

DESARROLLO DEL SERVICIO DE IMPRESIÓN DE TRANSACCIONES
MANEJADAS POR EL DISPOSITIVO CONTROLADOR DE
COMBUSTIBLE DE LA EMPRESA SCI

ANDRÉS FELIPE FONSECA FLÓREZ

UNIVERSIDAD SANTO TOMAS DE AQUINO
DIVISIÓN DE INGENIERIAS
FACULTAD DE INGENIERÍA ELECTRÓNICA
BOGOTÁ
2016

DESARROLLO DEL SERVICIO DE IMPRESIÓN DE TRANSACCIONES
MANEJADAS POR EL DISPOSITIVO CONTROLADOR DE COMBUSTIBLE DE
LA EMPRESA SCI

ANDRÉS FELIPE FONSECA FLÓREZ

MONOGRAFIA PARA OPTAR TÍTULO DE INGENIERO ELECTRÓNICO

TUTOR
EDUARD GALVIS RESTREPO

UNIVERSIDAD SANTO TOMAS DE AQUINO
DIVISIÓN DE INGENIERIAS
FACULTAD DE INGENIERÍA ELECTRÓNICA
BOGOTÁ
2016

Andrés Felipe Fonseca
Estudiante

Diego Andrés Rodríguez
Gerente de Tecnología, SCI Sur América

Ing. Eduard Galvis
Tutor de proyecto de Grado

Bogotá, Agosto de 2016

Este trabajo se lo dedico a mi familia por su amor incondicional, a mi novia por su apoyo en todo el proceso y mis amigos y compañeros por toda su ayuda.

AGRADECIMIENTOS

Agradezco a mi señor padre por haberme brindado su apoyo tanto emocional como económico que permitió culminar mi carrera profesional. Agradezco a mi madre por todo su apoyo incondicional mientras estuve avanzando en mi carrera. Les agradezco el sacrificio y la fe que pusieron en mí en el desarrollo de toda mi educación. Siempre estuvieron brindándome sus experiencias y enseñanzas para formarme como una persona responsable, honesta y profesional.

Gracias a mi hermana que ha sido parte importante en mi desarrollo educativo. Le agradezco haber estado siempre conmigo en todos los momentos y haberme brindado alegría que ayuda a llevar una vida sana.

Doy gracias a la facultad de ingeniería electrónica de la Universidad Santo Tomas y a sus docentes que me formaron como profesional durante todos estos años. Les agradezco todas esas herramientas y conocimiento que me implantaron para permitirme ser ingeniero. Especialmente doy gracias a mi tutor, el docente Eduard Galvis por haber puesto su confianza en mí y haberme orientado en el desarrollo de este trabajo.

Agradezco al ingeniero Diego Andrés Rodríguez que me dió la oportunidad de realizar la pasantía en la empresa SCI Sur América. Gracias por todos esos consejos y por permitirme hacer lo que más me gusta, programar. También quiero agradecer a mis compañeros de trabajo, especialmente el grupo de integración por toda su ayuda y confianza en el desarrollo del proyecto.

Agradezco a Lina María Palencia por haberme brindado su apoyo emocional y afectivo durante todo el desarrollo de la pasantía. Doy gracias por haber puesto su fe en mí y haberme ayudado hasta donde le era posible, incluso más que eso.

Por ultimo quiero agradecer a todos mis compañeros de clase y amigos, con quienes trabaje durante toda la carrera, para que juntos pudiéramos optar nuestro título de ingenieros electrónicos.

CONTENIDO

	Pág.
1. INTRODUCCIÓN.....	10
2. PROBLEMA.....	11
2.1 PLANTEAMIENTO DEL PROBLEMA.....	11
2.2 DELIMITACIONES	12
3. ANTECEDENTES.....	13
4. JUSTIFICACION	15
5. OBJETIVOS	16
5.1 OBJETIVO GENERAL.....	16
5.2 OBJETIVOS ESPECÍFICOS.....	16
6. MARCO TEÓRICO	17
6.1 BASES TEÓRICAS	17
6.1.1 Programación Orientada a Objetos.....	17
6.1.2 Lenguaje de programación C#.....	17
6.1.3 Terminal de punto de venta.	18
6.1.4 Librería Microsoft Point Of Service.	21
6.2 RESEÑA DE LA EMPRESA	21
6.2.1 FuelShield	23
6.2.2 FMFuelSender.....	24
7. DISEÑO METODOLÓGICO	26
8. DESARROLLO DEL PROYECTO	28
8.1 DEFINICIÓN DE PARÁMETROS	28
8.2 CREACIÓN DE LA LIBRERÍA DE ENLACE DINÁMICO.....	31
8.2.1 Clase <i>cPrinterConfig</i>	31
8.2.2 Clase <i>cReciptBuilder</i>	32
8.3 INTEGRACIÓN DE LA LIBRERÍA	33
8.4 RESULTADOS	35
CONCLUSIONES.....	40
REFERENCIAS	41

ANEXOS	43
ANEXO 1. SECCIÓN DE IMPRESORA POS- MANUAL FMFUELSENDER	43
ANEXO 2. FICHA TECNICA IMPRESORA POS EPSON TM-T88IV	45
ANEXO 3. DIAGRAMA UML	47

LISTA DE FIGURAS

	Pág.
Figura 1. Capas del papel térmico	19
Figura 2. Impresora POS EPSON TM-T88IV	20
Figura 3. Controlador de combustible FuelShield	24
Figura 4. Esquema metodológico	26
Figura 5. Plataforma web FleetSAP	29
Figura 6. Parámetros de transacciones en FleetSAP	29
Figura 7. Logo de FleetSAP en la impresión	32
Figura 8. Software EPSON OPOS ADK	33
Figura 9. Ventana de configuración del servicio FMFuelSender	36
Figura 10. Pestaña de configuración de impresora POS	36
Figura 11. Ventana para verificar configuración del servicio FMFuelSender	37
Figura 12. Impresión de transacción	38
Figura 13. Impresión de prueba	39

RESUMEN

En este documento se describe todo el desarrollo del proyecto de la pasantía elaborada por el estudiante en la compañía SCI Sur América. Este es un proyecto de tipo desarrollo de software. Consiste en la integración de una funcionalidad de impresión de las transacciones manejadas por el controlador de combustible *FuelShield* de la empresa. Este controlador permite autorizar el abastecimiento de combustible a vehículos de empresas que son clientes de SCI. También permite llevar un registro de estas transacciones a través de un software encargado de leer estas transacciones y subirlas a los servidores de SCI, dicho programa es conocido como *FMFuelSender*. Las empresas que son clientes pueden acceder a esta información a través de una plataforma web conocida como *FleetSAP*. El objetivo es que al momento de realizar el abastecimiento de combustible los clientes puedan obtener una impresión con información de esta transacción y también se encuentre esta información en *FleetSAP*.

Para realizar el proyecto se desarrolló una librería de enlace dinámico (DLL) la cual contiene los procesos para realizar la conexión de la impresora y la estructura de los datos contenidos en la impresión. Luego se ha utilizado esta librería para integrar la funcionalidad de impresión en el software *FMFuelSender*. Este documento muestra los resultados finales del proyecto.

PALABRAS CLAVE: Servicio de Windows, controlador de combustible, dispositivos de punto de venta, librería de enlace dinámico, abastecimiento de combustible.

1. INTRODUCCIÓN

Uno de los esfuerzos actuales de la ingeniería electrónica es canalizar estos hacia el desarrollo de herramientas informáticas y tecnológicas, motivadas por la gran demanda de productos que pretenden satisfacer necesidades personales del diario acontecer, así como desafíos empresariales que buscan mejorar la eficiencia y productividad en sus modelos de negocios. Uno de los mayores desarrollos informáticos se encuentra en el contexto de las tecnologías de la información y comunicación.

En el modelo de negocios de muchas compañías alrededor del mundo, se incorpora la utilización de internet para ofrecer sus productos, un ejemplo de estas compañías es la organización SCI Holdings LLC (SCI en lo sucesivo). Dentro del portafolio de productos y servicios de la empresa SCI (Scientific Control Instruments por sus siglas en inglés), se incluyen herramientas de soporte para contribuir a la solución de algunos problemas que afrontan aquellas empresas que utilizan flotas de vehículos para realizar sus actividades comerciales. Dentro de los problemas típicos que reportan las flotas de vehículos se encuentran las pérdidas de combustible mayormente por hurtos. Atendiendo necesidades de este tipo, la empresa SCI desarrolla soluciones de hardware y software encargado de la administración de flotas de vehículos y control de combustible. SCI ofrece a las empresas que utilizan flotas de vehículos, un sistema de administración de abastecimiento de combustible para evitar estos problemas, y así aumentar el control y la eficiencia en la utilización de este recurso. La información vehicular para el control de combustible, puede ser consultada y administrada mediante una plataforma informática a través de internet

Cabe resaltar que el control de combustible mencionado, también se puede realizar sin obligatoriamente utilizar internet, en razón a que no siempre las empresas que utilizan flotas de vehículos, cuentan con uno de los dispositivos de hardware diseñados por SCI para tal fin. El trabajo planteado en esta pasantía empresarial, tiene por objeto facilitar la impresión de todos estos registros de abastecimiento de combustible, permitiendo así que los clientes de SCI puedan acceder a esta información sin utilizar internet, lo cual se traduce en una flexibilización del acceso a dicha información vehicular. El desarrollo del trabajo busca impactar positivamente en temas como la logística, supervisión de activos y en concreto en el control actual de administración de abastecimiento de combustible ofrecido por la compañía SCI.

2. PROBLEMA

2.1 PLANTEAMIENTO DEL PROBLEMA

SCI es una compañía que ofrece soluciones para administrar, automatizar y consolidar los procesos de administración de flotas de vehículos y de combustible, mediante herramientas de aplicaciones web y equipos electrónicos dirigidas a empresas particulares que así lo requieran. Una de sus principales soluciones es el control de combustible, servicio que se ha destacado para evitar robos y fraudes en empresas que utilizan este recurso en sus vehículos. El controlador de combustible es conocido como *FuelShield*, este dispositivo está integrado a las bombas en estaciones de servicio de las empresas que solicitan este producto. El controlador se conecta a la red LAN de la estación, un computador que también se encuentra conectado a esta red ejecuta un programa computacional que descarga todos los registros de despacho de combustible del controlador y los sube a la plataforma web de SCI, mejor conocida como *FleetSAP*.

Los clientes de SCI pueden consultar los registros asociados al consumo de combustible, en la plataforma *FleetSAP*. Los clientes también pueden editar parámetros en la administración de su flota, es decir, cambiar, editar o agregar vehículos, conductores, bombas, estaciones de servicio, entre otros. El dispositivo *FuelShield* se encarga de autorizar o negar el despacho de combustible basándose en un código de permiso, información que se maneja en *FleetSAP*. El código de permiso está asociado a cada vehículo lo cual permite al controlador de combustible diferenciar a cuales vehículos se les permite o no realizar el despacho. De igual manera sucede con los conductores, cada uno tiene un código de permiso.

Es necesario la implementación de una funcionalidad de impresión de transacciones del dispositivo de control de combustible de la compañía SCI. Esto debido a que todas las transacciones que controla el dispositivo, son subidas directamente a la plataforma web de la compañía y para realizar consultas, solo se puede hacer desde allí. La funcionalidad de impresión mejora el proceso de control de combustible, en razón a que facilita el acceso rápido a la información de despachos realizados, sin la necesidad de entrar a la plataforma web. Esto permitirá que conductores y encargados del abastecimiento de combustible puedan mantenerse informados de las transacciones realizadas sin utilizar el programa *FleetSAP*, por ejemplo

empresas que no siempre conceden acceso a todos sus trabajadores a esta plataforma.

Para la realización del proyecto, se utiliza una impresora de punto de venta (también conocida como impresora de punto de servicio). El dispositivo de control de combustible está conectado a un computador utilizando la red LAN en la estación de servicio, en este computador se ejecuta el programa que descarga las transacciones del dispositivo y las sube a la plataforma web. Este programa computacional es conocido como *FMFuelSender* y es directamente desarrollado por la empresa SCI. Para el desarrollo del presente proyecto, se requiere agregar al programa computacional, las instrucciones y funcionalidades que permitan que todas las transacciones del dispositivo *FuelShield* que son subidas a la web, también se impriman. Dichas funcionalidades están habilitadas bajo la configuración del usuario en el configurador del servicio *FMFuelSender*.

2.2 DELIMITACIONES

Este proyecto está limitado a la impresión de registros de despacho de combustible con impresoras de punto de venta marca EPSON ofrecidas por SCI, específicamente la referencia TM-P80, modelo que funciona de manera inalámbrica. La empresa escogió el modelo basado en la popularidad que tiene en Estados Unidos además de la posibilidad de poder usarse inalámbricamente, pues permite ser llevado con facilidad. Sin embargo el software será capaz de utilizar cualquier impresora de punto de venta marca EPSON que pueda ser compatible.

La impresión de las transacciones se realiza utilizando un computador que ejecuta un sistema operativo Windows versiones XP o superiores. La impresora debe estar conectada al computador y no al controlador de combustible debido a que el diseño del dispositivo de control de combustible no tiene un puerto adicional al cual se le pueda dedicar una funcionalidad de impresión.

3. ANTECEDENTES

Existen situaciones empresariales en donde es muy importante llevar datos de forma impresa. Una manera adecuada de desarrollar actividades de impresión es utilizar impresoras que no manejen tinta para evitar el desperdicio de cartuchos. Las impresoras que se utilizan para la impresión de recibos también conocidas como impresoras de punto de venta (POS por sus siglas en inglés), son una posible solución al problema del desperdicio de cartucho de tinta, dichas impresoras están hechas para imprimir mucho pero sin usar tinta y aprovechar su funcionalidad térmica.

Existen muchas impresoras POS en el mercado, algunas marcas reconocidas son, SAT, Bixolon, EPSON, Star, APT, entre otras. Sin embargo, la marca EPSON es la más conocida y sus dispositivos son los más sencillos de encontrar comercialmente, una de las mayores ventajas de la marca EPSON es su soporte, pues está disponible en múltiples idiomas, algo necesario en SCI, debido a que la empresa tiene varias sedes en diferentes países. Las impresoras EPSON de la serie TM, son impresoras compatibles con un software llamado EPSON OPOS ADK. Este software es utilizado por desarrolladores para integrar los servicios de impresión a un computador personal (PC). Con este software se pueden ubicar los dispositivos conectados en un PC, asignarles un nombre el cual permita identificarlos e inclusive permite probarlos. Esta información se encuentra disponible para miembros desarrolladores de EPSON Expert [1], debido a las políticas en el contrato de la pasantía en SCI no se pueden mostrar estos documentos, pero si es posible encontrar información en la web [2].

Por otro lado, la compañía SCI también se encarga de fabricar dispositivos para controlar el consumo de combustible de las diferentes empresa que necesitan este recurso, evitando de esta manera el desperdicio y el hurto de combustible, en razón a que dichos dispositivos permiten autorizar o rechazar las solicitudes de abastecimiento de combustible y también permite llevar registro de estas solicitudes. Particularmente, el dispositivo denominado “controlador de combustible *FuelShield 2000*” de SCI no posee un servicio de impresión debido a que la tecnología con que fue diseñado inicialmente no planteaba utilizar esta funcionalidad. Sin embargo, SCI está desarrollando un controlador de combustible que si ofrecerá dicha característica, aun así no se debe desestimar el primer controlador y se buscará la forma de ofrecer este servicio.

Las impresiones hacen que este conocimiento de consumo esté disponible para el personal idóneo que necesite consultarlo, propiciando el ahorro de tiempo ya que no hay necesidad de revisar dicha información en internet, cuando el acceso a dicha red sea limitado. Esto permitirá que el servicio ofrecido por la empresa tenga un valor diferencial para los clientes.

4. JUSTIFICACION

Varios clientes de la empresa SCI, le han solicitado a la empresa un servicio de impresión de transacciones administradas por el dispositivo de control de combustible *FuelShield 2000*. Los clientes han manifestado en diversas ocasiones al servicio técnico de soporte de la compañía, la necesidad de poder entregar información de la transacción a sus conductores sin tener que utilizar internet para ingresar a la plataforma web *FleetSAP*. La realización del proyecto planteado en este documento, ayuda a reconocer rápidamente transacciones realizadas de suministro de combustible para así llevar un mayor control de combustible comparado con el que se lleva actualmente utilizando el mismo dispositivo.

La realización del presente proyecto es de importancia para la empresa porque representa un valor agregado al servicio de control de combustible que ofrece la empresa SCI con este controlador y además que facilita los conductores y encargados del abastecimiento de combustible, la versatilidad de llevar un reporte por transacción del manejo de combustible de cada vehículo sin utilizar la plataforma web *FleetSAP*. Esta última versatilidad cobra sentido en razón a que no todos los empleados de las diversas empresas clientes de SCI, les son concedidos permisos para acceso a la plataforma web *FleetSAP* y para ellos particularmente es importante tener evidencias físicas del abastecimiento de combustible realizado en determinados intervalos de tiempo

Actualmente la empresa SCI está trabajando en el desarrollo de un nuevo controlador de combustible que tendrá integrado una impresora que permitirá realizar la impresión de transacciones, sin embargo el dispositivo conocido como *FuelShield 2000* no deberá estar limitado a poder realizarlas, esto ayudará a que este dispositivo no se deprecie en su valor y continúe siendo utilizado por otras compañías clientes de la empresa SCI.

5. OBJETIVOS

5.1 OBJETIVO GENERAL

Desarrollar y agregar la funcionalidad de impresión de transacciones manejadas por el dispositivo de control de combustible “*FuelShield*” de la empresa SCI.

5.2 OBJETIVOS ESPECÍFICOS

- Realizar un proceso de aprendizaje en programación orientada a objetos para ejecutar el proyecto, debido a que el software con el que se va a trabajar está desarrollado con este lenguaje de programación.
- Diseñar un modelo de recibo impreso con los datos de las transacciones manejadas por el dispositivo de control que son subidas a la plataforma web “*FleetSAP*”.
- Implementar una librería DLL (biblioteca de enlace dinámico), que permita utilizar funciones que sean capaces de estructurar el recibo que contiene la información de la transacción y que permitan realizar la impresión de este.
- Agregar la funcionalidad de impresión en el servicio “*FMFuelSender*”, encargado de subir transacciones manejadas por el dispositivo de control de la compañía SCI.
- Generar una nueva versión del programa de configuración del servicio “*FMFuelSender*” que permita seleccionar la impresora que se va a utilizar o deshabilitar el uso del servicio de impresión.
- Agregar al manual del software “*FMFuelSender*” todo lo relacionado con el uso de la funcionalidad de impresión desarrollada en este proyecto.

6. MARCO TEÓRICO

6.1 BASES TEÓRICAS

6.1.1 Programación Orientada a Objetos.

La programación orientada a objetos es un tipo de programación que utiliza una descripción basada en objetos reales, este tipo de programación representa todos los elementos del mundo tan fielmente como sea posible. El objeto es la unidad básica para la construcción y diseño de este tipo de programación. Las descripciones de estos objetos se encuentran en las clases. Una clase es la representación de un grupo de objetos que comparten características similares, las clases poseen propiedades y métodos.

Las propiedades son la descripción de todos los atributos y características que poseen el objeto. Los métodos son todos los comportamientos y acciones que realiza el objeto en cuestión cuando recibe un mensaje. Comparado con la programación tradicional esto último se puede referir a las funciones.

En este tipo de programación los objetos se comunican entre ellos por medio de mensajes, los mensajes son los encargados de la ejecución de los métodos.

Las clases pueden estar relacionadas entre sí, formando una jerarquía, esto quiere decir que una clase puede heredar todas las propiedades y características de otra clase para formar un nuevo conjunto de objetos con nuevas características y que además comparten características de la clase mayor. [3]

6.1.2 Lenguaje de programación C#

El lenguaje de programación C# es un orientado a objetos, el lenguaje ha sido desarrollado por Microsoft para su plataforma .NET. Consecuentemente las características de abstracción, encapsulamiento, herencia, polimorfismo y modularidad son contempladas en este lenguaje de programación.

El lenguaje de programación C# posee diferentes tipos de variables que permiten estructurar los datos. Por ejemplo, el tipo de datos enteros: *byte*, *short*, *int* y *long*, Representan variables de tamaño de 8, 16, 32 y 64 bits respectivamente. El tipo de

datos de coma flotante: *float*, *double* y *decimal*, representan variables de tamaño de 32, 64 y 128 bits. El tipo de dato de carácter *char* posee un tamaño de 16 bits.

Las clases, por su parte, se definen mediante la palabra clave *class* seguida del nombre que se le desea proporcionar a la clase y posteriormente se definen los elementos que se encuentran dentro de la clase (Métodos, propiedades, etc.) que indican la estructura del objeto.

Como estructuras para la toma de decisión, se utilizan comando como *if*, *switch for*, *while* o *foreach*. El comando ***if***, por su parte plantea una condición. Se utiliza la palabra clave y luego la expresión de decisión, luego dentro de las llaves se plantea el procedimiento en caso de que la condición sea verdadera Ej. *If(a==3){ }*. La instrucción ***Switch*** se utiliza como una forma específica de indicar una instrucción condicional dependiendo del análisis del valor de una variable. Dentro de cada ***case*** se define el procedimiento en caso de que el análisis condicional coincida con el caso. Ej. *switch (i) { case 1: ... break; case 2: ... default: ... break; }*. Por otro lado, el comando ***For***, se utiliza para recorrer el código encerrado en los corchetes dependiendo de la condición descrita en el comando. Además permite la inicialización de una variable y permite definir una acción final antes de terminar cada iteración. Ej. *For (int i = 0; i < 10; i++) { }*. El comando ***While***, recorre el código encerrado en los corchetes siempre y cuando que se cumpla la instrucción condicional del mismo. Ej. *while (i < 10) { }*. Do-While: Recorre el código encerrado en los corchetes siempre y cuando que se cumpla la instrucción condicional del mismo. A diferencia del control *While*, esta condición se consulta al finalizar la ejecución de la primera iteración. Finalmente, el comando ***Foreach*** recorre todo el código encerrado en él para cada elemento de una colección. Ej. *foreach (char c in charList) { }*. [4]

6.1.3 Terminal de punto de venta.

Los terminales de punto de venta son aquellos dispositivos que se utilizan en los procesos comerciales, de cobro y salida de mercancía. Son comúnmente utilizados en restaurantes, mercados y todos aquellos establecimientos que tienen que ver con estos procesos. Generalmente un terminal punto de venta está compuesto por los siguientes dispositivos, monitor, scanner de código de barras, lector de tarjetas, impresora de recibos, entre otros. Una de las principales características de estos dispositivos es que son ensamblables y siempre están interconectados, además de que son pequeños y no ocupan mucho espacio.

El desarrollo de este proyecto se centra en la impresora de recibos, esta impresora es un dispositivo capaz de emitir los comprobantes de ventas, vouchers, facturas y reportes necesarios. Estas impresoras generalmente son conocidas como impresoras POS (en inglés Point Of Sale), estas son pequeñas, ágiles en el proceso de impresión y no consumen tinta [5].

Las impresoras POS son impresoras térmicas, es decir utilizan el calor para realizar la impresión. Utilizan un papel termosensible, este se vuelve negro al contacto con unas agujas a alta temperatura, quedando así una impresión [6].

El papel térmico posee una cara opaca al cual se le aplica calor, este proceso provoca que la otra cara del papel se oscurezca como si estuviese quemada quedando así un dibujo. Esta otra cara del papel es brillante y posee unos compuestos químicos que permiten esta reacción al calor [7]. El papel térmico posee tres capas que se pueden ver en la figura 1.



Figura 1. Capas del papel térmico [8]

Capa Soporte: Esta capa es donde quedara impresa la imagen y es la base de estucado que se le aplica al papel.

Capa de Preestuco: Es una capa lisa que permite garantizar una superficie uniforme del papel el cual ayuda a darle definición a la imagen.

Capa Térmica: Sus principales compuestos son colorante, sensibilizador y desarrollador de color, estos compuestos son los que reaccionan entre ellos al ser sometidos al calor. [8]

El papel térmico tiene ciertas ventajas con respecto al papel común, son económicos, las impresoras que lo utilizan son silenciosas, al no utilizar tintan podrían considerarse ecológicos, se pueden considerar fiables debido a que no utilizan un sistema mecánico complejo si no simplemente un sencillo sistema de transporte de papel causando que no necesite demasiado mantenimiento por averías. [7]

La empresa SCI ofrecerá a sus clientes la impresora EPSON TM-P80, una impresora que funciona inalámbricamente a través de la red WIFI o por medio de conexión USB, esta impresora maneja papel de 80mm de ancho. Debido a la limitante por el alquiler de la impresora es necesario la búsqueda de modelos compatibles o similares, se encontraron los siguientes modelos que pueden cumplir con esto. La compañía EPSON ofrece las referencias TM-T20 y TM-T88IV (ver figura 1), las cuales están disponibles en el mercado de alquiler y son modelos muy utilizados en nuestro país, funcionan por medio de conexión USB y también utilizan papel de 80 mm de ancho.



Figura 2. Impresora POS EPSON TM-T88IV [9]

6.1.4 Librería Microsoft Point Of Service.

Esta librería permite que los desarrolladores que utilizan *.NET Framework* puedan utilizar las terminales de punto de venta en sus proyectos. La librería *Point of Service* proporciona una manera simple y consistente de interactuar con los dispositivos de punto de venta, permite diseñar un conjunto de interfaces y clases que ayudan a la creación de software que ayuda a los vendedores en el uso de dispositivos comunes tales como cajones de efectivo, lectores de códigos de barras, impresoras de recibos y que poseen una funcionalidad “*Plug and Play*”.

La librería posee tres clases importantes para trabajar con impresión, una se llama *–PosPrinter–*, la cual permite trabajar con los desarrollos necesarios para utilizar una impresora POS. La segunda clase es llamada *–PosCommon–*, esta define la interfaz de programación de las constantes, propiedades y métodos que son comunes a todos los dispositivos de terminal de venta. Y por último una clase llamada *–PosExplorer–*, que permite la inicialización de los dispositivos POS.

PosExplorer posee métodos que permiten encontrar las impresoras instaladas en el equipo que se está ejecutando un servicio llamado *MicrosoftPointOfService* y permite la creación de un objeto de tipo *PosPrinter* basándose en el nombre lógico que posea una impresora.

PosCommon posee métodos que permiten conectar y desconectar la impresora a la aplicación *.NET*, también permite saber el estado de la impresora, es decir, si la impresora esta enlazada o no.

PosPrinter posee métodos que ayudan con la impresión caracteres, el corte de papel e impresión de imágenes en mapa de bits. También permite obtener información de datos de la impresora, como lo son la cantidad de caracteres que acepta en el ancho del papel. Y permite conocer información actual sobre la impresora, como lo son, falta de papel en la impresora o la tapa del compartimiento del papel abierta [10].

6.2 RESEÑA DE LA EMPRESA

SCI Sur América SAS, es una empresa colombiana parte de la división de SCI Holdings LLC que tiene sede en Clearwater, Florida. Estas empresas trabajan en conjunto para su marca *FleetSAP*. La marca desarrolla soluciones de hardware y software encargado de la administración de flotas de vehículos y control de combustible. En Colombia se encuentran las instalaciones de investigación y desarrollo de esta marca [11].

Según el objeto social, SCI Sur América SAS se dedica a “la prestación de los servicios consultoría, asesoría, control, instalación y administración de herramientas y soluciones informáticas así como la construcción, desarrollo, evaluación, comercialización, distribución de todo tipo de herramienta informática, tecnológica, hardware, software y servicios afines; dentro y fuera del país. La representación, comercialización de elementos, materiales, tecnología y servicios afines. Se realizan servicios de evaluación, preventa, pos venta e implementación de software, hardware, elementos tecnológicos que sirvan para la cobertura interna y externa de los diferentes sistemas desarrollados, el soporte técnico, comercialización y representación de elementos y materiales de tecnología y servicios afines con la función de desarrollo” [12].

La marca *FleetSAP* ofrece una aplicación web fácil de usar que proporciona a las empresas y organizaciones la versatilidad de gestionar, automatizar y consolidar todos sus procesos de flota de vehículos en un solo sistema. El sistema está hecho para ayudar a hacer frente a las incidencias, optimizar procesos, mejorar el rendimiento, garantizar el cumplimiento, reducir costes y mitigar el riesgo en todos los procesos relacionados con la administración de flota de vehículos.

La empresa SCI ofrece 6 principales productos/servicios a saber [13][14]:

- *FuelShield*,
- *LevelShield*,
- *TrackShield*
- *PointOne*
- *MileShield*
- *VisionShield*

FuelShield proporciona a los administradores de flotas una variedad de soluciones de dosificación de combustible y de seguimiento de registro de dispensación. *LevelShield*, por su parte, es un módulo para la administración de los tanques de almacenamiento que ayuda a obtener un adecuado control de inventario de combustible. *TrackShield*, es el sistema de control de flota que proporciona ubicación en tiempo real, registro de viajes de vehículos y permite hacer seguimiento a las actividades de conducción. El producto *PointOne* es un servicio que proporciona información del vehículo de forma automática por medio de la

computadora de este, estos datos son: kilometraje registrado por el odómetro, horas en el que el motor estuvo encendido, consumo de combustible, temperatura del motor, nivel de batería, entre otros. *MileShield*, es un servicio que proporciona información del vehículo relacionada únicamente a la información entregada por el odómetro y horas en el que el motor estuvo encendido. Finalmente, el producto llamado *Vision* (también se conoce como *VisionShield*) proporciona la protección del almacenamiento de combustible, bombas, perímetros y otros activos a través de la transmisión de vídeo en directo [15].

6.2.1 FuelShield

Hablando específicamente sobre el *FuelShield*, que interviene directamente en el proyecto, este es el dispositivo de control de combustible de SCI, se coloca directamente sobre una bomba de combustible existente o sobre un depósito de combustible móvil (ver figura 3). Este dispositivo permite realizar un seguimiento preciso de cada suministro de combustible también permite autorizar y escoger este suministro dependiendo de una identificación que tiene cada vehículo. Estas identificaciones pueden ser ingresadas por un teclado numérico, por una tarjeta de banda magnética, una llave inteligente conocida como llave Dallas, que posee un número único y un número programable, o un Tag RFID este último es un dispositivo que envía un número a través de radio frecuencia a la bomba para identificar un vehículo.

El sistema de *FuelShield* es capaz de registrar la hora y fecha de la transacción, también puede registrar el nombre del empleado o conductor por medio de una identificación, registra el kilometraje obtenido por el odómetro del vehículo y las horas en el que el motor del vehículo ha estado encendido, adicionalmente también registra el tipo de combustible por medio de un código que escoge el cliente.



Figura 3. Controlador de combustible FuelShield

6.2.2 FMFuelSender

FMFuelSender es el software desarrollado por SCI que lee las transacciones del dispositivo controlador de combustible *FuelShield*. Este programa es un servicio Windows, es decir, su ejecución se hace en segundo plano y administrado por el sistema operativo. El software toma todas las transacciones del dispositivo controlador de combustible y las guarda dentro del disco duro del computador que se encuentra ejecutando Windows XP o superior, estas quedan de respaldo allí. Todas estas transacciones leídas son subidas a los servidores de SCI en donde pueden ser consultadas a través de la plataforma *FleetSAP*. El programa *FMFuelSender* se comunica con los servidores de SCI por medio de WebServices, estos son servicios instalados en la web y permite la comunicación entre la base de datos en el servidor de SCI y el programa, evitando así que hackers u otro tipo de personas puedan conocer la estructura de la base de datos o inclusive manipularla. El software se comunica con el dispositivo por un puerto serial RS232, sin embargo, al instalar estos dispositivos se utiliza un módulo que convierte ese protocolo RS232 a una comunicación LAN inalámbrica, es decir, Wi-Fi, este módulo es conocido como Wi-Box y es proporcionado por una empresa llamada Lantronix que lo desarrolla. Es decir, tanto el módulo como el computador donde se ejecuta el

programa debe estar en la misma red, esta forma de utilizar el *FMFuelSender* se conoce como “*CommonMode*”.

Este software *FMFuelSender* puede ser configurado como “*ServerMode*” y es instalado en los servidores de SCI. Esto sucede así porque el dispositivo *FuelShield* es capaz de transmitir las transacciones utilizando comunicación GPRS, señal de datos celular. Es decir que todas las transacciones pueden llegar directamente a los servidores de SCI y el programa *FMFuelSender* se encarga de organizar y enviar estos mensajes al WebService. Este modo conocido como “*ServerMode*” puede ser utilizado por los clientes de SCI que posean servidores y utilizarlo de esta manera si así lo desean.

El programa tiene múltiples opciones para realizar su función cuando se configura en “*CommonMode*”, como por ejemplo selección del intervalo de tiempo para leer el controlador de combustible, aprobación de publicación de transacciones de cantidad 0 al WebService, actualización de la hora y fecha del dispositivo controlador de combustible, entre otras. Por último, al software se le puede conectar uno o múltiples controladores de combustible siempre y cuando se encuentren en la misma red LAN, gracias al dispositivo Wi-Box de Lantronix. El servicio de impresión solo podrá realizarse en el modo de configuración “*CommonMode*”.

7. DISEÑO METODOLÓGICO

Para el desarrollo del proyecto se utiliza un tipo de investigación formativa, en la cual se ha hecho una revisión de la documentación de Microsoft para realizar programas en el software Visual Studio y de los productos pertinentes de EPSON para realizar el proyecto. Se ha revisado la documentación relacionada con la programación orientada a objetos y con el lenguaje de programación C# ofrecida por Microsoft. Además se ha tenido en cuenta toda la documentación relacionada a *Microsoft POS for .NET V.1.12*. Esta última versión se ha elegido debido a que es compatible desde el sistema operativo *Windows XP* en adelante.

Se define todo el proceso de diseño e implementación a partir de un diagrama en bloques (ver figura 4).

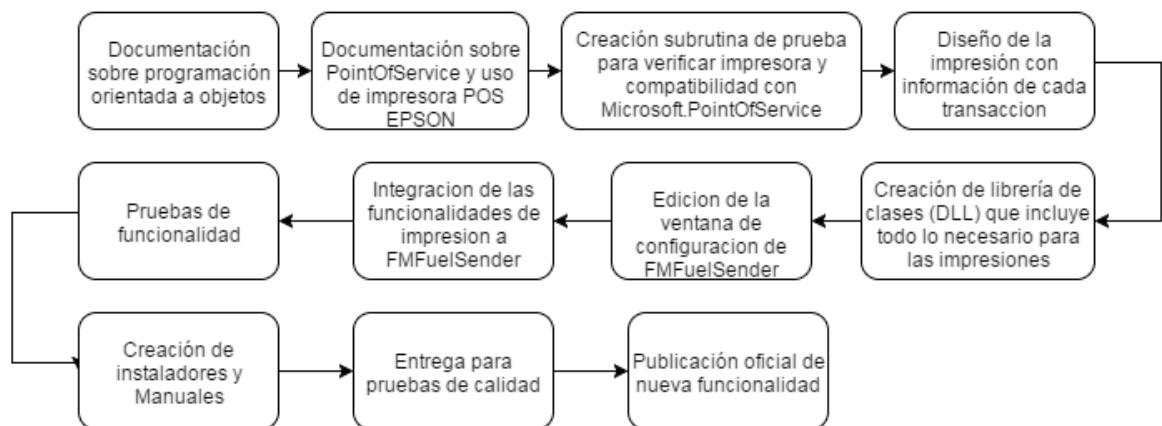


Figura 4. Esquema metodológico

La pasantía comienza con un proceso de documentación acerca de programación orientada a objetos y sobre la librería *Microsoft PointOfService*. Esto se realiza para la creación de una subrutina de prueba que permite verificar la funcionalidad de la impresión y la compatibilidad de la librería y adquirir el conocimiento para realizar el proyecto. Luego de este proceso se continua con el desarrollo para el proyecto para la empresa SCI, en donde se elabora una librería de enlace dinámico (DLL) que permite estructurar los datos en la impresión y la conexión a la impresora. Se integra esta librería al software *FMFuelSender* para imprimir transacciones controladas por el dispositivo *FuelShield*. Por último se realizan pruebas y el manual de instalación y configuración. Se entregan al área de calidad, en SCI Distribution

también con sede en Clearwater, para la evaluación y publicación oficial de la nueva funcionalidad.

Las técnicas por las cuales se obtiene información acerca del proyecto a desarrollar, son simplemente a través de la indagación bibliográfica de la documentación encontrada. Para procesar la información encontrada en los documentos se realizan subrutinas en archivos ejecutables. Dichas subrutinas permitieron verificar el conocimiento adquirido y permitieron que el estudiante confirme su funcionamiento.

Para realizar el software se utiliza el ambiente de desarrollo integrado llamado *Microsoft Visual Studio 2013 Community* (se puede descargar de la página de este [16]) que permite desarrollar programas de computador bajo el sistema operativo *Microsoft Windows*. Además se dispone de una impresora POS de marca EPSON (TM-T88IV). Por último se tuvo la unidad principal del controlador de combustible *FuelShield 2000* que permite generar transacciones de despacho de combustible para verificar las impresiones al finalizar el proyecto.

8. DESARROLLO DEL PROYECTO

El desarrollo de esta pasantía se orienta primordialmente al desarrollo de software, específicamente utilizando conceptos de programación orientada a objetos. Dichos conceptos se tuvieron que estudiar al principio de la pasantía debido a que el estudiante responsable no conocía acerca del tema al empezar el proyecto.

Al principio de la pasantía se elabora un cronograma de actividades el cual sirve de guía para la elaboración del proyecto, también se utiliza un documento de especificaciones creado con anterioridad por uno de los desarrolladores de la compañía, este documento es exigido por la empresa con la finalidad de tener ideas claras y poder facilitar todo el proceso de los desarrollos. En este documento se explica detalladamente la finalidad del proyecto así como una guía detallada de los elementos que componen el proyecto: Objetivos, librería de acceso dinámico a utilizar, impresora a utilizar, formas de Windows que se diseñaran, elementos en la configuración, entre otros.

El desarrollo del proyecto implica agregar a la ventana de configuración del servicio de Windows *FMFuelSender*, las opciones necesarias para utilizar el servicio. En el documento de especificaciones del proyecto se define que se deben imprimir las transacciones del dispositivo *FuelShield* justo después de publicar estas a la plataforma web, que en resumen esto significa que se imprimen las transacciones justo después de haber sido publicadas en la base de datos por el programa *FMFuelSender*. Por último se ha documentado toda la información pertinente a este servicio de impresión para facilitar la configuración y uso de esta funcionalidad a otros usuarios.

8.1 DEFINICIÓN DE PARÁMETROS

Para definir los parámetros de cada transacción que se van a imprimir, es necesario conocer el funcionamiento de *FleetSAP* (ver figura 5), la plataforma web en la que acceden los clientes de la empresa SCI y donde pueden administrar su flota de vehículos, la dosificación de combustible de estos, su inventario de combustible, etc.

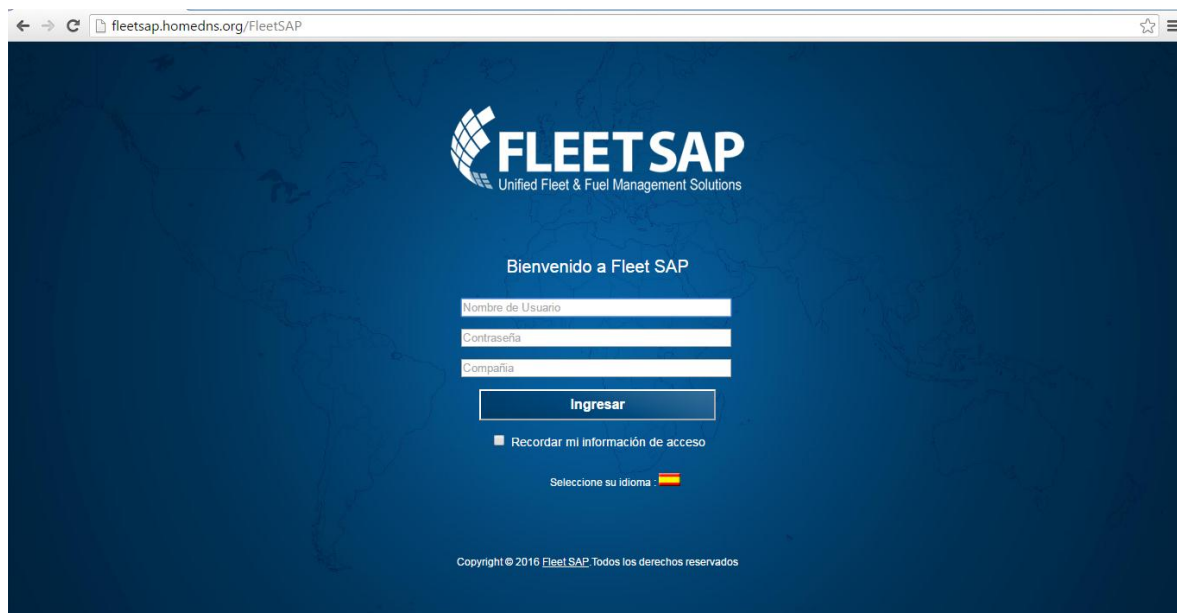


Figura 5. Plataforma web *FleetSAP*

Al principio de la pasantía se considera realizar una inspección del funcionamiento de *FleetSAP* para adquirir el conocimiento e identificar el uso de la plataforma. *FleetSAP* tiene un módulo llamado *suministro de combustible* en él se encuentra los parámetros que el cliente puede revisar tomando así la decisión de escoger estos parámetros para imprimir (ver figura 6).



Figura 6. Parámetros de transacciones en *FleetSAP*

Parámetros escogidos:

-*TagID*, es un número hexadecimal de 10 caracteres que permite identificar el vehículo en la plataforma, este código es único por compañía, es decir, cada empresa cliente tiene un solo TagID para un solo vehículo. El dispositivo *FuelShield* consulta este número dentro de su memoria para permitir autorizar transacciones.

-*ProjectID*, es un número entero positivo que permite identificar un grupo de vehículos que trabajan con un fin específico, es decir, un código que permite agrupar vehículos.

-*CardNumber*, este es un número de 10 dígitos que permite identificar los empleados o conductores, al igual que el TagID cada empresa cliente tiene un solo CardNumber para un solo empleado.

-*Odometer*, este número representa el valor del kilometraje registrado por el odómetro del vehículo que tiene al momento de realizar la transacción.

-*HoursCounter*, este número representa el valor en horas del tiempo en el que ha estado el vehículo encendido desde que se realizó la última calibración hasta el momento en el que se realiza la transacción.

-*RecordType*, este es un número entero positivo establecido por SCI para identificar el tipo de transacción, esto quiere decir que permite identificar transacciones de ingreso de combustible o salida de combustible al tanque enlazado a la bomba y si estas son adicionadas manualmente en la plataforma web o son realizadas con el controlador de combustible.

-*DateTime*, esta es la hora y fecha de la realización de la transacción.

-*CompanyID* es un número entero positivo que permite la identificación de la compañía cliente en la base de datos de los servidores de SCI.

-*PumpName*, es la identificación o nombre que le dio el cliente a la bomba en la plataforma web *FleetSAP*.

-*TransactionState* es el estado en el que se encuentra la transacción, es decir que permite saber si la transacción ya se encuentra subida a los servidores de SCI o si se encuentra pendiente de subir, etc.

-*FuelPermCode*, conocido como el código de permiso, este número entero positivo de 10 dígitos permite establecer una autorización al empleado o conductor, el dispositivo *FuelShield* consulta este número en su memoria interna para permitir autorizar la dosificación de combustible en caso de que sea solicitado, esto depende de la configuración del controlador de combustible.

-*StartCounter*, este es un número que utiliza el controlador de combustible para establecer un contador de galones al inicio de cada transacción.

-*EndCounter*, este número establece el valor final del contador de galones dispensados al final de cada transacción. Ambos contadores permiten calcular la cantidad de galones en cada transacción.

-*Total*, es la diferencia de los contadores y permite identificar la cantidad de galones dispensados en la transacción.

8.2 CREACIÓN DE LA LIBRERÍA DE ENLACE DINÁMICO

Se elabora una librería de clases (DLL o librería de enlace dinámico) para estructurar la funcionalidad de impresión, este fue uno de los requerimientos establecidos por SCI para realizar este proyecto. Realizar esta librería facilitaría los cambios en el diseño de las impresiones, ya que simplemente se cambia la librería por otra para cambiar este diseño.

La librería de clases posee dos clases, *cPrinterConfig* y *cReceiptBuilder*. La primera se encarga de realizar la conexión y desconexión de la impresora y todas las actividades relacionadas a ella, por ejemplo el corte de papel, la impresión de líneas y la impresión del logo de *FleetSAP*, etc. La clase *cReceiptBuilder* se encarga del diseño de la impresión, establece como está organizado los datos, cantidad de caracteres que tiene el ancho de la hoja de la impresora, alineaciones de los textos en la impresión, y detalles de decoración, etc. El archivo generado (DLL) (librería de enlace dinámico) se nombró como “**RecepitPOSPriter.dll**”. Debido a la cláusula de confidencialidad establecida con SCI y el estudiante, no se pueden detallar los métodos creados allí ni la descripción de estos, pero si se pueden mencionar los métodos utilizados de *Microsoft PointOfService*. Adicionalmente se realizó un diagrama UML en donde se encuentra un poco más detallada esta información (ver anexo 3).

8.2.1 Clase *cPrinterConfig*

GetDevice(DeviceType, LogicalName) es un método de la clase *PosExplorer* que es utilizado para obtener un objeto de tipo *PosPrinter* y realizar la conexión con la impresora. *DeviceType* es el tipo de dispositivo es decir Impresora POS y *LogicalName* es el nombre que identifica la impresora.

Open() y Claim(WaitTime) son métodos de la clase *PosCommon* que establecen la conexión con la impresora. *Open* abre la conexión con la impresora y *Claim* establece el tiempo límite de espera para conectar a la impresora.

PrintNormal(PrinterStation, StringText) este método de la clase *PosPrinter* permite realizar la impresión, *PrinterStation*, se refiere al tipo de estación en la impresora, en este caso establecida para imprimir recibos. *StringText*, Texto a imprimir.

Close() este método de la clase *PosCommon* cierra la conexión con la impresora.

PrintBitMap(PrinterStation, Path, Aline) este método de la clase *PosPrinter* realiza la impresión de una imagen mapa de bits. *PrinterStation*, se refiere al tipo de estación en la impresora, en este caso establecida para imprimir recibos. *Path*, ubicación del archivo de imagen. *Aline*, alineación de la imagen en este caso, centrado.

CutPaper() este método de la clase *PosPrinter* realiza el corte del papel solamente se puede ejecutar si la impresora posee el corte automático, en caso contrario se debe saltar el método.

8.2.2 Clase cReceiptBuilder

Esta clase posee todos los métodos que organizan los datos y establecen el diseño de la impresora. Debido a la cláusula de confidencialidad no se puede describir los métodos que se encargan de esto. La impresión está organizada de la siguiente manera, en la cabecera de la impresión se encuentra el logo de **FleetSAP** (ver figura 7), luego se encuentra la información sobre hora y fecha en la que se realiza la impresión, después todos los datos relacionados a los vehículos como TagID, kilometraje registrado por el odómetro, etc. Luego se imprime la hora y fecha de la transacción. Posteriormente se imprimen los últimos datos relacionados a la autorización de la transacción e información de ella por ejemplo nombre de la bomba, contadores, etc. Finalmente se imprime un mensaje de agradecimiento como pie de página.



Figura 7. Logo de FleetSAP en la impresión

8.3 INTEGRACIÓN DE LA LIBRERÍA

El servicio *FMFuelSender* desarrollado por la empresa SCI, posee una clase llamada *cFsSanttigs*. Esta clase describe un objeto encargado de las configuraciones del servicio. A esta clase se agregaron dos parámetros para configurar la impresora, uno es un valor booleano que indica si el servicio debe o no debe imprimir. Y el segundo obedece al nombre lógico que posee la impresora, este dato permite identificar la impresora instalada.

El usuario puede definir el nombre lógico a una impresora con un software adicional llamado EPSON OPOS ADK (ver figura 8). Este software permite administrar todos los dispositivos de punto de venta instalados en un computador. También permite hacer pruebas. Este se puede descargar de la página de soporte de EPSON [17].

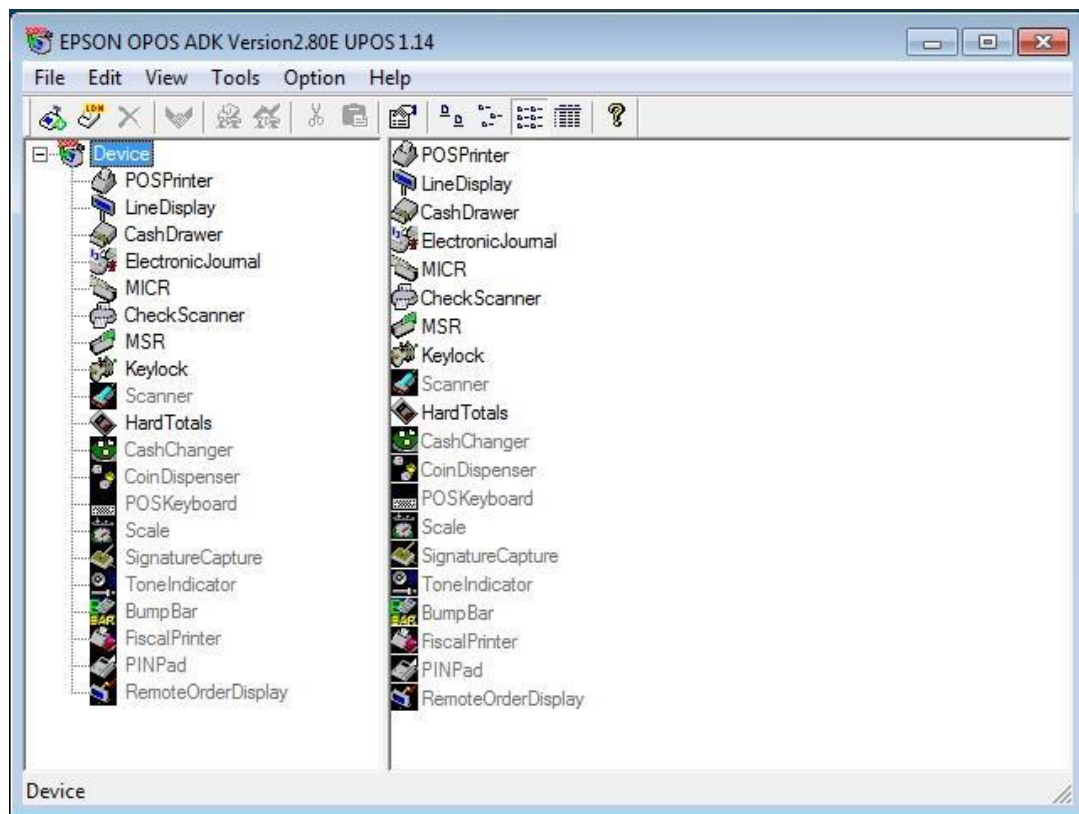


Figura 8. Software EPSON OPOS ADK

El servicio *FMFuelSender* posee una ventana de configuración que no solo permite escoger las opciones para utilizar el servicio si no también permite iniciar y detener el servicio. Allí se agregó ambos parámetros de utilización de la impresora. La clase

cFsSanttigs posee un método que permite guardar los parámetros de configuración en un archivo de extensión (xml). Estos archivos permiten guardar datos y son muy útiles porque tienen mucha facilidad al acceso de estos datos. En SCI este tipo de formato se utiliza para almacenar las configuraciones de todos los programas y desarrollos.

Para escoger el nombre lógico de las impresoras instaladas en el equipo y que anteriormente fueron configurados con el software EPSON OPOS ADK. Se utiliza un combo box, esta utilidad permite escoger datos a partir de una lista. Para llenar esta lista se utiliza un método de la clase *PosExplore* de la librería de Microsoft *PointOfService*. Este método se conoce como *GetDevices(DeviceType)* y retorna una lista de dispositivos encontrados del tipo *DeviceType* en el sistema operativo. Para este caso el tipo es *impresora POS*.

Cuando el usuario cambia de impresora de la lista en la lista desplegable (combo box), se llena una caja de texto con información importante recuperada de la impresora. Como por ejemplo, descripción de la impresora, fabricante de la impresora, versiones de hardware, los nombre lógicos que posea la impresora, entre otros. Es posible asignarle más de un nombre lógico a un dispositivo POS con la herramienta EPSON OPOS ADK. Es decir que en la caja de texto se enlistaran todos los nombres lógicos de que tenga asignada la impresora, sin embargo en el combo box solamente aparecerá el primer nombre lógico asignado a esta.

En el desarrollo del proyecto también se incluye un botón de prueba de la impresora en esta ventana de configuración. Al hacer clic el usuario podrá imprimir una página de prueba en la impresora seleccionada. Y podrá observar la estructura de la impresión. Para realizar esta impresión, se utiliza la librería *ReceiptPOSPrinter.dll* elaborado por el estudiante en este proyecto.

Para tomar la decisión en la utilización de un logo apropiado para imprimir, se utiliza una imagen del tamaño 160 pixel por 50 pixel, sin embargo para llegar a esto se realizaron muchas impresiones buscando la que mejor se pudiera observar, es decir realizando transacciones hasta escoger el tamaño apropiado del logo para imprimir, esta decisión es tomada por el equipo de diseño gráfico de SCI.

Estando en ejecución, el servicio *FMFuelSender* debe imprimir las transacciones que ha leído del dispositivo *FuelShield* una vez el servicio haya subido estas a los servidores de SCI. Así que para realizar el proyecto se debe inspeccionar el código del servicio. Una vez encontrado en el código el proceso de subida de transacciones

a los servidores, se agrega allí toda la funcionalidad de impresión utilizando la librería (*ReceiptPOSPrinter.dll*) creada anteriormente.

Una vez terminado todo el desarrollo y posterior a la realización de todas las pruebas hechas por el desarrollador, este debe enviar los instaladores del nuevo software y el manual de uso a SCI Distribution para realizar las pruebas de calidad. Es decir que el estudiante agrega al manual de configuración y uso del servicio *FMFuelSender*, todo lo relacionado a la impresora y también genera el archivo que permite instalar el servicio, para ello se utiliza la herramienta de la empresa Flexera LLC, esta herramienta es conocida como *InstallShield*, y permite generar los archivos de instalación de los desarrollos en el entorno de desarrollo Microsoft Visual Studio.

8.4 RESULTADOS

Se ha desarrollado una librería de enlace dinámico (DLL) en C# que permite utilizar una impresora POS para la impresión de transacciones manejadas por el dispositivo controlador de combustible *FuelShield*. Esta librería se ha nombrado **ReceiptPOSPrinter.dll**.

El proyecto también incluyó la adición de los parámetros de configuración (ver figura 9) que permite utilizar la impresora en la ventana de configuración del servicio *FMFuelSender* (ver figura 10).

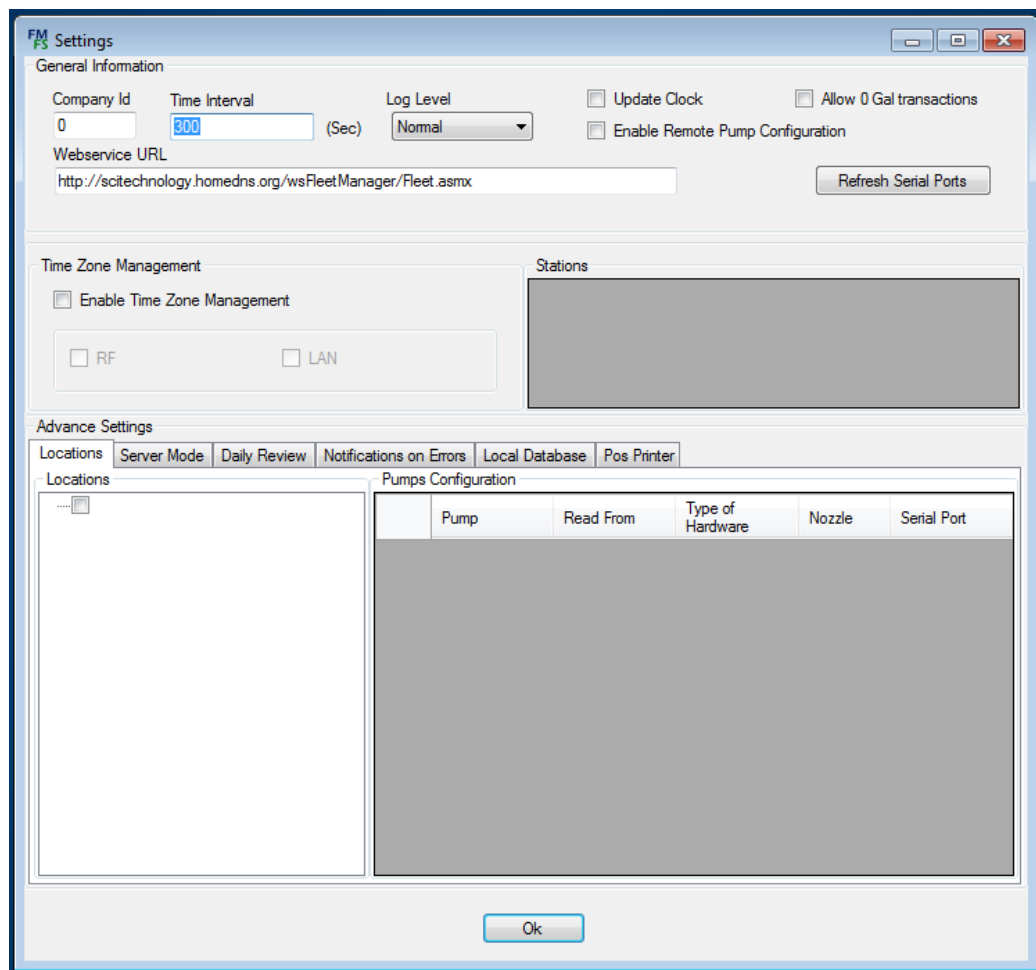


Figura 9. Ventana de configuración del servicio *FMFuelSender*

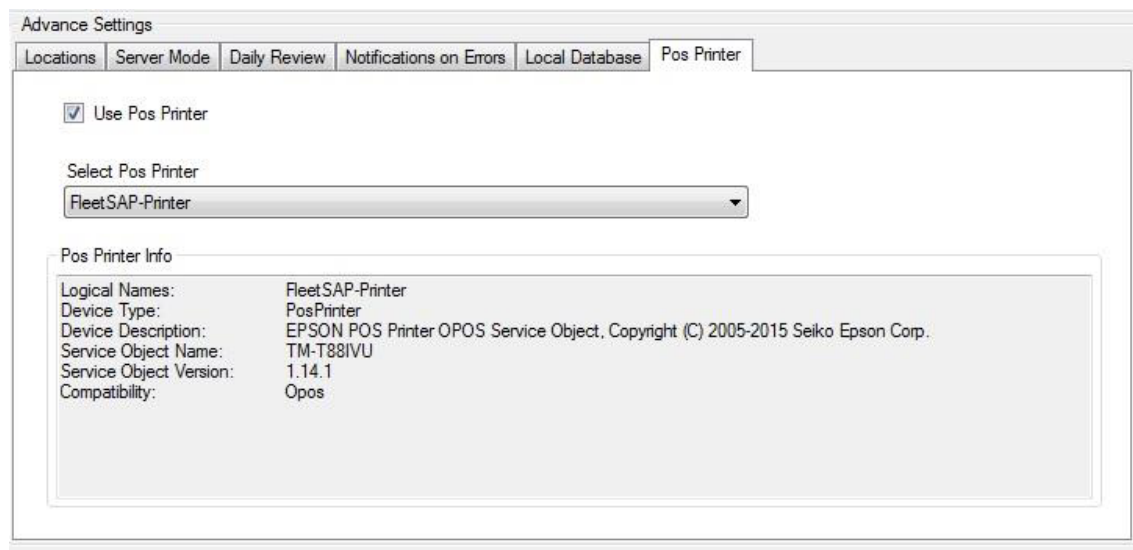


Figura 10. Pestaña de configuración de impresora POS

Se puede observar en la ventana principal de configuración del servicio, una caja de texto con el nombre lógico de la impresora POS. También se puede ver que en caso de no utilizar esta simplemente aparece la palabra “Disable” (ver figura 11).

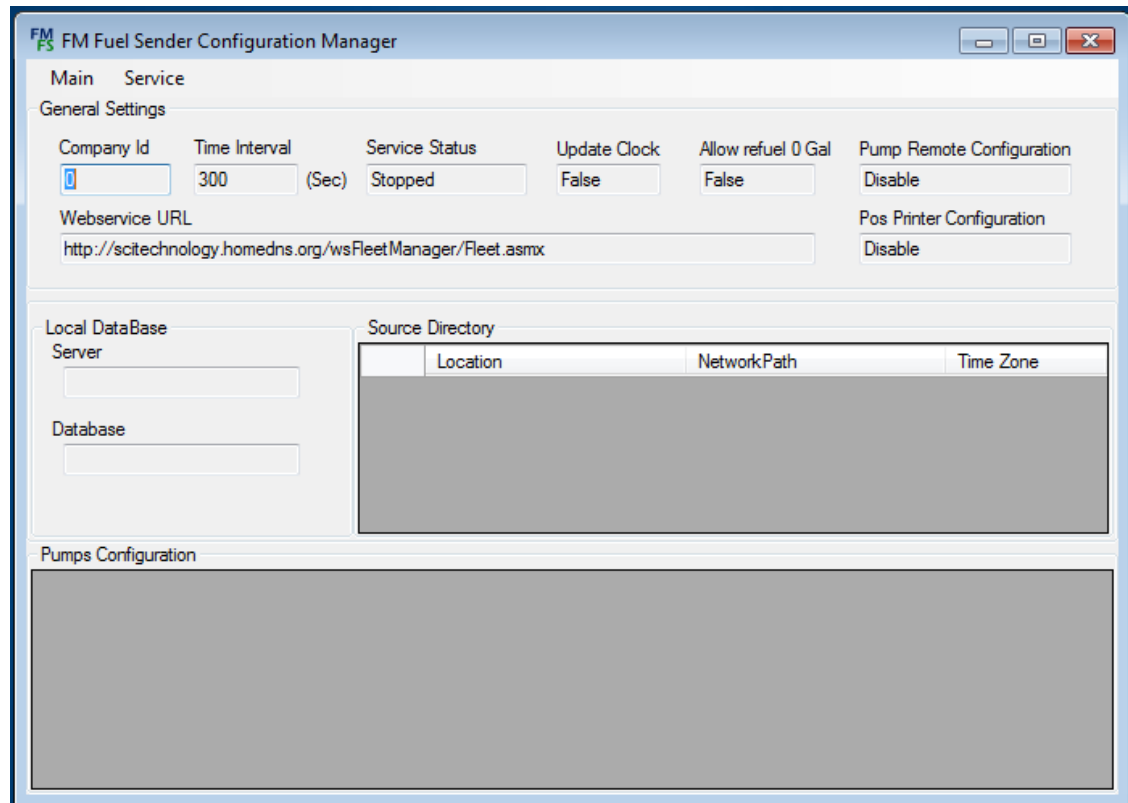


Figura 11. Ventana para verificar configuración del servicio FMFuelSender

En la librería *ReceiptPOSPrinter.dll* se incluye el diseño de la impresión de la transacción pues en ella está la estructura y la organización de los datos de la hoja impresa. Como resultado se obtiene las impresiones o recibos de las transacciones realizadas con el dispositivo (ver figura 12).



-----Print Time-----

Date: 21/04/2016
Hour: 07:38 a.m.

-----Vehicle-----

Tag ID: 9000000003
Project ID: 0
Card Number: 0000000000
Odometer: 1
Hours Counter: 0
Record type: 3

-----Transaction Time-----

Date: 21/04/2016
Hour: 07:38 a.m.

-----Transaction Information-----

Company ID: 2
Pump Name: test001
Transaction State: Updated
Fuel Perm Code: 0000000000
Start Counter: 16801,3
End Counter: 16805,9

Total: 4,6

Thank You

Figura 12. Impresión de transacción

En la pestaña de configuración de la funcionalidad de impresión, se creó un botón que permite realizar una impresión de prueba. Esta impresión permite verificar la configuración de la impresora y observar la estructura de los datos en el papel (ver figura 13).

```

*****
FLEETSAP
United Fleet & Fuel Management Solutions
*****
-----TEST Print Time-----
Date:                                06/04/2016
hour:                                7:48
-----TEST Vehicle-----
Tag ID:                             20304TEST
Project ID:                         0101TEST
Card Number:                        0000TEST
Odometer:                           10203TEST
Hours Counter:                       4TEST
Record type:                         3TEST
-----TEST Transaction Time-----
Date:                               01/01/2016
Hour:                               12:00
-----TEST Transaction Information-----
Company ID:                         01TEST
Pump Name:                          PUMPTTEST
Transaction State:                  LOCALTEST
Fuel Perm Code:                    000000TEST
Start Counter:                      0001TEST
End Counter:                        9999TEST
-----
<<< This is a Test Print Receipts >>>
*****

```

Figura 13. Impresión de prueba

Otro de los resultados del proyecto desarrollado, es la adición de todos los elementos necesarios para utilizar la funcionalidad de impresión en el servicio *FMFuelSender*. La parte relacionada a la configuración que se encuentra en este documento se encuentra en la sección de anexos (ver anexo 1). El proyecto se realizó utilizando una impresora EPSON TM-T88IV (ver ficha técnica en anexo 2).

CONCLUSIONES

- Al término de esta pasantía empresarial se ha entregado de conformidad a lo pactado con la empresa SCI, una librería de enlace dinámico (DLL) en C# que permite utilizar una impresora POS para la impresión de transacciones manejadas por el dispositivo controlador de combustible *FuelShield*.
- Para el diseño de impresión en papel de recibos es necesario tener en cuenta, el tamaño en cantidad de datos necesarios como la cantidad de caracteres por línea, para una correcta visualización. Existen impresoras que poseen corte automático y otras que no, este es un importante parámetro que se consideró en la impresión, para permitir un corte al final del recibo.
- Las librerías ayudan a los desarrolladores a independizar una funcionalidad de un software del programa principal. Esto ayuda a que desarrolladores creen porciones del programa y se tenga acceso más fácil a realizar mantenimiento de código.
- La librería *ReceiptPOSPrinter.dll* puede llegar a ser utilizada en otro desarrollo de SCI que necesite impresión de transacciones de abastecimiento de combustible, por ejemplo un desarrollo para otro controlador de combustible. Pues en ella se encuentran los métodos que organizan los datos en la impresión y estos no suelen cambiar en SCI.
- Los logotipos que se pueden imprimir en impresoras térmicas, deben tener un tamaño acorde al área de impresión e inclusive en la mayoría de impresoras deben ser una imagen monocromática como en el caso de este proyecto, incluir otro tipo de imagen generaron una excepción.
- Realizar pruebas a versiones nuevas de un software, es una tarea muy importante y permite verificar el funcionamiento y una de estas pruebas debe estar orientada a recrear una experiencia de usuario donde el programa se está ejecutando bajo las condiciones que un cliente promedio lo requiera. Los resultados obtenidos han sido de calificación positiva por parte del personal pertinente en la empresa, lo cual permitió avanzar en la realización del manual para entregar el desarrollo a SCI Distribution, cuyo personal estará encargado de las pruebas pertinentes de calidad.
- La documentación y específicamente, los manuales son parte crucial en el desarrollo de software. La documentación permite que los clientes aprendan a utilizar el software. Sumado a esto, los manuales técnicos permiten a los instaladores y empleados de la empresa (SCI en este caso particular) reconocer las nuevas funciones en los servicios que se ofrecen a los clientes.

REFERENCIAS

- [1] EPSON Expert. [En línea] <<http://www.epsonexpert.com>>. [Citado en 07 de junio de 2016].
- [2] LavaBlast Software Inc. Using Microsoft POS for .NET in 2011. [En línea]. <<http://blog.lavablast.com/post/2011/06/06/Using-Microsoft-POS-for-NET-in-2011.aspx>>
- [3] Arenas Hernández Luis, Introducción En: Asignatura de programación orientada a objetos en JAVA [En línea]. <<http://fcasua.contad.unam.mx/apuntes/interiores/docs/98/opt/java.pdf>>. Facultad de contaduría y Administración. Universidad Nacional Autónoma de México. [Citado en 07 de junio de 2016].
- [4] Jose Antonio Gonzales Seco. Estructuras de control en lenguaje C#. [En línea]. <<http://www.devjoker.com/contenidos/Tutorial-C/167/Estrcuturas-de-control.aspx>>. [Citado en 17 de octubre de 2016]
- [5] MBCEStore México. Punto de venta. [En línea]. <<http://www.mbcestore.com.mx/punto-de-venta/>>. [Citado en 07 de junio de 2016].
- [6] Guías Prácticas. Impresora Térmica. [En línea]. <<http://www.guiaspracticas.com/impresoras/impresora-termica>>. [Citado en 27 de septiembre de 2016].
- [7] Ofitor. ¿Qué es el papel térmico? [En Línea]. <<http://www.rollospapeltermico.es/que-es-el-papel-termico-personalizado/>>. [Citado en 06 de Diciembre de 2016].
- [8] Enroyo. ¿Qué es el papel térmico? [En Línea]. <<http://www.enroyo.com/tienda/content/7-que-es-el-papel-termico>>. [Citado en 06 de Diciembre de 2016]
- [9] EPSON TM-T88IV Serie. Sección Galería. [En línea]. <<https://www.epson.es/es/es/viewcon/corporatesite/products/mainunits/overview/8907>>. [Citado en 27 de septiembre de 2016].
- [10] Microsoft. POS for .NET v1.12 SDK Documentation. [En línea]. <[https://msdn.microsoft.com/en-US/library/bb429024\(v=winembedded.11\).aspx](https://msdn.microsoft.com/en-US/library/bb429024(v=winembedded.11).aspx)>. [Citado en 07 de junio de 2016].
- [11] FleetSAP LLC. About us. <<http://www.fleetsap.com/about.html>>. [Citado en 07 de junio de 2016].
- [12] Computrabajo. Información sobre SCI Sur América S.A.S y ofertas de trabajo. [En línea]. <<http://www.computrabajo.com.co/empresas/ofertas-de-trabajo-de-sci-sur-america-sas-6E08B506D43AD8A6>>. [Citado en 07 de junio de 2016].

- [13] PointOne. <<http://sciww.com/point-one/>>. [Citado en 26 de Noviembre de 2016].
- [14] MileShield. <<http://sciww.com/mile-shield/>>. [Citado en 26 de Noviembre de 2016].
- [15] Página principal de SCI Distribution. Sección: “Solutions”. <<http://sciww.com>>. [Citado en 07 de junio de 2016].
- [16] Microsoft. Visual Studio. <<http://www.visualstudio.com>>. [Citado en 30 de julio de 2016]
- [17] EPSON. OPOS ADK. [En línea] <<https://download.epson-biz.com/modules/pos/index.php?page=soft&scat=38>>. [Citado en 30 de julio de 2016].

ANEXOS

ANEXO 1. SECCIÓN DE IMPRESORA POS- MANUAL FMFUELSENDER



FMFuelSender v2.0.31.4 Manual de usuario

FMFuelSender v2.0.31.4

Manual de usuario

Preparado por : Departamento de sistemas.

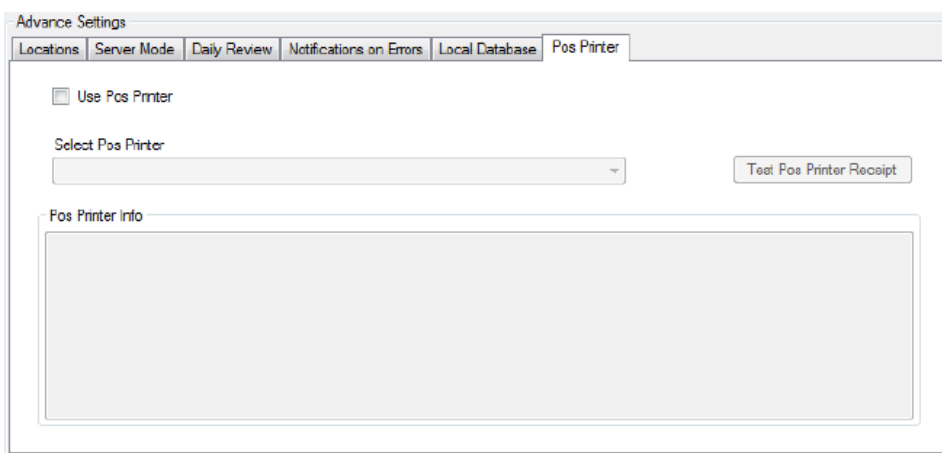
Publicado por : Departamento de sistemas.

Referencia : MANU01SP_FMFuelSenderServiceManualdeUsuario

Fecha de publicación : Abril 2016

IMPRESORA POS (POS PRINTER)

Permite configurar las funcionalidades de impresión del servicio. Las impresoras Pos (punto de venta) son dispositivos capaces de imprimir recibos de forma térmica, es decir necesitan rollos de papel especiales. Esta función permite imprimir en un recibo las transacciones que son subidas a FleetSAP. Para usar esta funcionalidad se recomienda utilizar impresoras EPSON que aceptan papel de 80mm. Antes de realizar la siguiente configuración, revise el anexo 3.



Parámetro	Descripción
Usar impresora Pos (Use Pos Printer)	Permite habilitar las funcionalidades de impresión de transacciones que son subidas a FleetSAP.
Seleccionar Pos Printer (Select Pos Printer)	Este parámetro permite seleccionar el nombre lógico asignado a la impresora en donde se realizarán las impresiones. Para asignarle un nombre lógico, revisar anexo 3.
Información de impresora Pos (Pos Printer Info)	En este campo se puede observar la información de la impresora, esto es útil para ver en detalle toda la información disponible de la impresora. Este es un parámetro que no se puede editar.
Prueba de recibo de impresión (Test Pos Printer)	Botón que le permite realizar una prueba. En la impresión podrá observar el diseño del recibo y así verificar la funcionalidad del servicio de impresión

- Una vez finalizada la configuración puede hacer clic en "Ok" para aceptar todas las configuraciones.

ANEXO 2. FICHA TÉCNICA IMPRESORA POS EPSON TM-T88IV

Epson TM-T88IV Serie



FICHA TÉCNICA



Dedica más tiempo a tus clientes confiando el punto de servicio a esta exclusiva impresora térmica de recibos

Ventajas de la Epson TM-T88IV

La TM-T88IV es una impresora térmica de diseño compacto y de altas prestaciones. Creada con todas las funcionalidades de su predecesora, la TM-T88III, ofrece impresiones rápidas y una gran fiabilidad. Está especialmente indicada para aplicaciones de restauración, supermercados y comercio en general donde una alta fiabilidad e impresiones rápidas y silenciosas son requeridas.

Impresión rápida y versátil

Es ideal para grandes volúmenes gracias a su velocidad máxima de 200 mm/seg (un 25% más que su predecesora). Es también capaz de imprimir fuentes suavizadas y escalables con una selección de distintos tamaños, así como códigos de barras (incluso de dos dimensiones) en vertical y apaisado. Además permite la impresión en dos colores (utilizando papel térmico especial) para mejorar el aspecto de tus recibos.

200 mm/seg. en modo gráfico

Una de las principales mejoras que incorpora este modelo es la impresión en alta velocidad, tanto para texto como en modo gráfico. Ya no es necesario utilizar las fuentes internas de la impresora para imprimir a alta velocidad. Ahora podrás utilizar fuentes de Windows, enviar los códigos de barras y los logos desde su aplicación a través del driver y sin que la velocidad se vea por ello afectada.

Diseño compacto y fácil de usar

Ofrece una carga de papel sencilla y corte de recibos automático. También tiene incorporada una interfaz para cajón monedero. Además el diseño de la cubierta se ha mejorado sustancialmente para evitar la posible entrada accidental de objetos o líquidos.

CARACTERÍSTICAS PRINCIPALES

EPSON®
EXCEED YOUR VISION

ESPECIFICACIONES DEL PRODUCTO

INTERFACES	
Interfaces	RS-232, Apertura de cajón
ENCENDIDO	
Fuente de alimentación	PS-180 (opción)
Consumo de energía	Standby: 0,1 A, Medio: 1,8 A
GENERAL	
Dimensiones del producto	145 x 195 x 148 mm (ancho x profundidad x altura)
Peso	1,8 kg
Color	Epson Dark Gray / Epson Cool White
Nivel de ruido	Operación: 55 dB (A)
FUENTES Y ESTILOS	
Velocidad de impresión	200 mm/s
Tamaño de letra	0,99 mm (ancho) x 2,4 mm (alto) / 1,41 mm (ancho) x 3,39 mm (alto)
Registro de caracteres	95 alfanumérico, 48 internacional, 128 x 11 gráfico
Caracteres por pulgada	20 cpp / 15 cpp
Densidad de puntos	180 ppp x 180 ppp

INFORMACIÓN LOGÍSTICA

Dimensiones embalaje individual	250 x 400 x 274 mm
Peso de la caja	5,9 kg
Cantidad para pedidos múltiples	2 Unidades

Epson TM-T88IV Serie

INCLUYE

- Equipo
- Manual de usuario
- Tapa del botón de conexión
- Cubierta de conector

ACCESORIOS OPCIONALES

- Epson WH-10: Soporte para montaje en pared para TM-T88IV, TM-T88V
- Epson UB-E02 C32C824151
- Epson UB-P02II C32C823891
- Epson UB-R03 C32C824461
- Epson UB-S01 C32C823361
- Epson UB-S09 C32C823861
- Epson UB-U01III C32C824111
- Epson UB-U02III C32C824121
- Epson UB-U03II C32C824131

ConverPlus

Epson Ibérica, S.A.U.
Tel.: 93 582 15 00
Tel.: 902 49 59 69 (Preventa)
Fax: 93 582 15 55

Central: Cerdanyola
Avda. Roma, 18-26
08290 Cerdanyola del Vallès (Barcelona)

www.epson.es

Las marcas comerciales o marcas registradas son propiedad de Seiko Epson Corporation o de sus respectivos propietarios. La información sobre los productos puede estar sujeta a modificación en función de las actualizaciones en sus especificaciones.

EPSON®

ANEXO 3. DIAGRAMA UML

