfrutar más mi trabajo.

5.4 Análisis de la Metodología Utilizada

No todas las herramientas utilizadas en mis prácticas profesionales fueron de lo mejor. Me toco desarrollar todo en un lenguaje al que ya no se le está dando soporte debido a su simplicidad y el cual ya fue reemplazado por un lenguaje mejor de alto nivel. Aunque Visual Basic resalta por lo fácil que es aprenderlo y aplicarlo a la larga trae más problemas de los que soluciona. Todo esto hablando personalmente, estoy seguro que muchas personas no compartirían mi opinión. Lo que más me molesto de tener que trabajar con este lenguaje fue el no sentir que estuviera avanzando,

dicho de otra manera, no sentí que estuviera aprendiendo algo que me abriera puertas más adelante.

El servidor de base de datos si fue un cambio agradable. A lo largo de mis estudios en la carrera el único lenguaje de base de datos que aprendí fue MySQL, aprende SQL Server me dio la motivación necesaria para realizar mis prácticas profesionales de la mejor manera. También el IDE que se utilizó para desarrollar es uno de mis favoritos y tuve la oportunidad de aprender algunos más de sus secretos. Por último el entorno de base datos (Management Studio), el cual fue nuevo para mí, también me gustó mucho y me agrado el haber aprendido a usarlo. Siempre que este aprendiendo algo nuevo puedo considerar lo realizado una victoria.

6. Conclusiones y Recomendaciones

En conclusión puedo decir que este proyecto me dejó muchas experiencias tanto buenas como malas. Lo primero que puedo decir es que el primer impacto que recibí fue un poco fuerte. Dicho de otra manera, me considero una de las pocas personas de mi carrera que pasó la mayoría de su tiempo de estudio aprendiendo tanto como le fuera posible y aprovechando cada oportunidad. Con esto en mente, el entrar a una empresa de verdad y ver cómo funcionan y manejan la información me dejó muy en claro que aún me faltaba mucho que aprender.

Sin embargo la experiencia más dulce que me quedó en todo este proceso es lo fácil que fue para mí adaptarme a mis responsabilidades y a lo que se me exigía como Ingeniero en Sistemas. Mi carrera no me dejó listo para cubrir cualquier rol en cualquier empresa de inmediato, pero si me dio las herramientas necesarias para adaptarme rápidamente a cualquier rol y a poder decir “sí lo desarrollo” sabiendo en el fondo que no tenía idea si lo que estaba diciendo era verdad.

Con esto no quiero decir que les estuviera mintiendo a mis superiores, si no que a pesar de no saber cómo hacerlo tenía la confianza suficiente para poder decir con certeza que lo podría hacer. Esta forma de pensar me permitió aprender mucho a lo largo de mi estadía y me enseñó un poco más lo que significa ser un Ingeniero en Sistemas.

Por otra parte las recomendaciones que le puedo dar a cualquier futuro egresado de la carrera es que se atrevan a hacer las cosas. Creo que pocos errores cuentan tanto como los que nos hacen mirar hacia atrás y preguntarnos qué hubiera pasado si lo hubiera hecho. Y esto no aplica solamente a puestos intimidantes en empresas importantes o en oportunidades de desarrollo personal, esta recomendación aplica para cada aspecto de la vida. No dejen que el miedo a fracasar o a sentirse tontos los detenga de hacer algo que les puede traer muchas cosas buenas, atrévanse.

Por último quiero recomendarles también que no pierdan a sus amistades ni corten los contactos que tienen con sus compañeros de la carrera. Incluso si no son sus amigos cercanos esos contactos les servirán mucho en un futuro. Una de las cosas que mejor sabor de boca me dejó la carrera es lo unidos que son los ingenieros entre sí. He tenido la oportunidad de recibir ofertas de trabajo en empresas o en proyectos de muchos de mis compañeros. Incluso de aquellos con los que no entable una conversación larga. Todo en la vida es más fácil en equipo y todo es más agradable en equipo también. No se cierren al mundo y a las personas a su alrededor, algún día les podría abrir una puerta importante.

7. Referencias Bibliográficas y Virtuales

[1]

http://www.tutorialspoint.com/software_testing_dictionary/capability_maturity_model.htm

[2] https://en.wikipedia.org/wiki/Capability_Maturity_Model_Integration

[3] https://en.wikipedia.org/wiki/Web_development

[4] https://en.wikipedia.org/wiki/Visual_Basic_.NET

[5] <https://en.wikipedia.org/wiki/ASP.NET>

[6] <http://www.w3schools.com/css/default.asp>

[7] <https://en.wikipedia.org/wiki/SQL>

[8] https://en.wikipedia.org/wiki/Microsoft_Visual_Studio

[9] https://en.wikipedia.org/wiki/SQL_Server_Management_Studio