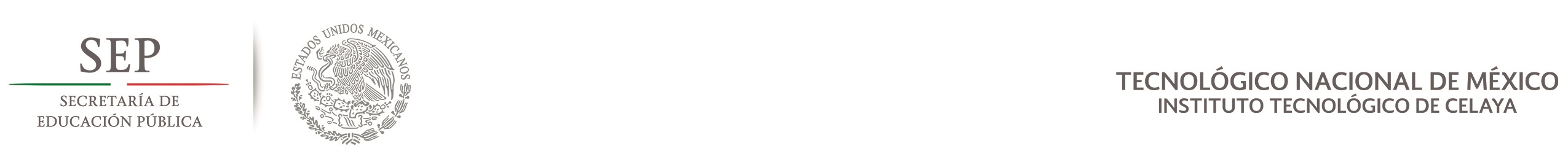
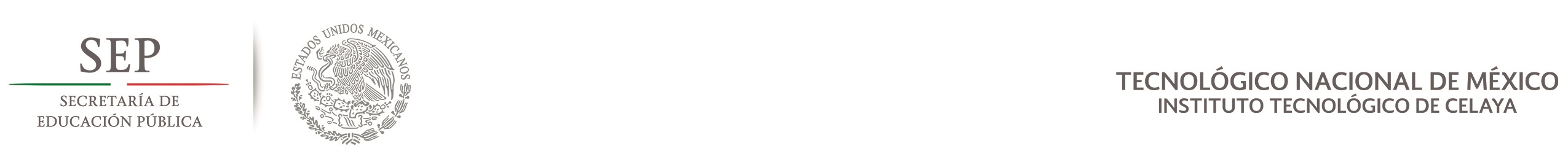
****

Departamento de Sistemas y Computación

**INFORME TÉCNICO DE RESIDENCIAS PROFESIONALES**

PORTAL DE PROVEEDORES Y COMPROBACION DE GASTOS CON VALIDACIÓN DE CFDI

**PINTURA ESTAMPADO Y MONTAJE SAPI DE CV**

11 DE JULIO DEL 2016 AL 15 DE ENERO DEL 2017

**Presenta:**

GUILLERMO ORTIZ REBOLLEDO

NO. DE CONTROL: 12030811

IGENIERÍA EN SISTEMAS COMPUTACIONALES

**Asesor Interno:**

ING. RUBÉN TORRES FRÍAS

**Asesor Externo:**

YANET RUIZ BARCENAS

**CELAYA, GUANAJUATO A 10 DE AGOSTO DEL 2016**

Índice

[1 Introducción 1](#_Toc471108167)

[2 Planteamiento del problema 2](#_Toc471108168)

[2.1 Establecimiento de prioridades 2](#_Toc471108169)

[3 Justificación 2](#_Toc471108170)

[4 Objetivos 3](#_Toc471108171)

[4.1 Objetivo general 3](#_Toc471108172)

[4.2 Objetivos específicos 3](#_Toc471108173)

[5 Método y desarrollo 4](#_Toc471108174)

[5.1 Procedimiento aplicado 4](#_Toc471108175)

[5.2 Programa de trabajo 5](#_Toc471108176)

[5.3 Descripción de las actividades 6](#_Toc471108177)

[6 Resultados obtenidos 6](#_Toc471108178)

[7 Conclusiones y recomendaciones 7](#_Toc471108179)

[7.1 Conclusiones 7](#_Toc471108180)

[7.2 Recomendaciones 7](#_Toc471108181)

[8 Referencias bibliográficas y virtuales 7](#_Toc471108182)

[ANEXO A 8](#_Toc471108183)

[1 Marco Teórico 8](#_Toc471108184)

[1.1 Tema 1 8](#_Toc471108185)

[1.2 tema 2 8](#_Toc471108186)

[1.3 tema 3 8](#_Toc471108187)

# Introducción

En los años recientes, la facturación electrónica ha cobrado importancia debido a su simplicidad y a las ventajas que tiene sobre las facturas en papel, además de que una factura electrónica cuenta con la misma información que una factura normal en papel.

En la empresa PEMSA es un requisito realizar una comprobación de gastos cada que algún empleado requiere hacer un viaje relacionado con el trabajo, ya sea visita a otra planta del corporativo, diplomados o cursos, visitas a proveedores, entre otros. Para realizar dicha comprobación de gastos es necesario ingresar en el sistema para registrar los datos del viaje, los gastos realizados y sus respectivos detalles, así como los comprobantes expedidos por los contribuyentes, es decir, las facturas electrónicas o también llamadas CFDI (Comprobante Fiscal Digital).

Es aquí donde surge la importancia de llevar a cabo una validación de tales comprobantes. El SAT (Servicio de Administración Tributaria) establece todos los campos necesarios que los comprobantes deben incluir y los criterios para establecer si este es válido o inválido.

Por esta misma razón, en PEMSA y las demás plantas de CIE Automotive es necesario que las facturas subidas al sistema de comprobación sean válidas ante el SAT.

En este reporte se describirá a detalle el proceso de realización de un módulo nuevo para el sistema de comprobación de gastos el cual se encarga de validar todas las facturas subidas.

# Planteamiento del problema

Se necesita implementar un módulo en el sistema de comprobación de gastos en el cual se puedan registrar los detalles de los gastos realizados y subir las facturas que les corresponden.

Los comprobantes o facturas subidas vienen tanto en formato XML como en PDF. El sistema se encargará de validar el archivo XML el cual contiene en su estructura toda la información que el SAT establece como necesaria.

Éste módulo deberá notificar al usuario en caso de que el comprobante subido sea inválido después de pasar por el proceso de validación, para que el usuario pueda subir otro comprobante o ponerse en contacto con la persona que expide la factura. Esto con el fin de tener en el sistema solamente las facturas o comprobantes válidos.

## Establecimiento de prioridades

Las prioridades que se tienen para llevar a cabo este proyecto se enlistan a continuación. Se tiene en cuenta desde que inician el proyecto hasta el día que terminan las residencias en la empresa.

1. Ponerse en contacto con los empleados del área de compras y de contabilidad y finanzas de todas las plantas del corporativo en México para reunir todos los requerimientos del proyecto.
2. Realizar investigación acerca de los requisitos que establece el SAT para llevar a cabo las validaciones de un comprobante.
3. Establecer una metodología, así como el lenguaje y entorno de desarrollo que la empresa establece para realizar el proyecto.
4. Llevar a cabo del desarrollo del proyecto.
5. Programar una junta con los empleados de los departamentos involucrados para informar sobre el proyecto y dar indicaciones de cómo usar el sistema.
6. Publicar la nueva versión del sistema para que las personas de compras y contabilidad puedan hacer las pruebas necesarias.
7. Atender cualquier caso de errores o bugs que se puedan presentar.

# Justificación

Como se menciona en el marco teórico anexo a este documento, una factura es un documento que avala la compra-venta de un bien o el préstamo de un servicio y además clave en la declaración de impuestos ya que a través de ella el SAT puede reembolsar los impuestos correspondientes a los contribuyentes.

El uso de comprobantes electrónicos o CFDIs trae varios beneficios en listados a continuación:

* La reducción del costo de los gastos de facturación. Se eliminan los gastos de impresión, mensajería y almacenamiento y esto ha implicado una reducción de costos importante para las empresas y los usuarios.
* La seguridad y rapidez en la emisión de comprobantes. Las facturas electrónicas se deben de realizar por medio de un PAC autorizado. Esto significa que este organismo cuenta con todos los requisitos impuestos por el SAT en materia de seguridad para timbrar documentos fiscales. Esto a su vez optimiza y facilita la carga de comprobantes en el sistema de la empresa.
* Existen menos errores y contratiempos a la hora de realizar una comprobación de gastos en el sistema de la empresa cuando un usuario sale de viaje y/o realiza una compra ya que se valida el comprobante desde antes de ser subido al repositorio de la empresa.
* Mayor control de los documentos y los registros de gastos en el sistema.

A partir del año 2014, conforme al artículo 29 del Código Fiscal de la Federación, todos los contribuyentes que tienen ganancias superiores a cuatro millones de pesos anuales están obligados a emitir Comprobantes Fiscales Digitales a través de internet.

Por esta misma razón es de suma importancia para PEMSA y todas las demás plantas del corporativo en México poder registrar y sobre todo validar estas facturas electrónicas en el sistema local de comprobación de gastos.

# Objetivos

## Objetivo general

Desarrollar un módulo para el sistema de comprobación de gastos que se encargue de capturar todos los datos requeridos para que el usuario pueda registrar los gastos que realizó en algún viaje relacionado con el trabajo. Además, validar los comprobantes fiscales cargados por el mismo usuario y notificarle a este en caso de que este comprobante no cuente con la estructura y datos requeridos por el Servicio de Administración Tributaria. También es necesario implementar un módulo de consulta para que el usuario pueda ver todos los conceptos registrados en sus comprobaciones, así como los certificados de validez generados después de llevar a cabo la validación de los comprobantes fiscales digitales. Todo esto aplica para los diferentes casos que se pueden presentar: reembolsos, modificaciones y anticipos.

## Objetivos específicos

* Desarrollar una interfaz dentro del sistema de comprobación de gastos donde los usuarios puedan registrar los datos de la comprobación, así como la carga de los archivos PDF y XML (comprobante fiscal digital o factura electrónica).
* Desarrollar un módulo para validar los comprobantes cargados por los usuarios en base a los requerimientos del SAT.
* Desarrollar un módulo que permita a los usuarios consultar las facturas guardadas en el repositorio del sistema.
* Hacer las modificaciones necesarias en la base de datos dedicada al sistema de comprobación de gastos para guardar las validaciones hechas a los comprobantes.
* Desarrollar un programa o script que permita hacer la descarga de los archivos necesarios que publica el SAT en un servidor FTP para tener los requerimientos más recientes para las validaciones.
* Incorporar la validación de los comprobantes a los módulos de reembolso, modificación y anticipo dentro del sistema de comprobación de gastos.

# 5 Método y desarrollo

## 5.1 Procedimiento aplicado

Para el proceso de desarrollo se utilizó el método ágil de desarrollo ‘Scrum’. Como se explica en el marco teórico de este reporte, Scrum maneja tres roles principales: Scrum Master, Product Owner y el Team (equipo). La distribución de los roles para nuestro caso se ilustra a continuación:

* Scrum master: Yanet Ruiz Bárcenas (Jefa inmediata).
  + Elimina cualquier obstáculo que pueda existir que impidan al equipo lograr algún objetivo.
  + Se asegura de que se lleve a cabo el proceso de forma correcta y que se cumplan las reglas.
* Product Owner: Annel Morales Marrón (Departamento de Compras).
  + Se asegura de que el equipo trabaje de manera adecuada desde la perspectiva del negocio.
* Equipo de desarrollo: Guillermo Ortiz Rebolledo (Desarrollador).
  + En este caso en particular, el equipo solamente se conforma por una persona.
  + Se dedica a analizar, diseñar, desarrollar, probar y documentar durante el proceso de desarrollo del sistema que se tiene como objetivo.

Esta metodología de desarrollo ha tenido un auge durante los últimos años en el sector del desarrollo de software ya que trae muchos beneficios a los equipos que lo practican, en este caso, por ejemplo:

* Existe mayor flexibilidad a cambios con base en las necesidades del cliente, en este caso siendo el departamento de compras y el de finanzas. El marco de trabajo puede adaptarse a sus necesidades sin necesidad de hacer cambios drásticos o perder tiempo o progreso.
* La calidad del producto, en este caso los módulos que serán agregados al sistema, serán de alta calidad ya que se pueden agregar y pulir las características de estos en cada iteración.
* Existe más productividad ya que el equipo de antemano sabe sus responsabilidades y trabaja de manera autónoma.
* Se reducen riesgos ya que se desarrollan primero las funcionalidades de mayor valor y se conoce la velocidad con la que el equipo avanza en el proyecto.

Las reuniones en Scrum se realizan de manera diaria. En este caso la gran mayoría de reuniones se llevaron a cabo diariamente a excepción de los días donde Yanet Ruiz, el Scrum master, por motivos de la empresa tenía que salir de planta. Las reuniones no duran más de quince minutos en el escritorio del Scrum master y se tocan temas acerca del progreso del proyecto, qué se hizo el día anterior, que se hará ese día y si existen problemas u obstáculos que nos impidan llegar al objetivo.

Existen también las reuniones de planificación las cuales se realizan cada mes o cada quince días. En estas discutimos lo que se planea hacer y planear el tiempo que nos llevará hacer el trabajo. Después de cada una de estas reuniones, tanto las reuniones Scrum como las de planificación tienen una parte de revisión, donde se presenta el trabajo completado y la retrospectiva donde los miembros del equipo dan sus impresiones y opiniones sobre las reuniones que se han realizado. Esto se hace con el objetivo de tener una mejora continua.

## 5.2 Programa de trabajo

|  |  |
| --- | --- |
|  | Sistema de Báscula |
|  | Sistema de Comprobación de Gastos |
|  | No se labora. |
|  | Días festivos. |
|  | Actividades de la empresa. |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Mes** | **Lun** | **Mar** | **Mié** | **Jue** | **Vie** | **Sáb** | **Dom** |
| Jul  2016 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 11  Recorrido por la planta.  Entrega de papeles. | 12  Pláticas de inducción. | 13  Examen de seguridad.  Examen médico. | 14  Asignación de lugar de trabajo. | 15  Introducción al sistema de báscula. | 16 | 17 |
| 18  Realizar modificaciones al sistema de báscula. | 19  Obtener IPs de los usuarios para identificar planta.  Implementar autenticación por medio del ID de la planta.  Agregar campo ‘Estatus’ a la tabla Báscula. | 20  Crear una tabla en la base de datos llamada ‘Acceso Báscula’ para registrar el acceso de los usuarios en las diferentes plantas. | 21  Crear una tabla para Vigilantes ligada con Planta y Estatus.  Agregar campo de Estatus en las tablas de Producto, Contenedor y Transportista. | 22  Direccionar al usuario a la báscula disponible de acuerdo a los permisos que tenga.  Mostrar fecha y tiempo correctos en tiempo real. | 23 | 24 |
| 25  Cada vez que el usuario del sistema (vigilante) guarde en registro de pesaje, requerir contraseña y autenticar. | 26  Agregar función de impresión al sistema.  Cada que el vigilante registre un pesaje, guardar en la tabla ‘Pesaje’ e imprimir informe. | 27  Guardar el ID del vigilante que realiza el pesaje y su usuario de red en la tabla ‘DetallePesaje’. | 28  En la sección de segunda pesada, mostrar información de la primera pesada cuando se selecciona el folio correspondiente. | 29  Al momento de la impresión mostrar una lista de impresoras disponibles en el equipo donde se ejecuta el sistema.  Imprimir informe de primera y segunda pesada cuando se guardan los datos de la segunda. | 30 | 31 |
| Ago  2016 | 1  Hacer pruebas con web services del SAT sobre validez de facturas de prueba. | 2  Hacer un programa de prueba para leer archivos XML y obtener datos para hacer llamadas al web service. | 3  Con base en el anexo 20 de la Resolución Miscelánea Fiscal, añadir validaciones al programa de prueba para facturas. | 4  Reunión con personal de Contabilidad y de Compras para propuesta de proyecto de Comprobación de Gastos. | 5  Establecimiento de requerimientos para sistema de comprobación de gastos.  Reunión con jefe inmediato para introducción al sistema anterior. | 6 | 7 |
| 8  Iniciar sistema web para el proyecto de Báscula.  Establecimiento de requerimientos y especificaciones. | 9  Configuración de proyecto en Visual Studio.  Configuración de archivo Web.config con detalles de repositorio y base de datos. | 10  Añadir sección de catálogos (productos, transportistas, contenedores, etc.) | 11  Añadir módulo para agregar elementos al catálogo (Productos, transportistas, contenedores, etc.). | 12  Añadir módulo de búsqueda en los catálogos. | 13 | 14 |
| 15  Realizar una investigación sobre los requisitos del SAT para validar una factura (XML). | 16  Hacer diagramas de flujo para cada una de las validaciones identificadas en la investigación realizada. | 17  Revisar validaciones propuestas.  Comparar validaciones propuestas con las validaciones que realizan otros proveedores. | 18  Codificar validación del esquema utilizado en la factura. (Estructura y contenido de los nodos y atributos en el XML). | 19  Pruebas y validaciones del esquema con base en los archivos proporcionados por el SAT en su portal de internet. | 20 | 21 |
| 22  Codificar validación para los namespaces utilizados en las facturas. | 23  Pruebas y validaciones de los namespaces con base en los archivos y referencias que establece el SAT. | 24  Codificar validación para la igualdad de sellos en los archivos XML de las facturas cargadas. | 25  Pruebas y validaciones de igualdad de sellos con XML’s de facturas de prueba. | 26  Validaciones generales de los algoritmos.  Junta para verificar proceso del sistema. | 27 | 28 |
| 29  Codificar validación de **contenido** de las facturas. | 30  Pruebas y validaciones del contenido de las facturas cargadas de acuerdo a los nodos los atributos que establece el SAT. | 31  Codificar validación del **timbre fiscal digital** utilizado para sellar las facturas. | 1  Pruebas y validaciones del timbre fiscal digital en facturas de prueba teniendo con certeza cuáles son válidas e inválidas. | 2  Validaciones generales de los algoritmos.  Junta para verificar proceso del sistema. | 3 | 4 |
| Sep  2016 | 5  Codificar validación de la estructura del **timbre fiscal digital** por medio de archivos que proporciona el SAT. | 6  Pruebas y validaciones de la estructura y validez del timbre fiscal digital con facturas de prueba. | 7  Codificar validación de **vigencia de la factura o comprobante** (CFDI) con base en el web service del SAT. | 8  Pruebas y validaciones de vigencia con facturas de prueba.  Verificación de fechas de comprobantes. | 9  Validaciones generales de los algoritmos.  Junta para verificar proceso del sistema. | 10 | 11 |
| 12  Realizar investigación acerca de los sellos utilizados para firmar los certificados de las facturas. | 13  Descargar archivo de texto proporcionado por el SAT que contiene todos los sellos, su status y su vigencia. | 14  Prueba y validaciones de la validez de los certificados y sellos utilizados por el PAC para firmar una factura. | 15 | 16 | 17 | 18 |
| 19  Añadir todas las validaciones necesarias en una clase en Visual Basic para poder implementarla en el sistema. | 20  Crear las tablas correspondientes en la base de datos para almacenar la información de las validaciones hechas en el sistema. | 21  Modificar las tablas que ya existían anteriormente en la base de datos en los casos donde sea necesario. | 22  Crear los procedimientos almacenados correspondientes para almacenar la información temporal de las validaciones en la base de datos. | 23  Añadir código en la clase de las validaciones de los XML para ejecutar los procedimientos almacenados que se ocupan para insertar información en la base de datos. | 24 | 25 |
| 26  Crear un programa de consola en Visual Basic donde se descargue el archivo de texto desde el servidor FTP del SAT. | 27  Crear una tabla en la base de datos para almacenar todas las líneas de texto en el archivo. | 28  Agregar código para inserción a la tabla en la BD.  Realizar pruebas del programa de consola para descargas. | 29  Implementar validaciones a los comprobantes al cargar el archivo XML en la sección de reembolso del sistema. | 30  Pruebas de validación de comprobantes en el módulo de reembolso del sistema. | 1 | 2 |
| Oct  2016 | 3  Codificar función para imprimir un **Certificado de Validez** que se genera una vez que el comprobante es invalido y detalla los errores que contiene. | 4  Investigación de librerías de .NET y Visual Basic para generar archivos PDF con base en código HTML.  Implementación de librerías. | 5  Creación de módulo de **consulta** de comprobaciones.  El usuario puede consultar las comprobaciones de gastos realizadas. | 6  Creación de módulo de **consulta global** para usuarios con perfil de ‘administrador’.  El usuario puede consultar comprobaciones de usuarios de todas las plantas. | 7  Integración del módulo de impresión de certificados a los módulos de consultas de comprobaciones. | 8 | 9 |
| 10  Agregar validación en el módulo de consultas para verificar si los comprobantes dentro de la comprobación realizada ya han sido validados. | 11  Correcciones en algunas tablas en las bases de datos.  Reestructuración de las tablas temporales de validaciones. | 12  Agregar pantalla modal de carga cuando el usuario carga un archivo XML y necesita ser validado. | 13  Modificar la clase de ValidacionXML para que lea los registros en la base de datos en lugar de leer el archivo XML en cada validación. | 14  Modificar el proceso de guardado de archivos en el módulo de validación de comprobantes. | 15 | 16 |
| 17  Guardar temporalmente el archivo XML en una carpeta local del proyecto para su validación, después copiar los archivos validados al repositorio y limpiar carpeta local. | 18  Modificar programa de consola para descarga del archivo de texto con los sellos del SAT para que almacene una nueva copia del archivo cada vez que se ejecute y organizar los archivos anteriores por fecha. | 19  Agregar los campos de UUID en las tablas correspondientes en los diferentes módulos de comprobación de gastos (reembolso, anticipo y modificación). | 20  Agregar código para borrar registro de las tablas temporales de validación involucradas cuando el usuario da clic en el checkbox ya sea en modificaciones, reembolsos o anticipos. | 21  Implementar módulo de validación de comprobantes en las secciones de reembolso y anticipo del sistema. | 22 | 23 |
| 24  Corregir detalle con el módulo de modificación donde no se cargan los datos de la comprobación cuando el usuario da clic en el GridView. | 25  En el módulo de modificación, hacer una verificación para no volver a validar comprobantes que se hallan subido anteriormente. | 26  En los tres módulos (anticipo, reembolso y modificación), leer los datos del comprobante que se carga y definir si ya ha sido usado en otro concepto. | 27  Agregar opción al sistema web de báscula para ver los detalles del elemento seleccionado en el módulo de catálogos. | 28  Agregar opción para borrar y editar elementos de los catálogos desde la misma tabla o GridView donde se enlistan. | 29 | 30 |
| 31  Agregar módulo en el sistema web de báscula para crear una hoja de cálculo en Excel a partir de la tabla con todos los pesajes registrados. | 1 | 2 | 3  Agregar GridViews para Vigilantes, Productos y transportistas con base en las tablas de la base de datos. | 4  Agregar opción para edición de filas en los GridView para modificar los registros de la base de datos. | 5 | 6 |
| Nov  2016 | 7  Agregar procedimiento almacenado en la base de datos para la carga de tablas temporales. | 8  Agregar código en el sistema para ejecutar el procedimiento para cargar las tablas temporales y además limpiarlas. | 9  Llenar las tablas temporales con información de las tablas productivas cuando se carga una comprobación en el módulo de Modificación. | 10  Agregar procedimiento almacenado para borrar los contenidos de las tablas temporales para los casos de excepción. | 11  Corregir módulo de reembolso y de modificación.  Solo cuando es un archivo XML se guardan datos específicos como UUID, RFC, etc. | 12 | 13 |
| 14  Pruebas para sistema de báscula de escritorio. | 15  Modificar módulo de informe de pesaje para que muestre los detalles de la primera y segunda pesada. | 16  Crear archivos de CrystalReports dentro de Visual Studio para imprimir informes. | 17  Pruebas y correcciones de impresiones de informes. | 18  Verificar y agregar compatibilidad con Internet Explorer. | 19 | 20 |
| 21  Eliminar y cambiar menú de catálogos y agregar sesiones al sistema web. | 22  Agregar filtros de búsqueda en cada sección de catálogos. | 23  Agregar ‘Scroll’ a todas las tablas o GridViews de catálogos. | 24  Agregar opción para descargar ZIP, XML o PDF en la consulta de comprobaciones. | 25  Agregar código para la verificación de comprobantes que ya se hayan validado anteriormente y agregarlos a la base de datos. | 26 | 27 |
| 28  Asistencia a diplomado. | 29  Asistencia a diplomado. | 30  Agregar opciones de tablas temporales nuevas a procedimientos almacenados. | 1  Agregar valores parametrizables en el código al archivo de configuración del sistema así como la base de datos. | 2  Agregar función para insertar en la tabla principal CertificadoValidez cuando el usuario guarda una comprobación de algún concepto en alguno de los 3 módulos. | 3 | 4 |
| Dic  2016 | 5  Crear procedimiento almacenado para duplicar registros y archivos cuando el usuario indica un ID de archivo. | 6  Crear procedimiento almacenado para duplicar los registros de las tablas de validación cuando el usuario indica un ID de archivo. | 7  Implementar en el código del sistema instrucciones para ejecutar ambos procedimientos. | 8  Modificar código en los 3 módulos para dependiendo si se cargan archivos PDF o XML ejecutar los insert en la base de datos con diferentes parámetros. | 9  Modificar procedimiento almacenado ‘ComprobantesArchivos’ para insertar los UUID en las tablas cuando se duplica un registro de comprobante. | 10 | 11 |
| 12  Asistencia en evento organizado para los trabajadores. | 13  Pruebas del sistema. | 14  Pruebas del sistema. | 15  Pruebas del sistema. | 16 | 17 | 18 |
| 19  Junta y conferencia telefónica con los futuros usuarios finales del sistema. | 20  Mantenimiento y soporte de sistemas y proyectos hechos. | 21  Mantenimiento y soporte de sistemas y proyectos hechos. | 22  Agregar parámetros de insert, update y filtros a las GridViews en catálogos. | 23  Agregar funciones nuevas en una clase del sistema web para hacer consultas específicas a la base de datos. | 24 | 25 |
| 26  Agregar función de guardado en todas las tablas o GridViews de catálogos. | 27  Agregar función de filtrado y limpiado en todas las tablas o GridViews de catálogos. | 28  Agregar función para filtrar los elementos que ya hayan sido usados en algún pesaje antes de actualizar algún registro del catálogo. | 29 | 30 | 31 | 1 |
| Ene  2017 | 2  Mantenimiento y soporte de sistemas y proyectos hechos. | 3  Mantenimiento y soporte de sistemas y proyectos hechos. | 4  Mantenimiento y soporte de sistemas y proyectos hechos. | 5  Mantenimiento y soporte de sistemas y proyectos hechos. | 6  Mantenimiento y soporte de sistemas y proyectos hechos. | 7 | 8 |
| 9  Mantenimiento y soporte de sistemas y proyectos hechos. | 10  Mantenimiento y soporte de sistemas y proyectos hechos. | 11  Mantenimiento y soporte de sistemas y proyectos hechos. | 12  Mantenimiento y soporte de sistemas y proyectos hechos. | 13  Mantenimiento y soporte de sistemas y proyectos hechos. | 14 | 15 |

## 5.3 Descripción de las actividades

Durante el periodo de residencias profesionales, se realizaron distintas actividades desde creación de sistemas nuevos, mantenimiento de bases de datos y sistemas que ya existen, así como adición de nuevos módulos a los mismos.

Como requerimiento de la empresa por cuestiones de convenios y licencias, todos los sistemas y aplicaciones hechos fueron desarrollados con la herramienta **Visual Studio** de Microsoft teniendo **Visual Basic** y **C#** como los lenguajes de programación y **ASP.NET** como framework para proyectos web. Se hizo uso del sistema gestor de base de datos relacionales **Microsoft SQL Server** para almacenar toda la información de los desarrollos. Cabe mencionar que, para cuestiones de compatibilidad con desarrollos locales así como los hechos por proveedores externos, se utiliza Internet Explorer como navegador predeterminado en todos los equipos de la empresa.

A continuación, se enlista de forma general las actividades que se llevaron a cabo durante la estancia en la empresa:

**Sistema de Báscula**

**Descripción General**

* Desarrollo de un **sistema para escritorio** (PC/Windows) para administrar los pesajes de la báscula ubicada en la entrada a la planta por parte de los vigilantes.
* Desarrollo de un **portal web** para administrar los catálogos (productos, transportistas, vigilantes, contenedores, etc.). Estos dos sistemas están ligados, pero van dirigidos a diferentes usuarios.

**Descripción Detallada**

El sistema va dirigido al personal de vigilancia que se encuentra en la entrada de la planta. En la entrada a las instalaciones de la empresa, se encuentra una báscula instalada en el concreto donde los camiones y automóviles en general que transporten productos deben registrar su peso tanto de entrada como de salida. Esto con el fin de llevar un control estricto de los productos que entran y salen de la planta, sus respectivos pesajes, el transportista que se encarga de trasladar los productos, el contenedor utilizado en caso de que aplique y el vigilante que llevó a cabo dicho pesaje. Para facilitar todo este proceso, se hizo la petición al departamento de sistemas de desarrollar un sistema que se encargara de agilizar todo este proceso.

Este sistema se divide en dos, la versión de escritorio (PC / Windows) y la versión web. La primera teniendo como usuarios finales al personal de vigilancia que opera en la planta y la segunda teniendo como usuarios al personal de compras y contabilidad de la empresa.

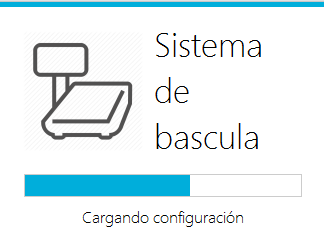
El primer paso a seguir fue identificar todos los requerimientos brindados por el personal de vigilancia de la planta. Con base en todos estos requerimientos, nos fue posible modelar la base de datos que se ocuparía para almacenar todos los registros necesarios.

El sistema de báscula en su versión de escritorio se divide en tres módulos principales: Primera pesada, segunda pesada y reportes. Como se mencionó anteriormente, todos los productos que sean trasladados desde o hacia las plantas del corporativo tienen que ser pesados a la entrada y salida de la planta con el fin de llevar un registro y control transparente tanto de los pesajes como de los productos y transportistas que llevan a cabo la operación.

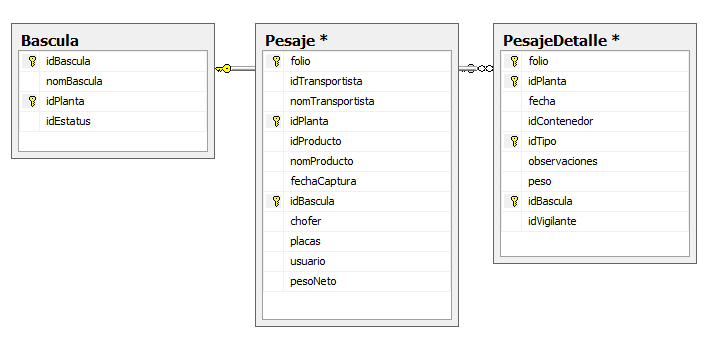
Como características generales, la ventana de la aplicación debe tener tres pestañas que indican las acciones a realizar, ya sea en alguna de las dos pesadas o la sección de reportes. En la esquina superior derecha de la ventana se debe mostrar el nombre de la planta en la que se está utilizando el sistema (son 10 plantas en las que se puede implementar este sistema), para poder conseguir esto, se tiene una tabla **Planta** en la base de datos con los datos generales como RFC y razón social, además se tiene una tabla **SegmentosIP** donde se almacenan los segmentos de IP que se utilizan en cada una de las plantas almacenadas en la base de datos. Para obtener la IP del equipo donde se ejecuta el sistema, se utilizan métodos nativos del **framework de .NET** como GetHostname y GetHostEntry para obtener la IP y compararla con los registros de la base de datos, permitiendo así determinar desde que planta está siendo usado el sistema.

Además de la identificación de la planta en la que se opera, también se debe llevar a cabo una autenticación del equipo donde se ejecuta el sistema. Cada equipo en la empresa (todas las plantas) tiene asignado un usuario el cual está registrado en el Directorio Activo. Por lo general cada empleado tiene asignado un usuario, pero en el caso de vigilancia, todo el personal de utiliza un solo equipo en la caseta de su respectiva planta. Al iniciar el sistema, la aplicación debe realizar una búsqueda del nombre del equipo en el directorio activo de la empresa, si es encontrada entonces se le garantiza el acceso, de lo contrario, se restringe el acceso y no se le dan permisos al equipo de operar el sistema.

Se utilizó la guía de diseño y framework para interfaces de Microsoft conocido como “**Metro**” uniformemente en todo el sistema de escritorio. La primera pantalla que se presenta en este desarrollo es una pantalla de carga en la cual se lleva a cabo la autenticación del equipo y la extracción de la IP para identificar la planta en la cual se ejecuta el sistema, tal como se mencionó anteriormente.



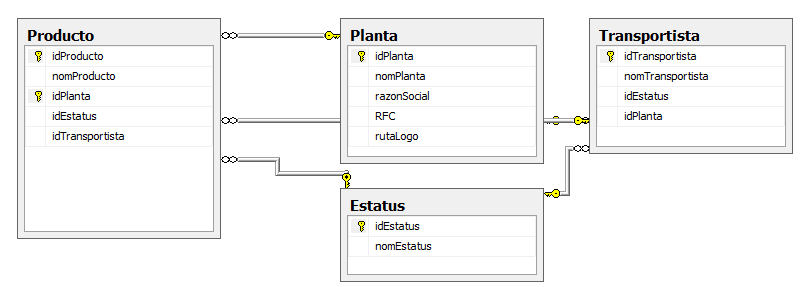
Se creó una tabla en la base de datos para Producto, Transportista, Estatus (Activo o Cancelado) y Planta con los campos de Nombre y Razón Social de las 10 plantas del corporativo en México. Se creó también una tabla con los registros de las básculas que se tienen en las diferentes plantas del corporativo las cuales también tienen un Estatus para saber si están activas o no. Sumado a todas estas tablas básicas, se modeló una tabla fundamental para el sistema donde se almacenarán todos los pesajes y los detalles de cada uno de los mismos. Estas tablas de la base de datos fueron estructuradas de la siguiente manera:



Tablas de Pesaje, PesajeDetalle y Báscula. Se tienen más relaciones con estas tablas, las cuales se muestran más adelante.

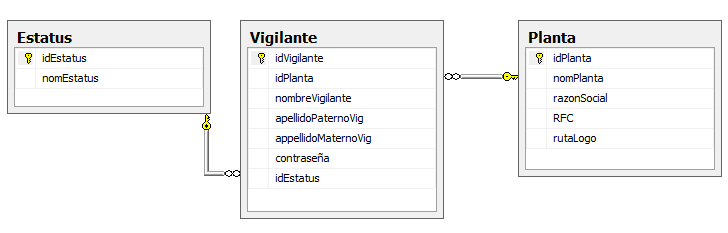
En el sistema se consideró que un producto puede tener varios transportistas, además de su estatus y la planta de donde proviene. Es por eso que, al registrar el primer pesaje, se habilita la lista para seleccionar un producto hasta después de haber seleccionado un transportista.

Todos los elementos involucrados en el pesaje tienen un Estatus con el fin de hacer un filtrado en un futuro para identificar qué productos, transportistas, etc. están disponibles. Las tablas mencionadas se modelaron de la siguiente manera:



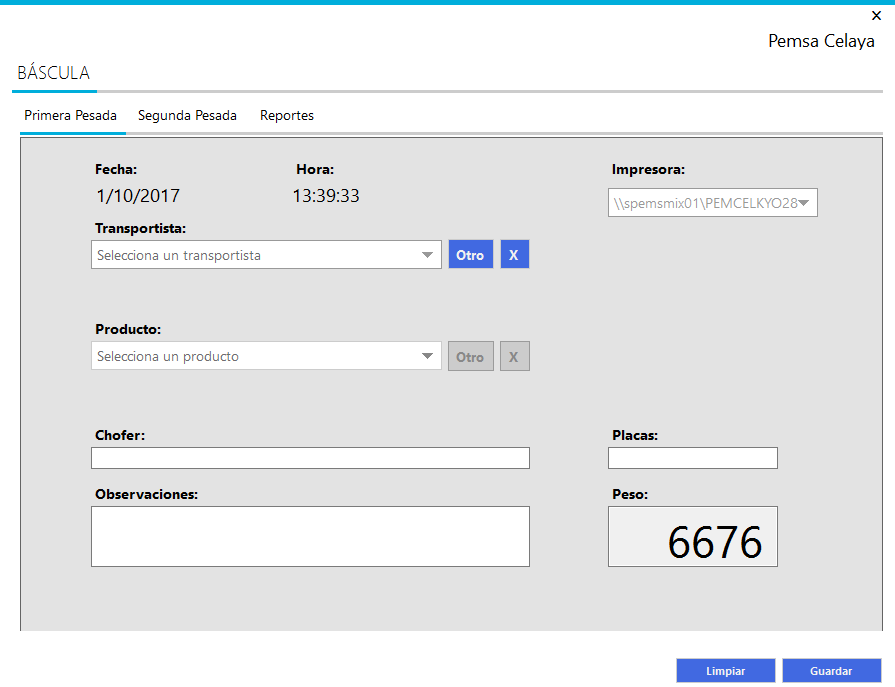
Al momento de realizar un pesaje, si el producto o transportista no están registrados en la base de datos, se debe tener la opción de agregarlos. Es por esto que en la aplicación se agregó una opción de ‘Otros’ junto a las listas de transportistas y productos, así el usuario puede registrar estos nuevos elementos al mismo tiempo que el pesaje.

En la base de datos, también fue creada una tabla de Vigilantes. En esta tabla se guardan los datos de nombre y apellidos, contraseña, estatus y la planta donde labora el vigilante. Este registro se lleva a cabo con anterioridad ya sea por el administrador o los usuarios del sistema web (se explicará a detalle más adelante).



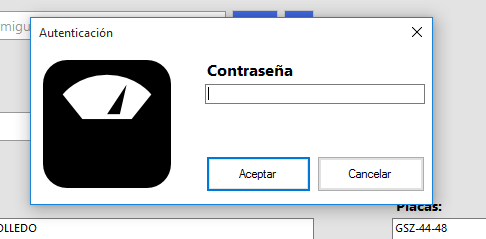
Para el primer módulo (**Primera Pesada**) se capturan los siguientes datos: el nombre del producto, la empresa transportista y nombre del chofer que se encarga del traslado, la hora y fecha en la que se realiza el pesaje (se capturan automáticamente), las placas del vehículo utilizado, un campo para observaciones, una lista desplegable con las impresoras disponibles para imprimir los reportes y, por supuesto, una caja de texto que muestra el peso actual que registra la báscula.

Para mostrar el peso que registra la báscula en tiempo real se utilizaron las clases nativas de Visual Basic y .NET que permiten leer los datos que manda el puerto serial de la computadora. Esto debido a que la báscula se conecta por este medio a la computadora. Se agregó un elemento Timer en la aplicación el cual sirve como base para que tanto el peso que registra la báscula y el tiempo se actualicen constantemente cada segundo.

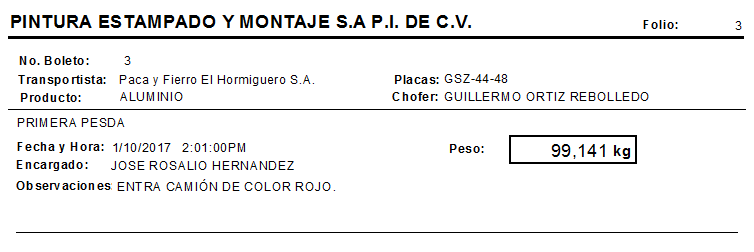


Ventana principal con las tres pestañas de primera pesada, segunda pesada y reportes.

El botón de ‘Limpiar’ ubicado en la parte inferior de la ventana sirve para dar un reset a todos los campos de texto en la ventana en caso de errores de escritura al llenarlos o simplemente para limpiar los campos. El botón de ‘Guardar’ sirve para registrar el pesaje una vez que se llenan todos los campos. Para esto el vigilante tiene que autenticarse con la contraseña proporcionada para ‘autorizar’ el registro del pesaje, de lo contrario se negará el permiso.

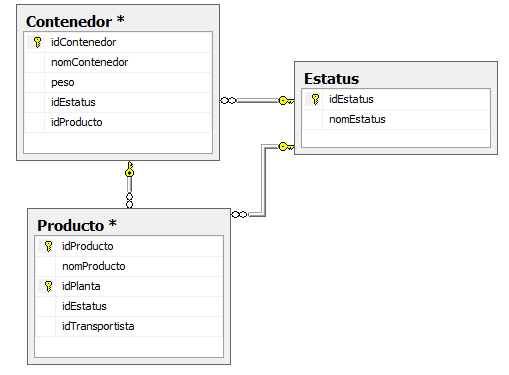


Una vez registrado el pesaje en la base de datos por medio de procedimientos almacenados, se imprime el reporte correspondiente en la impresora seleccionada (si no se selecciona ninguna, se imprime en la impresora por default). Los reportes son generados por las librerías CrystalReports que vienen integradas con Visual Studio. Para generar el reporte, se hizo una selección de las tablas y campos de la base de datos y se le da formato a la página. A continuación, se muestra un ejemplo de la impresión del reporte de la primera pesada:

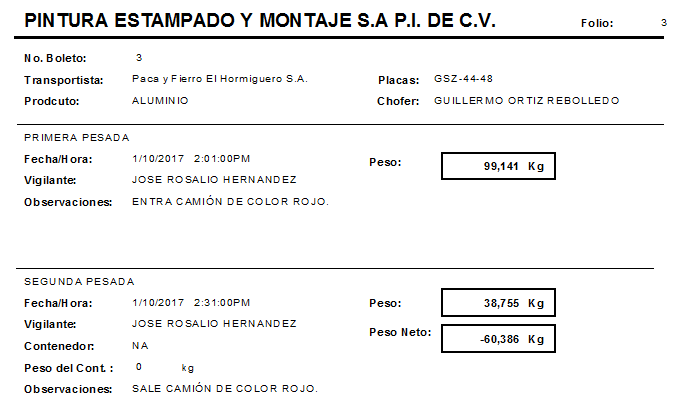


Para el módulo de **Segunda Pesada**, se tiene una lista desplegable con los folios de los pesajes que solo tienen registrado un pesaje. Se codificó un método para cargar los datos del pesaje con el folio seleccionado. Este se ejecuta automáticamente cuando el usuario selecciona un pesaje desde la lista mencionada.

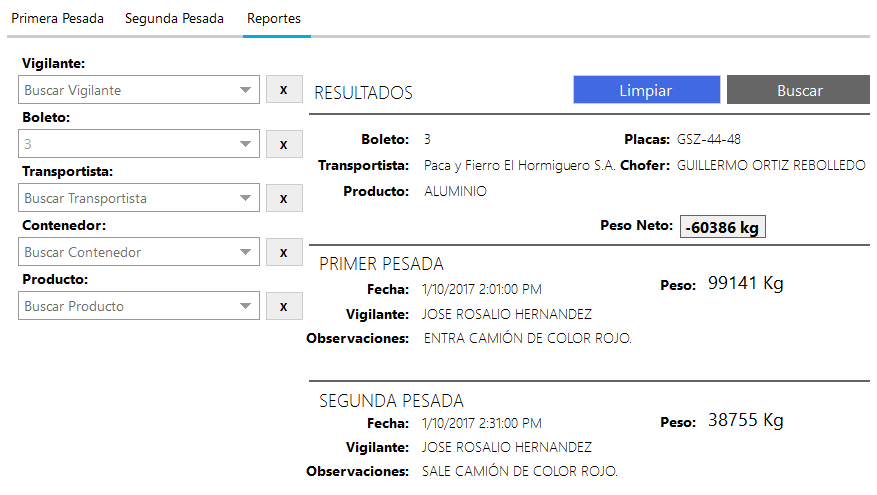
Una vez cargados los datos, se debe mostrar la misma información capturada en el primer pesaje. Además de esta información, se deben abrir nuevos campos, observaciones para salida y una lista desplegable para seleccionar un contenedor. Los contenedores están ligados a un producto, así que se lleva a cabo una validación, dependiendo del producto manejado en el pesaje, se le da la opción al usuario de elegir un contenedor. La tabla Contenedor se modeló de la siguiente manera:



Este módulo sigue el mismo proceso que la primera pesada. Una vez que se llenan los campos requeridos, se procede a validar el usuario y a ejecutar los procedimientos almacenados en la base de datos. Una vez realizado el pesaje, se le notificará al usuario que se llevó a cabo con éxito y se imprimirá un segundo reporte con la información de ambos pesajes.

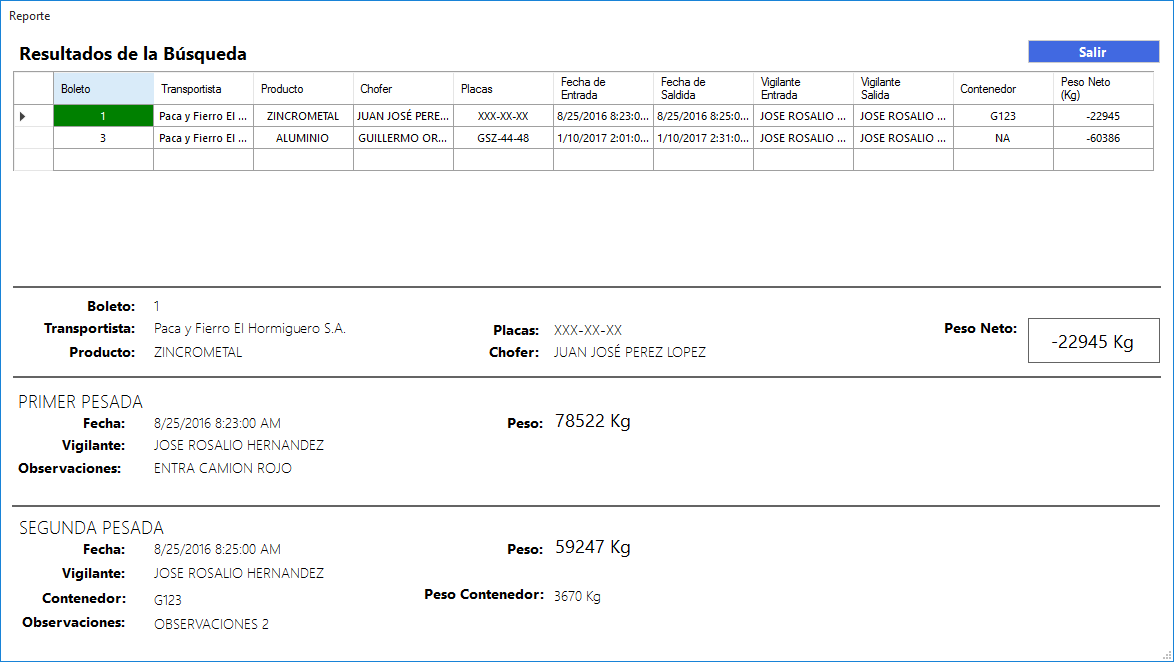


Para el último módulo del sistema de escritorio (**Reportes**) se tienen varios campos en los cuales el usuario debe seleccionar algún elemento como filtro para realizar la búsqueda de reportes en los pesajes registrados en la base de datos. Todos estos elementos son ‘Dropdown Lists’ los cuales, en base al índice seleccionado, se mandan como parámetros a la consulta SQL que se realizará cuando el usuario da clic en ‘Buscar’. En esta sección se creó una ventana modal donde se despliegan los resultados de la consulta que se ejecuta de acuerdo a los filtros de búsqueda de los pesajes. En caso de que la búsqueda solo arroje un resultado, entonces este se muestra en la misma ventana. Para poder lograr eso se concatenaron las diferentes condiciones en la cláusula ‘WHERE’ si el usuario selecciona un vigilante, boleto, producto, contenedor, etc.



Ejemplo de una búsqueda que sólo arroja un resultado, se despliega en la misma ventana. En la izquierda se encuentran los Dropdown Lists donde el usuario selecciona los criterios de búsqueda.

Para el caso de tener varios resultados, se manejó una tabla (**GridView**) donde el usuario puede navegar entre los pesajes (filas) y su respectiva información será mostrada en la parte inferior de la ventana. El usuario puede salir de esta pantalla cuando lo desea.



Ejemplo de una búsqueda donde existen dos o más resultados. Se abre una ventana modal donde el usuario puede ver la información correspondiente a los pesajes que cumplen con el criterio de búsqueda.

En resumen, de forma general, las tareas que se llevaron a cabo para este sistema fueron:

* Listado de requerimientos.
* Modelado de la base de datos relacional.
* Conexión de base de datos SQL Server con Visual Basic.
* Establecimiento de lenguaje de programación, librerías y frameworks a utilizarse.
* Programación de módulo de lectura de datos del puerto serial.
* Programación de módulo de autenticación de equipo.
* Programación de módulo de autenticación de usuario (vigilante).
* Programación de módulo de obtención de segmentos de la dirección IP de los equipos e identificación de la planta en la cual se utiliza el sistema.
* Creación de procedimientos almacenados para inserción de pesajes (primera y segunda pesada).
* Creación de procedimientos almacenados para almacenar información de nuevos productos y transportistas en la base de datos.
* Diseño y codificación de la sección de la primera pesada.
* Diseño y codificación de la sección de la segunda pesada.
* Diseño y consulta de información de pesajes para módulo de Reportes.
* Diseño y codificación de la sección de búsqueda y filtro de reportes.

**Sistema de Báscula Web**

Como se ha mencionado con anterioridad, el sistema de báscula consta de dos partes, la parte de escritorio y la parte web. La primera siendo utilizada por el personal de seguridad en la caseta de vigilancia de cada planta y la segunda dirigida a personal de compras de la empresa. En el portal web del sistema, se tiene acceso a la misma base de datos que su versión de escritorio y este permite ver un listado de todos los catálogos disponibles, así como agregar y modificar los registros.

Algunas de las especificaciones técnicas de este sistema son las siguientes:

* **Visual Studio** como ambiente integrado de desarrollo.
* **C#** como lenguaje de programación orientado a objetos para el lado del servidor o **back-end**.
* **Microsoft SQL Server** como sistema gestor de bases de datos. Utilizando la misma base de datos creada para la versión de escritorio del sistema.
* Por el lado del cliente o **front-end**, además de las herramientas básicas como **HTML** y **ASP.NET** para el **markup**, se utilizó el framework de diseño **Bootstrap** para poder lograr un diseño responsivo de la página y la librería **jQuery** y **jQuery UI** la cual nos permite realizar diferentes tipos de acciones sobre el **DOM** (Document Object Model) de la página y mostrar elementos visuales propios de una interfaz además de usar **Javascript** normal o ‘vanilla’ para realizar distintas operaciones de debugging y entre otras.
* Librería **iTextSharp** la cual nos permite, por medio de código en HTML, generar archivos en formato PDF o XLS de Excel para impresión de informes o reportes.

Una vez teniendo la base de datos estructurada previamente con la versión de escritorio del sistema, fue más rápido y ágil el desarrollo de la versión web. El propósito principal de este portal es poder agregar y modificar elementos a los catálogos, es decir, productos, transportistas, contenedores y vigilantes. El personal de oficinas de la empresa tendrá acceso al sistema y podrá llevar a cabo estas acciones para que los vigilantes puedan registrar pesajes con los nuevos elementos agregados.

Como todos los sistemas utilizados en la empresa, el portal cuenta con una ventana de ‘Login’. Para poder acceder al sistema, el usuario debe ingresar las credenciales correctas las cuales constan de su nombre de usuario (asignado desde que el empleado entra a la empresa) el cual debe estar registrado en el directorio activo y su respectiva contraseña. Para prevenir el ingreso de personas ajenas al sistema incluso dentro de la empresa se implementó el uso de sesiones en el sistema. Una vez que el usuario ingresa las credenciales correctas, se inicializan algunas variables de sesión para identificar que el usuario tiene permiso de usar el sistema.



Panel de Login para sistema web de Báscula.

Al igual que el sistema de escritorio, una vez que el usuario ingresa exitosamente al sistema, se lleva a cabo la extracción de la IP del equipo para identificar desde cual planta se está usando el sistema. Esta información se agrega a la parte superior de la ventana principal donde se encuentra una barra de navegación. En esta se muestran el nombre completo del usuario obtenido del directorio activo de la empresa y la planta donde labora, así como una opción para cerrar sesión.

La forma en la que se estructuró el portal es simple. Consta de una sola página donde se muestra el catálogo de acuerdo a la opción del menú que el usuario elige.

**Sistema de Comprobación de Gastos de Viajes**

**Descripción General**

* El objetivo principal para este sistema es implementar un **módulo de validación** de comprobantes fiscales. En base a todos los documentos proporcionados por el Servicio de Administración Tributaria, se deben llevar a cabo todas las validaciones esenciales al archivo XML de comprobante que proporcionan la mayoría de las empresas al facturar.
* Implementar el módulo de validación en la sección de **reembolso de gastos** de viaje del sistema. Al momento de cargar un archivo XML en la comprobación de algún concepto, el sistema debe validarlo y generar un certificado de validez (PDF) en caso de que el comprobante no pase todas las pruebas indicando los detalles de las validaciones realizadas y que aspectos son incorrectos.
* En la sección para **anticipos de gastos de viaje**, se lleva a cabo el mismo proceso que en un reembolso y se debe implementar el mismo método para la validación de comprobantes.
* En la sección **de modificación para comprobaciones de gastos,** se llevan a cabo cambios a las comprobaciones rechazadas por el personal de compras que se dedica a llevar a cabo la revisión de todas las comprobaciones realizadas en el sistema. Una vez que el estatus de la comprobación cambia, los usuarios pueden ingresar a esta sección a llevar a cabo las modificaciones o correcciones pertinentes para que su reembolso o anticipo pueda ser aprobado. Durante este proceso, los usuarios pueden volver a cargar algún comprobante o registrar nuevos comprobantes con nuevos conceptos (archivos XML). Aquí también se deben realizar las validaciones a los archivos XML y se imprime un certificado para el usuario (PDF) donde se indican los detalles de las mismas.
* En la sección de **Consultas y Consultas Globales** se puede hacer una búsqueda de las comprobaciones realizadas por el usuario, o los usuarios, en caso de tener un perfil de administrador. Aquí se debe agregar la opción para imprimir el certificado de validez de los comprobantes junto con los detalles de la comprobación (PDF).
* Realizar un **programa de consola que descargar todos los archivos** necesarios para realizar las validaciones a los comprobantes. Se descargan desde un servidor FTP del Servicio de Administración Tributaria.

**Descripción Detallada**

H3H3

# 6 Resultados obtenidos

< ¿Se logró el objetivo? ¿hay beneficios? ¿Cómo se logró? Razón por usar la aplicación realizada y resultados >

# Conclusiones y recomendaciones

## Conclusiones

Confrontar los objetivos vs lo que se hizo ¿Qué se planteó hacer o realizar? ¿Qué logró? y ¿cuáles son o serán los beneficios para la organización?

## Recomendaciones

<sugerencias para mejorar personales-externos>

# 8 Referencias bibliográficas y virtuales

## ANEXO A

# Marco Teórico

## Tema 1

## tema 2

## tema 3