



**UNIVERSIDAD NACIONAL AUTÓNOMA  
DE MÉXICO**

---

---

**FACULTAD DE INGENIERÍA**

**INFORME DE TRABAJO PROFESIONAL**

**DESARROLLOS PARA LA MEJORA DE LOS  
PROCESOS OPERATIVOS EN LOS RESTAURANTES  
DE PREMIUM RESTAURANT BRANDS**

**QUE PARA OBTENER EL TÍTULO DE:**

**INGENIERO EN COMPUTACIÓN**

**PRESENTA:**

**SERGIO CUÉLLAR VALDÉS**

**2012**



# Índice general

<b>Introducción</b>	<b>1</b>
Objetivos . . . . .	2
<b>1. Perfil</b>	<b>3</b>
1.1. Funciones de la posición . . . . .	3
1.2. Impacto de la posición . . . . .	3
1.3. Relaciones de trabajo . . . . .	4
1.4. Características del perfil . . . . .	4
<b>2. Descripción</b>	<b>5</b>
2.1. Objetivos . . . . .	5
2.2. Descripción del Sistema en restaurantes . . . . .	6
2.2.1. Software Libre . . . . .	6
2.2.2. GNU/Linux . . . . .	8
2.2.3. Apache Tomcat . . . . .	8
2.2.4. Servidor HTTP Apache . . . . .	9
2.2.5. PostgreSQL . . . . .	9
2.2.6. Mozilla Firefox . . . . .	10
2.2.7. Mozilla Thunderbird . . . . .	10
2.2.8. OpenOffice.org . . . . .	10
2.2.9. Sistema Punto de Venta SUS . . . . .	11
2.2.9.1. Toma de ordenes . . . . .	12
2.2.9.2. Control de Efectivo y despacho . . . . .	12
2.2.10. e-Reports . . . . .	12
2.2.11. Reporteador . . . . .	13
2.3. Algunos Proyectos desarrollados . . . . .	17
2.3.1. Metodología de desarrollo de software . . . . .	17
2.3.2. Auditoría Electrónica . . . . .	18
2.3.3. Reporte de Estimado de Usos Ideales en KFC . . . . .	24
2.3.4. Cierre de Lote de Tarjeta de Crédito . . . . .	26
<b>3. Proyecto de Control de Asistencia</b>	<b>31</b>
3.1. Tipos de sensores . . . . .	32
3.2. Ventajas y desventajas . . . . .	33
3.3. Prototipo 1: Lector Microsoft . . . . .	34

<i>ÍNDICE GENERAL</i>	II
3.4. Prototipo 2: Lector ZK Software . . . . .	40
3.4.1. Descripción de scripts . . . . .	42
<b>4. Resultados</b>	<b>45</b>
<b>Conclusiones</b>	<b>48</b>
<b>Apéndice A. Cambios a libfprint</b>	<b>51</b>
<b>Apéndice B. Modificación a tabla pp_emp_check</b>	<b>54</b>
<b>Apéndice C. Procedimiento almacenado para cálculo de horas</b>	<b>56</b>

# Índice de figuras

2.1. Página principal de e-Reports . . . . .	13
2.2. Reporteador . . . . .	14
2.3. Sistema en restaurantes PH . . . . .	15
2.4. Sistema en restaurantes KFC . . . . .	16
2.5. Caja Panasonic 790 . . . . .	16
2.6. Caja Panasonic 5500 . . . . .	17
2.7. AccelePort Xem de Digi con su tarjeta . . . . .	19
2.8. Resultado de una consulta al reporte web de Ticket de Auditoría . . . . .	23
2.9. Usos Ideales Diarios en KFC . . . . .	27
2.10. Cierre de Lote de Tarjeta de Crédito en e-Reports . . . . .	28
2.11. Ventanas de Xdialog de la alarma . . . . .	29
3.1. Huella Digital . . . . .	32
3.2. Microsoft Fingerprint Reader . . . . .	34
3.3. Ejemplo de imagen capturada por el dispositivo . . . . .	35
3.4. Pantalla de entrada/salida en sistema punto de venta. . . . .	37
3.5. Pantalla de entrada/salida con lector de huellas digitales. . . . .	38
3.6. Listado de asociados. . . . .	39
3.7. Lector de huellas digitales U300 de ZKSoftware . . . . .	40
3.8. Página web del dispositivo mostrando usuarios registrados. . . . .	41
3.9. Página web mostrando los registro en un día específico. . . . .	41
3.10. Reporte de asistencia en eReports. . . . .	43
3.11. Forma ideal para colocar el dedo en el lector. . . . .	44
3.12. Forma incorrectas para colocar el dedo en el lector. . . . .	44

# Introducción

La tecnología de la información (TI) es una herramienta valiosa dentro de la estrategia de negocios de Premium Restaurants Brand (PRB). En el departamento de Sistemas proveemos de soluciones tecnológicas a nuestros restaurantes, así como también establecemos estándares tecnológicos y de procesos en nuestras oficinas.

PRB maneja las marcas KFC y Pizza Hut, contando con más de 300 restaurantes a nivel nacional. El sistema operativo que se maneja en los restaurantes es GNU/Linux, teniendo como punto de venta (POS<sup>1</sup>), un desarrollo propietario realizado en su origen por Yum Restaurant Brands de Estados Unidos.

Las Tecnologías de la Información apropiadas, cuando se utilizan junto con los principios de administración de ingresos, puede ayudar a los restaurantes a aumentar éstos y sus ganancias. Las Tecnologías de la Información han revolucionado el panorama de los negocios en el mundo y la industria de los restaurantes no es la excepción. Las TI han modificado a varias industrias y ahora juegan un papel fundamental en las reglas que rigen el mundo de los negocios y en la forma en que se acercan a los clientes. Las ventajas de las TI en cuanto al incremento de la competitividad, reducción de errores y creación de nuevas funcionalidades son incuestionables en cualquier sector, incluido el de los restaurantes.

Los restaurantes han invertido en automatización de funciones relacionadas con la nómina, contabilidad, inventario, etc., con el objetivo de bajar costos. También existen los sistemas punto de venta (POS, Point of Sale), los cuales facilitan sustancialmente la operación del restaurante, ayudando a realizar el marcaje de productos de manera rápida, con lo que se logra atender un mayor número de clientes en menor tiempo, la modificación automatizada de los menús, cambio de precios, etc.

En PRB contamos con un POS creado a la medida, lo cual permite a la empresa realizar desarrollos que implican la modificación del mismo POS, ya que se cuenta con el código fuente de éste. Estas modificaciones implican programación de nuevas funcionalidades o de corrección de errores (bugs). Así mismo es posible realizar

---

<sup>1</sup>Point of Sale

desarrollos que se relacionen con el POS e interactúen con él, para la obtención y manipulación de datos generados por el punto de venta.

Diversas áreas de la empresa, como Operaciones y Finanzas, principalmente, considera los datos de la venta como muy importantes, ya que gracias a ellos pueden saber la tendencia que siguen las ventas, el comportamiento de los nuevos productos y/o promociones, los niveles de inventarios, los días y horas de mayores ventas, etc.

Una de las actividades del área de Desarrollo de la Dirección de Sistemas, se encarga de la programación de reportes web de los datos que ayuden principalmente a los gerentes de restaurante a visualizar los datos de las operaciones de su restaurante de manera casi inmediata.

## Objetivos

Desarrollo e implementación de reportes que ayuden a la operación de restaurantes a eficientar sus procesos, conocer resultados, ahorrar tiempo, identificar áreas de oportunidad. Así como también el desarrollo de aplicaciones que ayuden a la mejora en el sistema en general del punto de venta utilizado y de la operación del restaurante.

La plataforma a utilizar es GNU/Linux utilizando Apache Tomcat, Java, JSP, Perl, C, Bash, y otras herramientas de software libre.

# Capítulo 1

## Perfil

**Puesto Actual:** Programador Analista

**Puesto del Supervisor:** Gerente de Desarrollo de Restaurantes

**Nombre del Supervisor:** Aníbal Avelar

**Propósito de la función:** Atender requerimientos relacionados al desarrollo de sistemas en restaurantes por parte del área de Operaciones y Sistemas.

### 1.1. Funciones de la posición

- Desarrollo de reportes en restaurantes requeridos principalmente por el área de Operaciones.
- Desarrollo de scripts y programas que mejoren la operación del sistema del restaurante.
- Corrección y mejoras del sistema de punto de venta (SUS/FMS).
- Desarrollo de nuevas herramientas que coadyuven a la disminución del costo de operación.

### 1.2. Impacto de la posición

- **Requerimientos técnicos:** GNU/Linux, Java, Perl, bash scripting, C/C++.

- **Conocimientos adquiridos en el puesto:** Funcionamiento de la operación del negocio. Así como también, el uso de otras tecnologías nuevas que puedan ayudar al desarrollo de los sistemas.
- **Toma de decisiones:** El diseño y desarrollo de los reportes solicitados.

### 1.3. Relaciones de trabajo

- **Internos:** Se tiene una relación estrecha con soporte técnico para la resolución de problemas escalados, así como con el área de operaciones para obtener los requerimientos en reportes a desarrollar.
- **Externos:** Se le debe proporcionar soporte técnico y consultoría técnica en los reportes del sistema a las franquicias que así lo soliciten.

### 1.4. Características del perfil

- **Estudios:** Ing. en computación, ciencias de la computación y/o licenciatura en informática.
- **Experiencia mínima requerida:** 3 años
- **Habilidades requeridas:** Conocimientos en programación, en sistemas tipo UNIX, trabajo en equipo, conocimiento del idioma inglés.

# Capítulo 2

## Descripción

El área de Desarrollo de Sistemas de Restaurantes está conformado por cuatro personas encargadas de proveer varios servicios de apoyo a alrededor de 340 restaurantes propios de la franquicia de PRB, así como alrededor de 150 restaurantes de otros franquiciatarios. Los requerimientos de desarrollos nuevos son proporcionados generalmente por el área de Operaciones cuando éstos tienen que ver la operación misma del restaurante, o bien, por la misma área cuando se deben realizar actualizaciones, corrección de bugs, etc., al sistema.

### 2.1. Objetivos

Entre los servicios proporcionados por el área de Desarrollo de Sistemas de Restaurantes se encuentran:

- Desarrollo y mantenimiento de reportes solicitados por el área de Operaciones.
- Mantenimiento al punto de venta (programación de mejoras, nuevas funcionalidades, corrección de bugs).
- Programación de herramientas que coadyuven a la eficaz operación del restaurante.
- Desarrollo de aplicaciones que contribuyan a disminuir el costo de operación.
- Investigación de nueva infraestructura de sistemas en el restaurante (impresoras, tarjetas de vídeo, terminales, etc.)

## 2.2. Descripción del Sistema en restaurantes

Cada uno de los restaurantes cuenta con una computadora, cuyas características pueden variar en restaurantes, pero las mínimas especificaciones técnicas son:

- 1 gigabyte en memoria RAM.
- Procesador a 1.5 GHz.
- Disco duro de 80 gigabytes.
- Monitor a color de 15 pulgadas.
- Puerto AGP<sup>1</sup>.
- 2 puertos PCI<sup>2</sup>.

En la computadora de cada uno de los restaurantes se efectúan diversos procesos como son:

- Registro y procesamiento de la venta.
- Generación de reportes.
- Manejo de periféricos como impresora de tickets, impresora del gerente, cajas registradoras, terminales, etc.
- Actividades gerenciales (correo electrónico, consultas web de la intranet, uso de procesador de textos, hoja de cálculo, etc).

### 2.2.1. Software Libre

Una de las características principales del sistema en restaurantes, es que casi en su totalidad, a excepción del punto de venta que fue desarrollado por la compañía, es software libre.

El software libre es software que puede ser usado, estudiado y modificado sin restricción y el cual puede ser copiado y redistribuido en su forma modificada u original, ya sea sin restricciones o con mínimas restricciones que aseguren que los

---

<sup>1</sup>Accelerated Graphics Port

<sup>2</sup>Peripheral Component Interconnect

siguientes usuarios del software puedan seguir haciendo estas actividades. El software libre es generalmente disponible sin cargo alguno pero puede tener algunas cuotas, por ejemplo para ser distribuido en forma de CDs u otras formas.

En la práctica, para que un software sea distribuido como software libre, el código fuente de éste debe estar disponible para el usuario, así como también indicarle en una nota que se le conceden los derechos arriba mencionados. Esta nota puede ser, ya sea, una licencia de software libre o indicando que el código fuente está disponible al dominio público.

Las ventajas del software libre son:

- Bajo costo, lo que implica el ahorro en pago de licencias.
- Es posible adaptar el software a las necesidades que tenga cada usuario, teniendo como resultado un software personalizado.
- Al ser público el código fuente, permite que programadores hagan correcciones a errores y mejoren el software de manera rápida.
- De acuerdo a los dos últimos puntos, el software libre no depende de una única empresa u organización, lo que evita que se impongan condiciones en su uso, como ocurre con el software privado.
- La innovación tecnológica que surge gracias a que cada usuario puede hacer aportaciones a la mejora del software.

El sistema de software de los restaurantes se compone principalmente de:

- Sistema Operativo GNU/Linux kernel 2.6, basado en Knoppix con escritorio Xfce.
- Apache Tomcat
- Servidor HTTP Apache
- PostgreSQL
- Mozilla Firefox
- Mozilla Thunderbird
- OpenOffice.org
- Herramientas GNU (bash, ksh, awk, perl, etc.)
- Sistema punto de venta SUS/FMS (propietario, desarrollado por la misma compañía).

### 2.2.2. GNU/Linux

GNU/Linux se refiere al sistema operativo que utiliza el kernel Linux y las herramientas GNU. El kernel Linux puede ser instalado en una gran variedad de hardware, desde teléfonos móviles, computadoras tipo tableta, consolas de videojuegos, hasta mainframes y supercomputadoras. El kernel de Linux fue iniciado en 1991 por el finlandés Linus Torvalds. Las herramientas del sistema y librerías conocidas como GNU, fueron en un principio desarrolladas por Richard Stallman en 1983.

Algunas características del kernel de Linux son:

- Es multitareas, permite ejecutar diferentes programas al mismo tiempo.
- Es multiusuario, distintos usuarios pueden estar conectados a la misma computadora al mismo tiempo.
- Es multiplataforma, puede ser instalado en distintos tipo de arquitectura de procesadores.
- Es multihilo, Linux tiene soporte en kernel nativo para el control de múltiples hilos independientes.
- Tiene soporte a una gran cantidad de sistemas de archivos (ext2, ext3, ext4, ReiserFS, XFS, JFS, FAT32, etc.).

El kernel utilizado en los restaurantes no ha recibido ninguna modificación por parte del equipo de desarrollo, únicamente se han cambiado los parámetros por default, como mensajes de depuración de puertos USB.

### 2.2.3. Apache Tomcat

Apache Tomcat o simplemente Tomcat es un contenedor de servlets desarrollado por la Apache Software Foundation. Tomcat implementa las especificaciones de Java Servlet y JavaServer Pages (JSP) de Sun Microsystems (ahora Oracle).

Un servlet es un tipo de clase de Java que es usada para extender las capacidades de un servidor que contienen aplicaciones que son accedidas mediante un modelo de programación petición-respuesta. Si bien, un servlet puede responder cualquier tipo de petición, son comúnmente usados en aplicaciones contenidas en un servidor web.

Los JavaServer Pages, mejor conocidos como JSPs es una tecnología de Java que ayuda a los desarrolladores de software a generar páginas web de manera dinámica basadas en HTML, XML o algún otro tipo de documento.

### 2.2.4. Servidor HTTP Apache

El servidor HTTP Apache, o mejor conocido únicamente como Apache, es un servidor web que implementa el protocolo HTTP/1.1. Una de las mayores ventajas de Apache es que puede ser mejorado con la ayuda de módulos compilados que extienden la funcionalidad del servidor. Hay módulos que permiten la interacción con distintos lenguajes de programación como Perl, Python, PHP, etc. Existen módulos de autenticación para aumentar la seguridad del servidor, como `mod_auth`, `mod_access`, etc. Una de las características más importantes de Apache es el manejo de host virtuales, lo que permite que un servidor Apache maneje diferentes sitios web.

Tomcat y Apache pueden ser conectados a través del conector `mod_jk`. Eso es de gran ayuda, por ejemplo, cuando se tienen páginas programadas en PHP y en JSP, y se quiere tener un único puerto de acceso a la página principal. Con `mod_jk`, Apache recibe todas las peticiones y les da respuesta únicamente las de tipo PHP y HTML, y redirige a Tomcat las que tienen que ver con JSPs. Esta implementación la realicé en PRB para evitar el uso del puerto 80 para reportes en PHP y el 8080 para reportes programados en JSP. Se unificó el URL que es ingresado en los restaurantes.

### 2.2.5. PostgreSQL

PostgreSQL, o simplemente Postgres, es un sistema de gestión de base de datos relacional orientada a objetos (ORDBMS<sup>3</sup>). Algunas de sus características son:

- Uso de lenguajes procedurales o mejor conocidos como *store procedures*, permiten que bloques de código sean ejecutados por el servidor de base de datos y pueden ser escritos en lenguajes de programación distintos a SQL y C; y sirven para crear funciones definidas por el usuario (subrutinas, triggers, etc.). Pueden ser programados en Perl, Python, pgSQL, Tcl, principalmente.
- Uso de índices, triggers, transacciones anidadas, vistas.
- Numerosos tipos de datos y posibilidad de definir nuevos.
- Posee interfaces de programación de aplicaciones (API<sup>4</sup> en varios lenguajes como C, C++, Java, Perl, Ruby, PHP, entre otros.
- Es 100 % ACID<sup>5</sup>

---

<sup>3</sup>object-relational database management system

<sup>4</sup>Application Programming Interface

<sup>5</sup>*Atomicity, Consistency, Isolation, Durability*. En base de datos se denomina ACID a un conjunto de características necesarias para que una serie de instrucciones puedan ser consideradas como una transacción.

### 2.2.6. Mozilla Firefox

Mozilla Firefox es un navegador web disponible para varios sistemas operativos como Microsoft Windows, GNU/Linux, Mac OS X, FreeBSD y muchos otros. Para la visualización de páginas web, Firefox utiliza a Gecko, que es un motor de renderizado, que implementa la mayoría de los estándares web. Algunas de las características de Firefox son:

- Navegación por pestañas.
- Corrector ortográfico.
- Navegación privada.
- Soporte de complementos (plug-ins) desarrollados por terceros para incrementar la funcionalidad del navegador.
- Administrador de descargas.

### 2.2.7. Mozilla Thunderbird

Mozilla Thunderbird es un cliente de correo electrónico y noticias. Puede manejar múltiples cuentas de correo electrónico y noticias. Características como son la búsqueda rápida, filtrado de mensajes, agrupamiento de mensajes y etiquetas ayudan a manejar y encontrar mensajes de una manera fácil. Thunderbird incorpora un filtro de spam tipo Bayesiano para la clasificación de correo no deseado. Al igual que Firefox, a Thunderbird le pueden ser instalados complementos o extensiones que amplían la funcionalidad de éste. Soporta POP e IMAP, así como LDAP para el manejo de la libreta de direcciones. Está disponible para varios sistemas operativos como Microsoft Windows, GNU/Linux, Mac OS X, Opensolaris, etc.

### 2.2.8. OpenOffice.org

OpenOffice.org es una suite de aplicaciones cuyos componentes principales son procesador de palabras, hoja de cálculo, presentaciones, gráficas y base de datos. Está disponible para distintos sistemas operativos. El formato nativo de OpenOffice.org es el estándar ISO/IEC OpenDocument (ODF), pero también soporta la lectura y en la mayoría, la escritura de formatos propietarios como los de WordPerfect, StarOffice, MS Works, Rich Text format, los formatos de Microsoft Office, entre otros.

Posee diccionarios ortográficos en varios idiomas. Puede usar extensiones para agregar funciones adicionales. Las aplicaciones incluidas con OpenOffice.org son:

- OpenOffice.org Writer es el procesador de textos. Una de las características de Writer es que permite exportar documentos a PDF y HTML sin software adicional.
- OpenOffice.org Calc es una hoja de cálculo similar a Microsoft Excel, pero con más características.
- OpenOffice.org Impress es el programa usado para realizar presentaciones similares a Microsoft PowerPoint. Puede exportar presentaciones al formato SWF, permitiendo visualizar la presentación en cualquier computadora que cuente con un reproductor Flash.
- OpenOffice.org Base es un programa similar a Microsoft Access, el cual permite la creación y manejo de base de datos, elaboración de formularios y reportes.
- OpenOffice.org Draw es un editor de gráficos vectoriales y herramienta para elaborar diagramas, similar a Microsoft Visio.
- OpenOffice.org Math es una aplicación diseñada para la creación y edición de fórmulas matemáticas. Las fórmulas pueden ser incorporadas a otros documentos de OpenOffice.org, como en un documento de Writer o bien exportar la fórmula a otro formato de archivo como PDF.

### 2.2.9. Sistema Punto de Venta SUS

El sistema punto de venta utilizado tanto en PH como en KFC se llama SUS, el cual fue desarrollado por el área de desarrollo de Yum Restaurants de Estados Unidos. Nosotros contamos con el código fuente, con el cual podemos corregir errores, añadir funcionalidades o mejorar las ya existentes. El punto de venta no es software libre. El cambio a la configuración del punto de venta ocurre únicamente para realizar modificaciones a la seguridad (cambio de permisos para visualizar algunas opciones).

SUS es capaz de realizar toma de ordenes, el despacho de estas ordenes, retiros, funciones de control y actividades administrativas propias de un restaurante.

El sistema SUS divide las actividades realizadas en el restaurante en las siguientes áreas:

- Toma de ordenes.

- Control de efectivo.
- Funciones administrativas.
- Inventarios.
- Control de asistencia.
- Mantenimiento de tablas del sistema.
- Reportes.

Estas áreas corresponden también a las áreas en las que se les puede asignar responsabilidad al equipo del restaurante. Cada nivel de responsabilidad, junto con el asociado cuenta con un nombre de usuario y contraseña. Esto permite la asignación de responsabilidades, limitando el acceso a partes del sistema con información importante de los clientes y del restaurante.

#### **2.2.9.1. Toma de ordenes**

Esta función se utiliza principalmente en Pizza Hut, y en KFC en los restaurantes con servicio de entrega a domicilio, en donde es de suma importancia, que cada producto sea rápida y fácilmente ordenado para toda ocasión, de esta manera se agiliza enormemente el proceso de toma de ordenes.

#### **2.2.9.2. Control de Efectivo y despacho**

Las funciones de esta parte del punto de venta permiten realizar retiros de dinero de manera eficiente

#### **2.2.10. e-Reports**

*e-Reports* es una herramienta desarrollada por el área de Desarrollo a Restaurantes para consultar reportes via web, mediante Firefox, utilizando el servidor Tomcat junto con la base de datos de PostgreSQL. Cada restaurante cuenta con este reporteador para realizar distintas actividades, como son, la consulta de su inventario, la captura de gastos semivARIABLES, conocer el número de transacciones mediante gráficas, etc. Es una de las herramientas más usadas en los restaurantes, junto con el punto de venta, debido a la gran cantidad de información que se puede consultar.

Al ser una herramienta web, ésta puede ser consultada desde cualquier computadora que pertenezca a la VPN y conozca la clave para ingresar.

Los reportes son programados en JSP, JavaScript y en algunos casos se hace uso de scripts externos desarrollados en Perl y Bash principalmente. Además, se cuenta con una API desarrollada especialmente para e-Reports, en la cual ya existen métodos para la conexión a base de datos, la creación de tablas dinámicas, conversión de resultados de consulta de base de datos a arreglos de JavaScript, etc. El framework de JavaScript utilizado es BlueShoes<sup>6</sup>.



Figura 2.1: Página principal de e-Reports

### 2.2.11. Reporteador

Existe otra herramienta desarrollada por el área de Desarrollo a Restaurantes para consultar reportes vía web llamado *Reporteador*, el cual, a diferencia de *e-Reports*, es usado en mayor parte de los reportes que posee, para desplegar información relacionada directamente con reportes generados por el mismo punto de venta. El despliegue de estos reportes es programado en HTML, PHP y algunos CGI's<sup>7</sup> en Python. Se puede decir que *e-Reports* y el *Reporteador* se complementan, pero en un futuro, el plan es migrar todos los reportes de este último, para concentrar todo en *e-Reports*.

El *Reporteador* posee una clasificación de reportes, la cual está conformada por:

<sup>6</sup><http://www.blueshoes.org/en/javascript/>

<sup>7</sup>Common Gateway Interface

- Ingresos y gastos: Ventas Diarias, Ventas por Hora, Operaciones Diarias, Cancelaciones Diarias, Auditoría de Cajero y Ventas pollo por hora (para KFC).
- Inventarios: Inventario de críticos, usos ideales, Ordenes de compra, Recepciones.
- Mano de Obra: Modelo de Labor.
- Estadísticas: Pronóstico y Ensamble, Historial de Pedidos, Mix Diario por Productos, Mix por Cajero, Estadística Diaria.
- Home Service: Auditoría de Repartidor, Reporte de Vendedores, Mapa por calles, Mapa de Calles, etc.
- Planeación: Ya no cuenta con algún reporte.
- Auditoría: Notas de consumo, Retiros Parciales, Depósitos de Dólares.
- Utillerías: Ya no cuenta con algún reporte.

El *Reporteador* y *e-Reports* conviven en la misma dirección y mismo puerto gracias al mod\_jk, descrito en 2.2.4



Figura 2.2: Reporteador

A continuación se muestran un diagrama que representa a muy grandes rasgos el sistema en Pizza Hut:

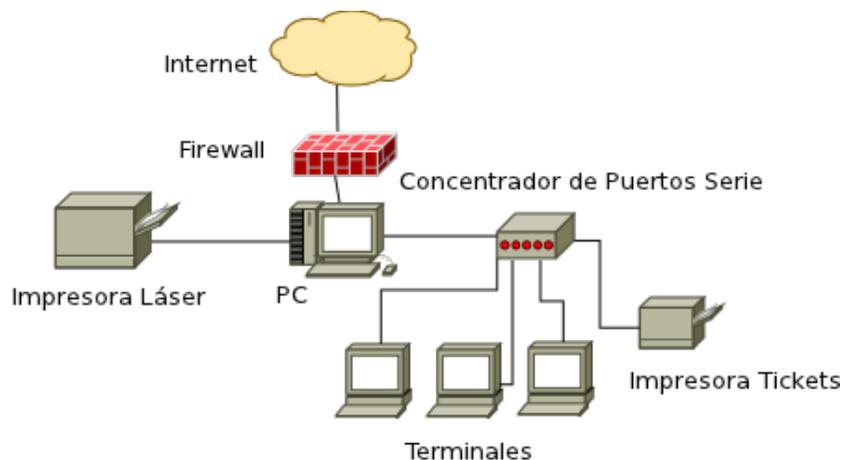


Figura 2.3: Sistema en restaurantes PH

El sistema se compone de una computadora en la cual se ejecuta el sistema punto de venta, se generan reportes propios del punto de venta, el gerente del restaurante consulta el correo electrónico, consulta reportes web propios del restaurante y de la intranet y realiza actividades propias de la operación (como cierres de inventarios, altas/bajas de empleados, etc.). Posee una impresora láser para imprimir documentos, reportes, etc.

La computadora tiene una conexión a internet protegida por un firewall y comunicación a la VPN<sup>8</sup> de la empresa.

Debido a la gran cantidad de equipo que se le puede conectar a la computadora, entre terminales, cajas registradoras e impresoras de tickets, principalmente, se hace uso de un concentrador de puertos seriales, a los cuales se conectan todos los dispositivos. Estos concentradores pueden ser de 6 o 12 puertos y pueden ser conectados más de uno, en caso necesario. La impresora es de matriz y se manda la impresión de los tickets una vez terminada de capturar la orden.

Las terminales, en caso de PH, sirven para tener el punto de venta, que se ejecuta en la computadora, en cada una de ellas, y así poder tomar varias ordenes al mismo tiempo.

La siguiente figura muestra el sistema en KFC a grandes rasgos:

A diferencia del sistema en PH, se cuentan con cajas registradoras en lugar de terminales. Estas cajas registradoras son de la marca Panasonic, y se tienen dos modelos: touchscreen (Panasonic 790, que cuenta con sistema operativo Windows XP Embedded) para restaurantes con un gran número de transacciones y de teclado

<sup>8</sup>Virtual Private Network, red privada virtual

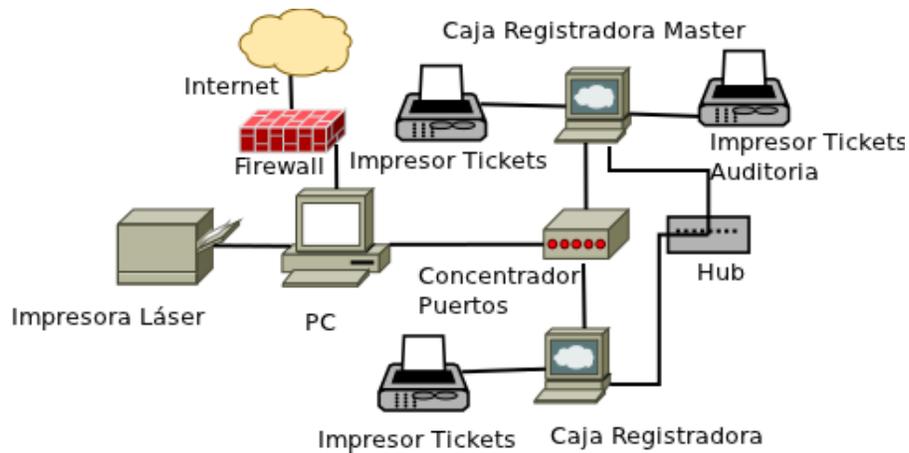


Figura 2.4: Sistema en restaurantes KFC

(Panasonic 5500, usando su propio firmware) para las demás. Se cuenta con un programa desarrollado en la empresa que convierte la salida de las cajas registradoras en registros de venta que son ingresados al punto de venta. Cada caja registradora cuenta con su propia impresora de tickets. Dentro del grupo de cajas registradoras se debe tener una especial llamada *master*, la cual es la encargada de transmitir programación a las secundarias, así como también de recibir información de venta de las demás cajas para ser impresas en el impresor de auditoría. A partir de un desarrollo realizado por mi, esta auditoría se dejó de imprimir y ahora es almacenada de manera electrónica.



Figura 2.5: Caja Panasonic 790



Figura 2.6: Caja Panasonic 5500

## 2.3. Algunos Proyectos desarrollados

### 2.3.1. Metodología de desarrollo de software

En el área de Desarrollo a restaurantes no se cuenta con una metodología específica para la elaboración de proyectos. Los pasos a seguir para el desarrollo de un nuevo sistema son:

- Reunión con el cliente para saber sus necesidades.
- Análisis de los requisitos del sistema.
- Diseño del sistema.
- Codificación del sistema.
- Pruebas en un ambiente especial (ambiente de pruebas).
- Pruebas en un ambiente de producción (restaurante).
- Retroalimentación por parte del cliente.
- En caso necesario, modificación y/o corrección del sistema a petición del cliente.
- Siguiendo fase de pruebas, donde se libera a un grupo pequeño, 5 o 10 restaurantes en un periodo de una semana; posteriormente se libera a otro grupo de restaurantes, sino hay algún problema, se continua con el siguiente paso.

- Liberación a nivel nacional.
- Mantenimiento.

En algunas ocasiones, la necesidad del cliente para la rápida liberación de un sistema y con previa autorización del director de Sistemas, éste se puede liberar sin cubrir las etapas de prueba.

A continuación se presentan algunas actividades desarrolladas por mi en la empresa.

### 2.3.2. Auditoría Electrónica

La operación del negocio debe ser vigilada por algún medio, en el cual se plasme un historial de ventas de cada caja registradora en KFC. Es por esto que se cuenta con un impresor de tickets llamado de auditoría, el cual imprime el resumen de venta de cada ticket de cada caja.

Una de las necesidades de negocio más importantes es la de reducción en el costo de venta. Por lo que se tomó la decisión de eliminar este impresor y almacenar la auditoría de manera electrónica. Con esto se ahorra los impresores de tickets, los gastos en mantenimiento de éstos y los rollos de papel en los más de 230 restaurantes KFC.

Ambos modelos de cajas cuentan con puertos seriales para conectar impresores, además de poseer con un puerto Ethernet RJ-45. Cada caja registradora cuenta con una dirección IP privada para la comunicación con la caja *master*, la cual se encarga, a través de este medio, de transmitirles programación y de recibir los tickets de las demás cajas, para formar la auditoría.

Por lo que se puede observar de esta descripción, es que hay dos maneras posibles de capturar la auditoría de la caja master: por el puerto serial que manda los datos al impresor, o mediante la captura de los paquetes de red que le llegan a la caja master provenientes de las demás cajas. Se optó por usar la salida del puerto serial hacia al impresor, ya que al mandar a imprimir se manda el ticket integro junto con los caracteres de control para la impresora. De la otra manera, se hubiese tenido que estudiar el protocolo de envío de información de las cajas a la caja master; lo que se sabe de manera certera, es que la comunicación es tipo UDP<sup>9</sup>.

Al estar configurada la caja master para manejar la auditoría, se debe especificar el puerto serie por el que se mandará a imprimir, en caso de no tener conectado

---

<sup>9</sup>User Datagram Protocol

Pin RJ-45	Señal	Dirección	Señal	Pin DB-9
3	GND (Tierra)	<->	GND (Tierra)	Carcaza
4	TxD (Transmit Data)	->	RxD (Receive Data)	2
5	RxD (Receive Data)	<-	TxD (Transmit Data)	3
6	SG (Signal Ground)	<->	SG (Signal Ground)	5
7	CTS (Clear to Send)	<-	DTR (Data Terminal Ready)	4
1	DCD (Data Carrier Detect)	<-	RTS (Request to Send)	7
2	RTS (Request to Send)	->	CTS (Clear to Send)	8
8	DTR (Data Terminal Ready)	->	DSR (Data Set Ready)	6

Tabla 2.1: Configuración RJ-45 a DB-9 para DIGI

un impresor, la caja marcará un error en su pantalla. Eso significa que la caja se encuentra continuamente solicitando un status a la impresora.

El concentrador de puertos seriales es el modelo AccelePort Xem de la marca Digi, el cual posee 8 puertos tipo RJ-45, con posibilidad de aumentar el número de módulos para poder soportar más dispositivos.



Figura 2.7: AccelePort Xem de Digi con su tarjeta

La configuración que indica el fabricante Digi para realizar un cable para impresora RJ-45 a DB-9 es la que se muestra en la tabla 2.1.

Para poder saber qué datos manda la caja hice uso de un cable con el que se pueda leer la comunicación de la caja registradora (DB-9) al impresor de auditoría (DB-25). Por lo que abrí el cable e inserté las líneas descritas en la configuración anterior para un RJ-45, de esta manera se cuenta con un cable capaz de leer los datos mandados por la caja registradora. Para la lectura de estos datos utilice el software *minicom*. Este es un programa de software libre utilizado para monitorear el puerto serie, análogo al *Hyperterminal* de Windows. Los resultados indicaron que la caja manda el ticket con únicamente la información necesaria como número de productos, el producto, el costo del producto, el subtotal, el total, fecha, hora, número de ticket y el cajero que lo tomó. Además de estos datos, se cuentan con caracteres de control especiales para el impresor.

Otra característica que observé fue que la caja constantemente se comunica con el impresor para saber si está conectado y este último le manda una respuesta. Por lo que el programa desarrollado en C tendría que: limpiar los caracteres de control, y emular la comunicación que establece la caja registradora para evitar que marque error.

El cable original que va de la caja (DB-9) al impresor (DB-25) tiene la configuración mostrada en la tabla 2.2

Pin DB-9	Señal DB-9	Pin DB-25	Señal DB-25
1	DCD	4 y 5	RTS y CTS
2	RxD	2	TxD
3	TxD	3	RxD
4	DTR	6 y 22	DSR y Ring indicator
5	SG	7	SG
6 y 9	Ring indicator y DSR	20	DTR
7 y 8	CTS y RTS	8	DCT

Tabla 2.2: Configuración original caja a impresor

De acuerdo a la configuración anterior y la indicada por el fabricante para el cable DB-9 a RJ-45, realicé la configuración de cable mostrada en la tabla 2.3.

RJ-45	DB-9
1	7 y 8
2	4
3	carcaza
4	2
5	3
6	5
7	4
8	6

Tabla 2.3: Nueva configuración de cable

Desarrollé en lenguaje C el programa que lee los datos de la caja y además emula la respuesta del impresor a la caja. El programa *portAudit* recibe como argumento el puerto serie del concentrador en el cual está conectado el cable. El programa *portAudit* recibe como variable de ambiente el tipo de caja: `PORT_TIPO_CAJA=790` y `export PORT_TIPO_CAJA=5500` ya que los caracteres de control son distintos para cada caja registradora. Creé un script encargado de establecer estas variables, la configuración del puerto serial (velocidad, paridad, etc.) y la ejecución del binario.

El programa detecta la caja que originó el ticket, y de acuerdo a esta información la almacena en un archivo de texto para cada caja, además de un archivo que

concentra la información de todas las cajas.

De acuerdo a la versión del sistema operativo en restaurantes, añadí una entrada en el archivo `/etc/inittab` para poder controlar de manera automática el inicio del servicio de almacenamiento de auditoría de manera automática de acuerdo al nivel de ejecución del sistema.

A continuación se muestran dos tickets de ejemplo de auditoría:

```

#335                LLEV
4 MIXTA
1 PQ8BI SE          125.00
1 PAFADO            35.00
1 NINGUNO           .01
      VTPR          160.00
      TOTL          160.00
      EFVO          200.00
      CAMB          40.00
NACHO
8377 10:51 #12 MAR.01'11  REG0003

```

```

#336                COME
1 MBBIG C           72.00
1 +PUR/RE           7.00
1 NINGUNO           .01
1 MANZANA
      TOTL          79.00
      EFVO          100.00
      CAMB          21.00
NACHO
8378 11:20 #12 MAR.01'11  REG0003

```

Los tickets número 335 y 336 indican que provienen de la caja 3. El #335 es un pedido para llevar y el #336 es para comer en el restaurante. Se muestran las cantidades de los productos, su precio, el total, el efectivo con que se pagó y el cambio, el número de ticket único, así como la fecha y hora. El ticket #335 que se manda a imprimir para dárselo al cliente se muestra a continuación:

```

PREMIUM RESTAURANT BRANDS SDE RL DE CV
PASEO DE TAMARINDOS 400 PISO1
BOSQUES DE LAS LOMAS, CUAJIMALPA
MEXICO, D.F. 05120

```

RFC PRB100802H20

```

-----
#335      LLEV
4 MIXTA
1 PQ8BI SE      125.00
1 PAFPADO      35.00
1 NINGUNO      0.00
-----
TOTAL A PAGAR      160.00
Ciento Sesenta Pesos 00/100 M.N.
EFVO      200.00
CAMB      40.00

```

Sucursal KFC Div. del Norte  
 AV. DIVISION DEL NORTE 2855  
 PARQUE SAN ANDRES C.P.04040  
 MEXICO, DISTRITO FEDERAL

NACHO

8377 10:51 #12 MAR.01'11 REG0003  
 No. ticket unico: 6  
 (solo se factura el mismo dia)  
 Gracias por su compra

Para la verificación del correcto funcionamiento del `portAudit` creé un script que valida que todas las cajas estén registrando auditoría, y que no falte ningún ticket. Esto se realiza verificando primero, la existencia de la auditoría de las cajas en donde se haya realizado venta. Después se busca el número de ticket en los archivos generados por la interfase que convierte la información de la caja registradora al punto de venta. Se revisa que en la auditoría correspondiente a la caja se encuentre el ticket. En caso de que no esté en ejecución el `portAudit`, que no se estén generando los archivos de auditoría o que falten tickets de auditoría se genera un correo electrónico de manera automática para avisar de la situación correspondiente. Este script se ejecuta mediante CRON cada veinte minutos.

Ejemplo de correo electrónico cuando falta el archivo de auditoría de alguna caja:

```

Received: from xxxxxxxx.xxx.com.mx (192.168.101.xxx) by xxxxxxxx.xxx.com
(xxx.xxx.xxx.xxx) with Microsoft SMTP Server id 14.0.639.21; Mon, 13 Jun 2011
15:00:31 -0500
Received: from xxxxx.xxx.com.mx (unknown [xx.xxx.xx.xxx])      by
linuxeon.prb.com.mx (Postfix) with ESMTD id 07CCC2531;      Mon, 13 Jun 2011
15:00:29 -0500 (CDT)
From: <mx0628r@xxx.com.mx>
To: <sergio.cuellar@xxx.com.mx>, <anibal.avelar@xxx.com.mx>,
<jesus.noriega@xxx.com.mx>, <juanjose.martinez@xxx.com.mx>,
<yaneth.yanez@xxx.com.mx>, <hugo.jacobo@xxx.com.mx>
Message-ID: <10776760.0.1307977209538.JavaMail.root@S062801>
Subject: Port de Auditoria CC 0628 13-06-2011-15:00: FALTA ARCHIVO DE ALGUNA
CAJA
Content-Type: text/plain; charset="ISO-8859-1"
Content-Transfer-Encoding: 7bit
Date: Mon, 13 Jun 2011 15:00:29 -0500
Return-Path: mx0628r@xxx.com.mx
X-MS-Exchange-Organization-AuthSource: mexxch02.xxx.com

```

X-MS-Exchange-Organization-AuthAs: Anonymous  
 MIME-Version: 1.0

Falta(n) archivo(s) de auditoria de caja(s):  
 04

- \* Revisar banderas de cajas
- \* Revisar conexion de cables
- \* Revisar concentrador

El gerente del restaurante puede revisar los tickets de auditoría y del punto de venta mediante un reporte tipo web, en donde le indica la fecha que requiere consultar y como opción tiene la de ingresar el número de ticket a buscar, en caso de que no ingrese este dato, se despliegan todos los tickets del día. Esto se logra gracias a que los tickets de auditoría son almacenados en un directorio específico. El reporte consulta el archivo de acuerdo a la fecha seleccionada. En la figura 2.8 se muestra la salida del reporte.

Reporte Auditoria Ticket			
#0212		LLEV	
	ETELEINA MGR		
1	P.SUPRM		63.00
1	COLA		
7	PQT.BIG		441.00
-----			
	SETL		504.00
TOTL		504.00	
	EFVO		550.00
	CAME		46.00
MARTA	CAJE		
2091	10:43	JUN.07'11	FNTK02 P1
#0213		LLEV	
1	POSFU CR		110.00
-----			
	SETL		110.00
TOTL		110.00	
	EFVO		220.00
	CAME		110.00
MARTA	CAJE		
2092	10:47	JUN.07'11	FNTK02 P1
#0214		COME	
	ETELEINA MGR		
2	TRIO HC		114.00
2	COLA		
-----			
	SETL		114.00
TOTL		114.00	
	EFVO		115.00
	CAME		1.00

Figura 2.8: Resultado de una consulta al reporte web de Ticket de Auditoría

Con esta solución se ha tenido un gran ahorro económico al evitar el consumo de papel, cintas de tinta, impresores y su mantenimiento. Igualmente se tiene un impacto benéfico a la ecología, ya que se deja de utilizar una cantidad importante de papel, teniendo una auditoría *paperless*.

### 2.3.3. Reporte de Estimado de Usos Ideales en KFC

En KFC (y Pizza Hut) se necesita saber con anticipación el número de transacciones que se tendrán para cada día, de esta manera se puede saber, mediante un pronóstico, el número de hamburguesas que se deben preparar, el número de piezas de pollo que se deben marinar para posteriormente cocinarlas, el número de purés que se deben tener listos, etc. De esta manera la operación del restaurante está preparada para el nivel de transacciones que se tengan durante el día. Sabiendo esto, el gerente sabe el nivel de inventario que debe tener para satisfacer las transacciones, el número de piezas de pollo que debe descongelar, el número de asociados (empleados) que debe tener a cierta hora del día para poder preparar los productos, etc.

El uso ideal es la cantidad de producto que el restaurante tuvo que haber usado, de acuerdo a su número de transacciones, y a la cantidad de producto establecida por la receta, sin incluir la merma. La merma son los productos, ingredientes o artículos que deben ser descartados debido a: deterioro, han expirado o se produjeron en exceso. La diferencia entre el uso ideal y el uso actual se le conoce como varianza, y puede ser expresada en cantidad de productos o en su valor monetario. Si la varianza es negativa significa que un producto o ingrediente fue utilizado más que la cantidad ideal; si es positiva, un producto o ingrediente fue utilizado menos que lo que dice el ideal.

El uso ideal se conoce una vez que han cerrado las ventas para el día y se generan los reportes del punto de venta. Para el gerente de restaurante es importante saber cuántos vasos, tapas, panes, kilos de papa debe tener listos para la operación de un día específico. Para esto debe conocer el pronóstico en el número de transacciones que tendrá para un día específico. No es lo mismo un domingo que un martes, ni un 16 de diciembre, que un 24 de diciembre. Una vez conocido el número de transacciones pronosticadas, el gerente, debe investigar cuál día tuvo ese número de transacciones, pero ahora reales. Una vez hecho esto debe consultar un reporte de esa fecha generado por el punto de venta donde se encuentran los usos ideales y obtener de el reporte el uso ideal de los productos que necesite.

Esto ocasiona que el gerente ocupe mucho tiempo en la búsqueda de estos usos ideales, por lo que desarrollé un reporte que automatice esta actividad.

Para facilitar la elaboración de este reporte, creé una tabla en la base de datos de PostgreSQL que almacene los usos ideales por fecha y por identificador de producto. Estos datos se obtiene de un comando que ya existe en el sistema y mediante un script en Perl se leen secuencialmente los resultados y se ingresan a la base de datos. Esta carga se realiza de manera diaria, una vez que se han generado los usos ideales del día anterior.

Esta es la descripción de la tabla de PostgreSQL encargada de almacenar los usos

ideales:

```

Table "public.op_inv_ideal_use"
Column |      Type      | Modifiers
-----+-----+-----
turn_date | date          | not null
turn_id   | smallint      | not null
inv_id    | character(6)  | not null
ideal_use | numeric(12,2) |
unit_cost | numeric(12,2) |
misc     | boolean       |
Indexes:
"op_inv_ideal_use_pkey" PRIMARY KEY, btree (turn_date, turn_id, inv_id)
Foreign-key constraints:
"fk1" FOREIGN KEY (inv_id) REFERENCES op_gr1_cat_inventory(inv_id) ON UPDATE RESTRICT ON DELETE RESTRICT

```

La llave foránea `inv_id` pertenece a la tabla `op_gr1_cat_inventory`, la cual contiene, entre otros datos, el nombre del producto y la unidad de medida.

En los restaurantes existe una aplicación llamada *e-Reports12*, la cual está desarrollada en JSP<sup>10</sup>. En esta aplicación se añaden los reportes requeridos clasificándolos de acuerdo a su finalidad. El reporte de “Usos Ideales” se encuentra en el menú de *Mano de Obra* → *Planeación* → *Usos Ideales Diarios*.

Creé archivos de configuración de productos clasificados de acuerdo a la línea de producción donde son utilizados. Los archivos son:

- Biscuits.conf
- Ensalada.conf
- Marinado.conf
- Pure.conf
- Sandwich.conf
- Trasempaque.conf
- Congelados.conf
- Home\_Delivery.conf
- Pure\_Ensalada.conf
- Servicio.conf

En ellos se encuentra el número de inventario del producto a desplegar en el reporte. A continuación se muestra como ejemplo una parte del archivo de `Servicio.conf`:

---

<sup>10</sup>Java Server Pages

```
#####
### Servicio ###
#####
# Caja regular
10519
# Tapa vaso 12 oz
10738
# Caja Economica
10517
# Servilletas
10570
# Tapa 32 oz
10830
# Bucket
10946
# caja chicky
10593
# Char porta vaso
10613
# Catsup sobre
10559
```

Teniendo estos archivos de configuración resulta muy fácil añadir o quitar productos de acuerdo a las necesidades de la operación.

El reporte obtiene de la tabla `op_gt_real_sist_mng` el pronóstico de número de transacciones y las transacciones reales más cercanas a la fecha seleccionada.

En la figura 2.9 se muestra una pantalla de resultados del reporte de usos diarios en KFC:

### 2.3.4. Cierre de Lote de Tarjeta de Crédito

El uso de tarjetas de crédito y débito es indispensable para cualquier negocio ya que al no aceptarlas se pueden perder valiosas transacciones. Las tarjetas, hoy en día, no sólo son un método o forma de pago, sino que da seguridad a quienes la aportan, tanto seguridad al no portar efectivo como seguridad de poder pagar a largo plazo en caso de que no cuenten con el dinero, además de poder tener diferentes beneficios con sus puntos que otorga el banco al realizar compras, como viajes, artículos entre otros.

Actualmente la compañía ya cuenta con el servicio de cobro de tarjetas en la marca PH y se encuentra en un mercado prueba para la marca KFC, ambos obtuvieron excelentes resultados, tanto en transacciones como en un aumento en el ticket

Premium Restaurant Brands

**Uso Ideal Diario**

Fecha: Monday, June 13, 2011, 19:27:45 PM  
 Centro de contribución: 222 - DIVISION DEL NORTE

Año: 2011, Periodo: 7, Semana: 26  
 Número de Transacciones Pronósticadas 213 para el día 2011-05-24  
 Fecha con el número de transacciones reales más parecidas: 2011-06-08

Biscuit		
Producto	Unidad Medida	Uso Ideal
MANTEQ.LIQUIDA	GALN	0.16
PAPEL P/BISCUIT	KILO	0.04
PREM.BISCU AWO	BLSA	4.43

Congelados		
Producto	Unidad Medida	Uso Ideal
COR.SUPREME	PZA	13.00
MEDALL.PECH.KFC	PZA	4.00
PAPA FRANCESA	KILO	29.93
POPCORN CHICKEN	BLSA	1.17
TIRAS EMPANIZAD	PZA	198.00

Ensalada		
Producto	Unidad Medida	Uso Ideal

Figura 2.9: Usos Ideales Diarios en KFC

promedio por lo que se ha tomado la decisión de realizar el lanzamiento nacional por fases hasta cubrir el 100 % de los restaurantes.

Existe una actividad muy importante que se debe realizar siempre al final del día, una vez que han concluido las ventas, el cierre de lote en las terminales bancarias. Con el cierre de lote se asegura que el banco depositará el dinero correspondiente a las ventas pagadas mediante tarjeta de crédito/débito, al siguiente día hábil. La importancia de esta actividad radica en las conciliaciones bancarias que se deben realizar de manera diaria, un retraso en el pago de este dinero ocasiona que no se puedan realizar a tiempo estas conciliaciones.

Por esta razón, la necesidad del negocio requiere de una aplicación en la cual los asociados ingresen el monto total cobrado de cada terminal bancaria, forzándolos a haber realizado el cierre de lote en cada dispositivo. La aplicación fue desarrollada para *e-Reports*.

Los únicos asociados autorizados para realizar el cierre de lote, son aquellos que poseen un nivel de seguridad 1 en el sistema de punto de venta. Creé una tabla en la base de datos de PostgreSQL (`ss_cat_terminals_ccard`) que contiene información relacionada al número de terminales bancarias con el que cuenta el restaurante. De esta manera, únicamente se despliegan el número de terminales bancarias con las que cuenta el restaurante.

Los datos que deben ser ingresados por cada terminal bancaria son:

- Monto que indica el cierre de lote de la terminal. En caso de no haber cobrado con una terminal se debe ingresar 0.0.
- Fecha en que se realiza el cierre de lote.
- Hora en que se realiza el cierre de lote.
- Número de transacciones que reporta la terminal.
- El campo de “Cierre Fallido” se utiliza en caso de que haya existido un error en la terminal bancaria, por ejemplo, por problemas de comunicación.

Una vez ingresados estos datos, se debe presionar el botón de **Obtener montos totales**, con esta acción, se obtiene la cantidad total de dinero registrado en las terminales bancarias y el total de ventas cobradas con tarjeta de crédito/débito que se tienen registrados en el sistema punto de venta. Estos totales, deben coincidir, en caso de no hacerlo, se muestra un mensaje indicando la diferencia y sugiriendo revisar los datos, y si es el caso, corregirlos. Después se debe presionar el botón de **Guardar y confirmar datos**, en esta parte es cuando se le solicita al asociado, elegir su nombre de usuario y teclear su contraseña. Una vez realizado esto de manera correcta se almacenarán los datos ingresados y en caso de haber existido diferencia entre los montos totales del sistema punto de venta y el de las terminales bancarias se manda un correo electrónico automático al restaurante, al gerente de zona y a las personas del área de Ingresos correspondientes. En la figura 2.10 se muestra la pantalla del reporte.

**Cierre Lote de Tarjetas de Crédito**  
 Fecha: **Wednesday, June 15, 2011. 11:10:18 AM**  
 Centro de contribución: **222 - DIVISION DEL NORTE**

**Fecha de Negocio: 11-06-15**

Obtener montos totales    Guardar y confirmar datos

Terminal	Monto	Fecha	Hora (HH:MM)	Num. Transacciones	Cierre Fallido
1					<input type="checkbox"/>
2					<input type="checkbox"/>
3					<input type="checkbox"/>
4					<input type="checkbox"/>

Monto Total Tarj. Crédito	Monto Total SUS

Figura 2.10: Cierre de Lote de Tarjeta de Crédito en e-Reports

Existe además, una alarma que funciona a manera de recordatorio, en caso de que no se haya realizado el cierre de lote a las 23:45 horas. La alarma, realizada en Perl

y Xdialog, consiste en una ventana que aparece en la computadora del restaurante recordando ingresar los datos del cierre de lote, así como también la generación de *beeps* en la misma computadora, y en caso de ser restaurante PH, en todas las terminales. Al desactivar la alarma, ésta solicitará por el usuario y contraseña de quien lo está realizando, esto con el fin de almacenar esta información en un archivo. En caso de que nadie la desactive, ésta automáticamente se apaga después de 15 minutos.

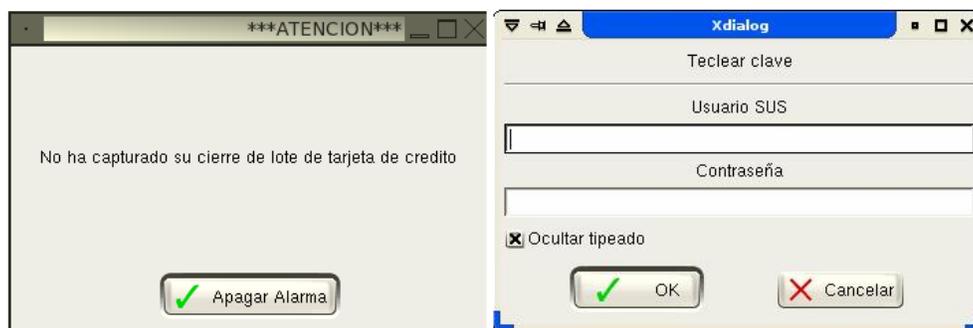


Figura 2.11: Ventanas de Xdialog de la alarma

Cada cierre de lote realizado en *e-Reports* es almacenado en un archivo de texto plano, con lo cual se puede tener un registro de este reporte. A continuación se muestra un ejemplo de archivo almacenado:

```
222|1|11-05-14|11-05-14|20:52|4874.20|5190.2|5190.2|0.0|28|SENI|14-5-2011 21:16:13
222|2|11-05-14|11-05-14|21:10|316|5190.2|5190.2|0.0|1|SENI|14-5-2011 21:16:13
222|3|11-05-14|11-05-14|00:00|0|5190.2|5190.2|0.0|0|SENI|14-5-2011 21:16:13
222|4|11-05-14|11-05-14|00:00|0|5190.2|5190.2|0.0|0|SENI|14-5-2011 21:16:13
```

Los campos indican:

**Primer campo** es el número del restaurante.

**Segundo campo** es el número de terminal bancaria.

**Tercer campo** es la fecha en la que se capturó el cierre de lote.

**Cuarto campo** es la fecha negocio, la cual es la fecha que tiene el sistema punto de venta. En algunos casos puede que no coincida la fecha actual con la fecha de negocio.

**Quinto campo** es la hora que marca la terminal bancaria al momento de realizar el cierre de lote.

**Sexto campo** es el monto cobrado en la terminal bancaria.

**Séptimo campo** es el monto total cobrado de las terminales bancarias.

**Octavo campo** es el monto total que reporta el sistema punto de venta que fue pagado con tarjeta de crédito/débito.

**Noveno campo** es la diferencia entre el monto total que reporta el sistema punto de venta y monto total cobrado de las terminales bancarias.

**Décimo campo** es el número de transacciones cobradas en la respectiva terminal bancaria.

**Onceavo campo** es el nombre de usuario que realizó el cierre de lote.

**Doceavo campo** es el timestamp del sistema al momento de realizar el cierre de lote.

## Capítulo 3

# Proyecto de Control de Asistencia

La tecnología de lectores de huellas digitales está siendo usada en una gran variedad de aplicaciones e industrias. Provee mayor seguridad y evita el mantenimiento de contraseñas. La situación actual del negocio es que las personas en los restaurantes poseen una contraseña con la cual registran su entrada, la cual les permiten comenzar a realizar operaciones el sistema punto de venta, y salida. Las horas trabajadas son tomadas en cuenta para el cálculo de la nómina. Las contraseñas pueden ser compartidas, dando lugar a que algún compañero realice las entradas de los demás, aún cuando éstos lleguen tarde. Así mismo la salida es un problema, ya es muy común que los empleados olviden realizar su salida, y el gerente puede ingresar sus horas de salida, lo que conlleva a malentendidos. Toda esta situación afecta directamente a la nómina, ya que no se están contando de manera real las horas trabajadas.

La oportunidad de negocio consiste en poder tener la certeza de que la persona que realiza la entrada/salida es realmente ella y que no hay posibilidad de suplantar a esta persona, con lo que se tendrían ahorros significativos en gastos de nómina. Además evitar las contraseñas para usar el sistema punto de venta implica una mayor seguridad y responsabilidad en las transacciones realizadas en él.

Las huellas digitales se pueden considerar como una tarjeta de identidad visible que la naturaleza nos dio. Éstas poseen un diseño único que representan únicamente a una persona. Las huellas digitales son diminutas arrugas en la piel de nuestros dedos, tanto de las manos como de los pies, las cuales resultaron de la evolución de nuestros ancestros y cuya finalidad es la de poder sujetar de mejor manera objetos. Además sirven para poder transmitir mejor las vibraciones lo que ocasiona que se puedan distinguir las características de las superficies.

Las huellas digitales se forman de acuerdo a una combinación de la carga genética del individuo y los factores ambientales en los cuales se desarrolla, por lo que es casi

imposible que el patrón de una huella digital se repita. Como consecuencia, las huellas digitales son únicas en cada individuo, incluso en gemelos. Dos huellas digitales pueden parecer iguales a simple vista, pero si se observan con más cuidado, ya sea por un investigador entrenado o un software especializado, éstos pueden encontrar las diferencias. Se les llama *minutias* a las características principales de una huella digital, con las que se pueden realizar comparaciones con otras huellas. Ejemplos de minutias son, que si una arruga tiene forma de Y, de U, el abrupto final de una arruga, etc. Las minutias son los patrones que se buscan y analizan en los programas especializados en reconocimiento de huellas digitales. En la figura 3.1 se muestra a detalle una huella digital.



Figura 3.1: Huella Digital

### 3.1. Tipos de sensores

Existen tres tipos de sensores (*scanners*) de huellas digitales: ultrasónicos, ópticos y capacitivos, de los cuales los más comunes son los últimos dos.

Los scanners de huellas digitales involucran la captura de una imagen digital de la huella usando luz visible. Este tipo de sensor es en esencia una cámara digital. La capa superior del sensor, donde el dedo es colocado, se conoce como la superficie de contacto. Debajo de esta capa existe una capa que ilumina la superficie del dedo. La luz reflejada del dedo llega a un dispositivo de carga acoplada (*CCD, charge-coupled device*) que captura la imagen de la huella digital. Una superficie de contacto sucia o rayada ocasiona una mala imagen de la huella digital.

Los scanners capacitivos, al igual que los ópticos, generan una imagen de la huella digital, pero en lugar de capturar la imagen mediante luz, se utiliza el principio de capacitancia. El sensor que contiene arreglos de pixeles funciona como una placa de un capacitor, la capa de la piel, conocida como dermis (que conduce electricidad) es la otra placa, y la epidermis funciona como un dieléctrico.

## 3.2. Ventajas y desventajas

Existen diferentes maneras en las cuales un sistema de seguridad puede verificar que alguien es un usuario autorizado. La mayoría de los sistemas buscan por una o más de las siguientes sentencias:

- Qué se tiene.
- Qué se sabe.
- Quién es.

Para entrar a un sistema tipo "qué se tiene", se necesita algún dispositivo como un token de autenticación, o una tarjeta RFID<sup>1</sup>. Para el sistema "qué se sabe" se requiere que se ingrese una contraseña o un PIN. El sistema "quién es" busca por evidencia física de que la persona que se dice ser, es realmente ella, en esta clasificación entran las huellas digitales y el reconocimiento de voz.

Estos últimos sistemas tienen varias ventajas sobre los demás, por ejemplo:

- Los atributos físicos son más difíciles de falsificar que las tarjetas de identificación.
- No se puede usar la fuerza bruta para adivinar una huella digital como se hace con las contraseñas.
- No se pueden olvidar las huellas digitales como sucede con una contraseña.

A pesar de ser tan efectivas, no son infalibles, y también tienen desventajas. Los scanners ópticos no siempre pueden distinguir bien entre la imagen de una huella digital, y una huella digital real. Y los scanners capacitivos pueden ser engañados con el molde de una huella digital. En uno de los peores casos, un criminal pudo haber cortado el dedo de una persona para poder acceder al sistema. Es por esto que algunos dispositivos ya cuentan con sensores de calor y pulso para poder distinguir estos casos.

Para hacer estos sistemas más robustos, conviene utilizar el análisis biométrico con un sistema convencional de identificación, como una contraseña. La necesidad del negocio no la requiere directamente, solamente como medio de respaldo en caso de que el dispositivo lector falle.

---

<sup>1</sup>Radio Frequency Identification

### 3.3. Prototipo 1: Lector Microsoft

Al comenzar a realizar el proyecto se contaba con un lector de huellas digitales modelo "Microsoft Fingerprint Reader", el cual es un lector con conexión USB. La desventaja en un principio radicaba en la incertidumbre de que el dispositivo funcionara en una computadora con GNU/Linux. Al conectarlo y ejecutar el comando `dmesg` se observa que el kernel identifica correctamente el lector.



Figura 3.2: Microsoft Fingerprint Reader

Este lector está basado en un chipset conocido como *DigitalPersona U.are.U 4000/4000B* el cual es usado en una gran cantidad de lectores, incluidos los de la compañía fabricante *DigitalPersona*. El lector es del tipo óptico, el cual únicamente captura imágenes y los envía a la computadora y es necesario el uso de software para procesar y comparar las huellas digitales. Además el sensor es capaz de detectar cuando un dedo ha sido colocado o removido y se le puede indicar cuando capturar la huella digital.

Haciendo una investigación se encuentra que existe software libre que permite la lectura y procesamiento de las huellas digitales en GNU/Linux. Este software se llama `libfprint`<sup>2</sup>. `libfprint` es una biblioteca que permite a los desarrolladores añadir soporte de lectores de huellas digitales a sus desarrollo. Algunas de sus características son:

- Está escrito en lenguaje C.
- Depende de la biblioteca `libusb` y `glib` para la comunicación.
- Ofrece una única interfaz de programación para la gran variedad de dispositivos que soporta.
- Soporta la captura y descarga en de imágenes en tiempo real desde el lector.
- Incluye funciones de procesamiento de imágenes.
- Permite el registro de huellas digitales de una persona (enrolamiento) y su verificación.

---

<sup>2</sup><http://www.freedesktop.org/wiki/Software/fprint/libfprint>

Los controladores propietarios de *DigitalPersona* y su kit de desarrollo de software (SDK<sup>3</sup>) envían las imágenes cifradas a través de USB. Debido a que el método de cifrado es desconocido, *libfprint* envía las imágenes sin esta protección. En la figura 3.3 se muestra una huella digital adquirida por el dispositivo.



Figura 3.3: Ejemplo de imagen capturada por el dispositivo

El lector manda una imagen de 348x289 píxeles separadas en dos transacciones, la primera es un encabezado de 64 bytes que es ignorada y después la imagen.

Uno de los grandes problemas con el sistema actual, es que a pesar de que funciona para lo que necesita hacer: tomar venta, generar reportes, etc., el sistema está sumamente desactualizado, por lo que la instalación de nuevo software puede conllevar a la actualización o instalación de nuevo software. Para la compilación del código fuente de *libfprint* se tuvo que instalar el siguiente software:

- *GLib* (ver. 2.28.0). Es una biblioteca de propósito general, originalmente parte de GTK+, pero a partir de la versión 2.0 los desarrolladores decidieron separarlas para crear un nuevo producto. GLib proporciona estructuras de datos avanzadas, como listas doblemente ligadas, tablas hash, arreglos dinámicos, árboles binarios, etc. Además provee de funciones para programación de hilos, colas, acceso seguro a memoria, timers, etc.
- *ImageMagick* (ver. 6.6.0-4). Es una aplicación que sirve para crear, editar y componer imágenes. Puede leer, convertir y guardar imágenes en una gran variedad de formatos.
- *libusb* (ver 1.0.8). Es una biblioteca que proporciona las funciones necesarias para el control en la transmisión y recepción de datos de un dispositivo USB.
- *nss. Network Security Services* es un conjunto de bibliotecas diseñadas por Mozilla para desarrollo de aplicaciones seguras cliente/servidor. Proporciona bibliotecas seguras como SSL<sup>4</sup> y S/MIME<sup>5</sup>. Esta herramienta fue la más difícil de compilar y adaptar para el sistema actual, ya que tuve que adecuar algunos

---

<sup>3</sup>Software Development Kit

<sup>4</sup>Secure Socket Layer

<sup>5</sup>Secure/Multipurpose Internet Mail Extensions

archivos del código fuente para que pudiese compilar. Además de que el código fuente que proporciona Mozilla no sigue un estándar (archivos `configure`, `Makefile` y `autoconf`) y su documentación es deficiente.

Originalmente `libfprint` no es multiusuario, el software está diseñado para almacenar la información de las huellas digitales en el `$HOME` de cada usuario, por lo que es necesario que los usuarios tengan una cuenta en el sistema. Para corregir este problema modifiqué el código fuente de `libfprint` con el objetivo de que se almacenara la información de todos los usuarios en un solo directorio y en subdirectorios de acuerdo al número de empleado registrado en el punto de venta.

Desarrollé una interfaz gráfica usando *Swing* para la entrada/salida y enrolamiento de los empleados. *Swing* es una biblioteca gráfica para Java. Incluye widgets para interfaz gráfica de usuario tales como cajas de texto, botones, desplegados y tablas.

Debido a que `libfprint` está programado en C y la interfaz gráfica es en Java, utilicé JNI (*Java Native Interface*) para funcionar. *JNI* es un framework de programación que permite que un programa escrito en Java ejecutado en la máquina virtual java (JVM) pueda interactuar con programas escritos en otros lenguajes como C, C++ y ensamblador, la desventaja más importante es la pérdida de portabilidad que ofrece Java.

Programé dos funciones en C usando JNI: la de verificación y la de enrolamiento. A continuación se muestra el encabezado del archivo de verificación:

```
JNIEXPORT jint JNICALL Java_PRBAS_PRBAttendance_verify
(JNIEnv *env, jobject obj, jint finger, jstring p, jstring u)
```

El apuntador `JNIEnv *env`, junto con `jobject obj` siempre deben ir al programar con JNI, ya que son interfaz con la máquina virtual de Java, después pueden ir los argumentos necesarios para la función, en este caso, `finger` que es un entero y representa al dedo a ser verificado (0-pulgar izquierdo, 1-índice izquierdo, 2-medio izquierdo, ... 9-meñique derecho), la cadena `p` indica la ruta donde se encuentra almacenada la información de las huellas digitales, y `u` es el usuario a verificar.

Una de las partes más importantes es poder ingresar al sistema punto de venta la información de entrada/salida de un empleado. No se tenía documentado exactamente cómo y dónde es que el sistema ingresa esta información, por lo que tuve que revisar el código fuente para conocerlo. Para ingresar la entrada/salida el empleado debe teclear su clave en el sistema y posteriormente ingresar a las opciones de *b) Entrada de Empleado* o *c) Salida de Empleado*. En la figura 3.4 se muestra la pantalla donde los asociados realizan su entrada o salida.

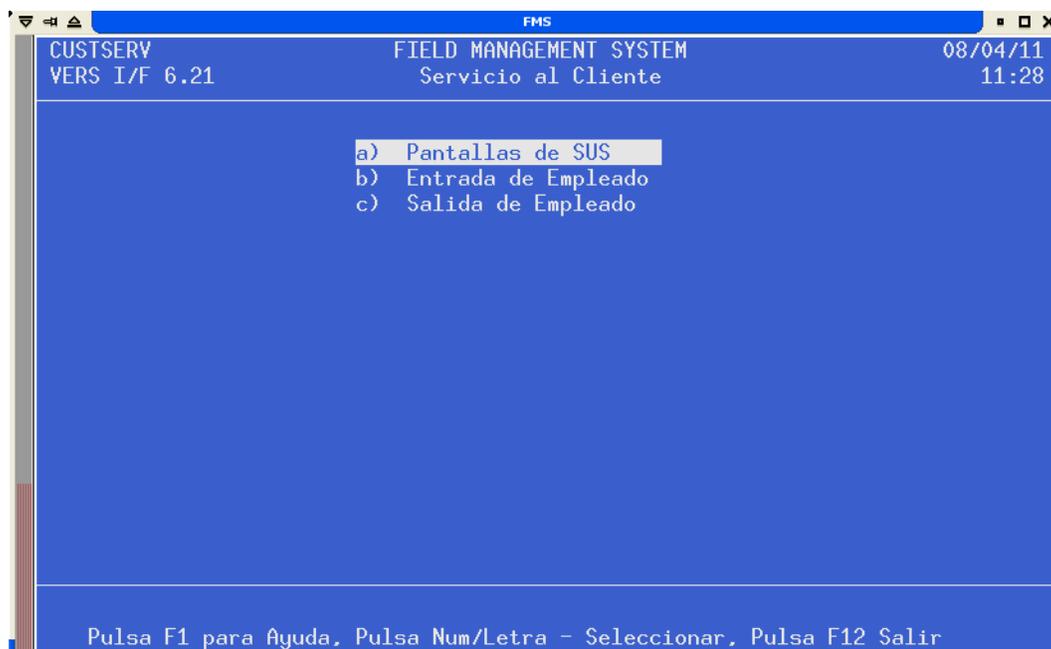


Figura 3.4: Pantalla de entrada/salida en sistema punto de venta.

Al realizar la entrada/salida el sistema utiliza los siguientes archivos binarios:

1. `/usr/fms/data/hrcempl.dat`
2. `/usr/fms/data/tkeclk.dat`
3. `/usr/fms/data/tkpaytrn.dat`

El primero contiene los datos de todos los empleados, una línea por cada empleado: nombre, apellido, edad, dirección, etc. El siguiente archivo únicamente sirve para saber si un usuario ya registro su entrada, y contiene una copia de la línea correspondiente del archivo antes mencionado por cada usuario registrado. Y el último archivo contiene una línea por cada operación realizada de entrada y salida, identificadas con TI para entrada y T0 para la salida.

El sistema punto de venta se utiliza principalmente mediante una interfaz gráfica, pero ésta se encuentra soportada por una gran cantidad de comandos que se pueden ejecutar en el shell; entre todos estos comandos, existe el comando `/usr/fms/etc/sysdd` que sirve para poder convertir un archivo binario a texto plano, y viceversa, siempre y cuando se sepa la estructura del archivo binario. Por ejemplo, a continuación se muestra una parte de la estructura del archivo `/usr/fms/data/tkpaytrn.dat`:

```
typedef struct  tkpaytrn_rec
{
    long    date;
    short   tran_group;
    char    fill1[2];
    char    emp_ssn[16];
    char    source;
    char    trancode[3];
}
```

El comando `/usr/fms/etc/sysdd` recibe de argumentos la operación a ejecutar, convertir de binario a texto es `btoa` y al contrario es `atob`. Después la estructura del archivo y al final el archivo, para este caso sería:

```
/usr/fms/etc/sysdd btoa lhs2s16cs3 /usr/fms/data/tkpaytrn.dat
```

La `l` es para el tipo de dato `long`, la `h` para `short`, la `s` para arreglos `char` seguido por el número de elementos y `c` para un sólo `char`.

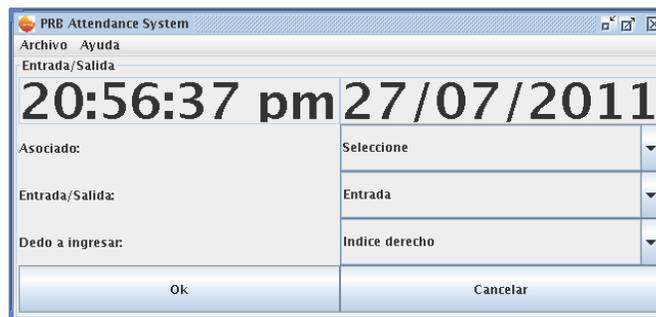


Figura 3.5: Pantalla de entrada/salida con lector de huellas digitales.

En la figura 3.5 se observa la interfaz en Swing de la aplicación de Entrada/Salida. Como se puede observar, ésta cuenta con la fecha y hora actual, con la cual se almacenará el registro. Al seleccionar el asociado se muestra una lista de todos los empleados registrados en el sistema punto de venta, los cuales son extraídos a través de comandos de SUS. En la figura 3.6 se muestra el despliegue de la lista de asociados.

La selección de entrada/salida sirve para ingresar TI o TO en el archivo correspondiente. Un área de oportunidad en el desarrollo es la detección automática si es entrada o salida. Lo que sí debe ser seleccionado, debido a la naturaleza de la biblioteca `libfprint`, es la selección del dedo de la huella digital a ingresar. Esto se debe a que el software no es capaz de realizar la comparación entre todas las huellas digitales almacenadas de un mismo usuario, y mucho menos de realizar esta comparación para las huellas de todos los usuarios. Por lo que se le debe decir el usuario y el dedo a ingresar, para que el software únicamente haga la comparación

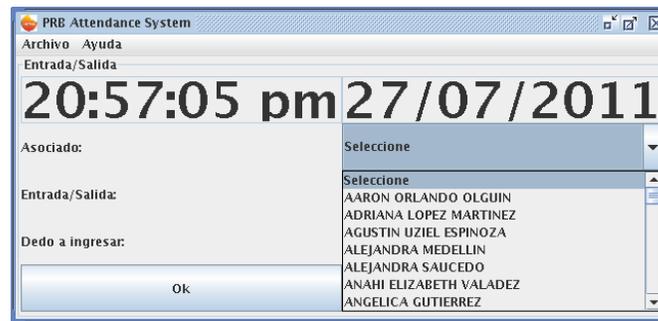


Figura 3.6: Listado de asociados.

de la huella a verificar y la que fue previamente almacenada. Al dar click en OK, se solicita al usuario que coloque el dedo correspondiente en el lector de huellas digitales, se recomienda colocarlo por alrededor de 3 segundos y retirarlo; el lector, como se mencionó anteriormente, es capaz de detectar la colocación y remoción de un dedo. En caso de que no se haya verificado la huella digital, la interfaz solicitará que se vuelva a colocar el dedo. Esta interfaz aparece una vez que el usuario ha seleccionado la opción correspondiente (b) o c) de la figura 3.4).

Para el almacenamiento de las huellas digitales o el enrolamiento, se utiliza una interfaz más simple, en la cual únicamente se selecciona el usuario y el dedo. El procedimiento al hacer click en OK es el mismo que en la verificación. Por razones de seguridad y de evitar el múltiple enrolamiento de una misma huella digital para diferentes usuarios, esta funcionalidad se activa a petición del gerente del restaurante.

Es importante mencionar, que la biblioteca `libfprint` es capaz de obtener una fotografía de la huella digital, ésta no es utilizada para el análisis de las minutias, `libfprint` genera un archivo de datos binarios, el cual no puede ser leído por un editor de textos. La imagen de la huella digital es eliminada, ya que de acuerdo a la nueva *Ley de Protección de Datos Personales en Posesión de los Particulares* se debe tener un mayor cuidado en el manejo en este tipo de datos para no caer en problemas legales.

Es por esto que busqué un dispositivo más especializado, el cual se describe con más detalle en la sección 3.4.

En las pruebas realizadas en este dispositivo observé que la verificación exitosa depende de la forma en que se haya capturado la huella digital en el enrolamiento. Debido a que el software únicamente usa una sola huella digital para obtener las minutias, al colocar de otra forma el dedo, ocasiona que muchas minutias no correspondan y arroje como resultado que la huella digital no corresponde. Esto ocasionaría un enorme problema en los restaurantes, elevando el número de soportes técnicos diariamente debido a que los empleados no pueden registrar su entrada ni

entrar al sistema.

### 3.4. Prototipo 2: Lector ZK Software

Existen en el mercado una gran variedad de dispositivos lectores de huella digitales, que además de almacenarlas y verificarlas, son capaces de generar diversos reportes de acuerdo a la información de sus registros utilizando software externo o dentro del dispositivo. Una de las principales características que deben tener los lectores para poder integrarlos fácilmente al sistema en el restaurante, es que se puedan descargar los registros a la computadora sin tener un software intermediario, el cual siempre es para Windows.

Encontré el lector de huellas digitales de la compañía ZK Software modelo U300 (ver figura 3.7) el cual cuenta con un puerto Ethernet RJ45 y USB. Es capaz de almacenar 1600 huellas digitales, 50000 registros, tiene sistema operativo GNU/Linux.



Figura 3.7: Lector de huellas digitales U300 de ZKSoftware

Una de las principales características de este lector, es que una vez configurada su dirección IP, se puede acceder a una página web para consultar los registros almacenados como se muestra en la figura 3.8. Se pueden obtener registros del día actual, de ayer, de la semana pasada, o bien especificar la fecha para desplegarlos.

El procedimiento para obtener los datos y el registro de los empleados es el siguiente:

1. Mediante cURL<sup>6</sup> se debe obtener la cookie de la página web del lector de huellas digitales, para poder consultar los registros.
2. Con la cookie previamente descargada y con cURL se accede a la dirección `http://192.168.101.23/csl/report`. En este caso la IP del lector de huellas

---

<sup>6</sup>cURL es una herramienta para usar en un intérprete de comandos para transferir archivos con sintaxis URL, soporta FTP, FTPS, HTTP, HTTPS, TFTP, SCP, SFTP, Telnet, DICT, FILE y LDAP.

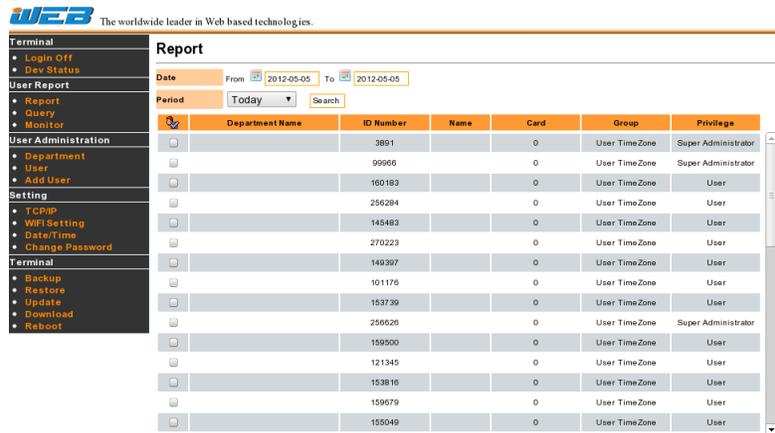


Figura 3.8: Página web del dispositivo mostrando usuarios registrados.

es 192.168.101.23. Con esto se obtiene la imagen de la figura 3.8 que sirve para obtener los números de empleados de los usuarios registrados.

3. Con cada uno de los números de empleados obtenidos en el punto anterior, se consulta con cURL el URL `http://192.168.101.23/action=run&sdate=2012-02-20&edate=2012-02-20&uid=23021` con la fecha inicial y final igual para obtener el día especificado, además del número de empleado con el que se registro, de esta manera se obtiene la página con los registros almacenados para ese usuario. En caso de que el usuario no tenga registros para ese día, se pasa al siguiente usuario.

Date	ID Number	Name	IN	OUT	IN	OUT	IN	OUT	More
2012-05-05	145483		08:06:38						<a href="#">More</a>
2012-05-05	149397		10:04:21						<a href="#">More</a>
2012-05-05	153739		07:14:52						<a href="#">More</a>
2012-05-05	121345		10:07:58						<a href="#">More</a>
2012-05-05	263448		07:09:07						<a href="#">More</a>
2012-05-05	130226		08:10:32						<a href="#">More</a>
2012-05-05	155892		09:57:54						<a href="#">More</a>
2012-05-05	157284		08:24:46						<a href="#">More</a>
2012-05-05	160231		11:06:07						<a href="#">More</a>
2012-05-05	158187		08:13:04	12:15:00					<a href="#">More</a>
2012-05-05	158386		09:09:15						<a href="#">More</a>
2012-05-05	158384		07:09:20						<a href="#">More</a>
2012-05-05	155121		07:09:14	12:03:15					<a href="#">More</a>

Figura 3.9: Página web mostrando los registro en un día específico.

4. Con ayuda de Perl y el módulo `HTML::TableExtract` se parsea la página HTML obtenida en el punto anterior y se obtiene principalmente, la columna de hora. Se parsean todas los renglones de hora que existan. Por lo menos debe existir un registro de hora, la cual se considera la entrada.
5. Con el ID del usuario, la fecha y la hora de entrada y/o salida, se realiza una

consulta a la base de datos de PostgreSQL para revisar si no ha habido registro de este usuario.

- Si tiene un registro de hora, es su entrada, se registra en base de datos y en los archivos necesarios en el punto de venta.
- Si tiene dos registros de hora, es su salida, y se registra en la tabla de la base de datos y se actualizan los archivos correspondientes del punto de venta.
- Existe una opción de configuración en el lector biométrico en donde se establece el tiempo mínimo en el que puede registrar otro evento. Éste parámetro se estableció en cuarenta y cinco minutos. Durante la primera fase de pruebas se desconocía este parámetro y se observaba que los usuarios hacían un uso incorrecto del lector, colocando más de una vez su dedo en un lapso muy corto de tiempo, por lo que se tenían muchos registros innecesarios.

### 3.4.1. Descripción de scripts

A continuación se describen a grandes rasgos los scripts involucrados en todo el proceso de registro:

- `getFPdata.sh`: Mediante este script, que es ejecutado mediante un cron, se realiza todo el proceso de registro. Recibe de parámetro la fecha a procesar. Dentro de este script obtienen mediante `getIDs.pl` los números de empleado que se tienen registrados. Con estos números de empleado se realizan consultas a sus registros mediante `parseHours.pl`.
- `getIDs.pl`: Es el encargado de obtener los números de empleado registrados. Pero además de los números de empleado, obtiene los valores de los *checkboxes* que se muestran en la figura 3.8 para poder hacer las consultas respectivas.
- `parseHours.pl`: Con él se obtienen, en caso de tener, los registros de entrada/salida de cada empleado, así como la inserción o actualización en la tabla de PostgreSQL `pp_emp_check` y el registro de las horas en los archivos del sistema de punto de venta.
- `clock_employee.sh`: Realiza el registro en los archivos punto de venta para que puedan ser analizados por los procesos de nómina.

El script `getFPdata.sh` se lleva a cabo en un proceso CRON cada cinco minutos. Lo que garantiza que una vez que el usuario ha realizado su verificación en el lector, éste pueda utilizar de manera casi inmediata el sistema punto de venta. No

todos los usuarios utilizan directamente las terminales, el único puesto que maneja directamente las terminales son las personas que atienden los destinos de servicio a domicilio y auto (este destino es aquel en que el cliente ingresa con su carro al estacionamiento y realiza la orden desde él); por lo que se ejecute cada cinco minutos el proceso que lee y procesa los datos del lector es correcto para la operación.

Además elaboré unos scripts para monitorear que el dispositivo se encuentre conectado a la red y pueda mandar correos electrónicos dirigidos al gerente del restaurante y de área para que estén enterados de la pérdida de comunicación del lector por alguna causa.

El gerente del restaurante no tiene acceso a la página web del lector biométrico, por lo que para que pueda visualizar los registros de una manera rápida y fácil desarrollé el reporte titulado "Reporte de Asistencias en Biométrico" dentro de eReports, donde puede seleccionar una fecha para poder conocer los horarios en que laboró cada empleado y poder cotejar con la nómina de cada semana, que ésta sea correcta.



Reporte de Asistencias						
No. empl	Nombre	Hora Entrada 1	Hora Salida 1	Hora Entrada 2	Hora Salida 2	Total horas
158206	CARLOS YULEM GARCIA	9:59:11	17:56:33			07:57:22
156584	ALBERTO BARCENAS	15:14:47	23:19:52			08:05:05
155505	ILSE NAYELI GOMEZ	15:26:26	23:6:2			07:39:36
154586	ADRIAN GONZALES	18:36:57	23:21:4			04:44:07
150659	BRANDON HERNANDEZ	18:2:53	22:3:49			04:00:56
150059	ALVARO REBOLLR REBOLLAR	10:2:29	18:4:13			08:01:44
121861	MARCOS ROBERTO ACOSTA	10:12:15	18:3:54			07:51:39
105333	CLAUDIA PARRA PARRA	16:46:35	21:15:42			04:29:07

Figura 3.10: Reporte de asistencia en eReports.

A la fecha el desarrollo ya se encuentra funcionando al 100 % y en prueba en dos restaurantes, no se ha liberado a más centros debido a que el área de Operaciones y Legal debe realizar adecuaciones a los contratos actuales, así como la de elaborar un plan de entrenamiento a los empleados por parte del área de Entrenamiento con asesoría de Operaciones y Desarrollo a Restaurantes.

Es muy importante que en el momento del enrolamiento o la verificación se coloque de manera adecuada el dedo, ya que de otra manera, existirán errores de captura/lectura. La correcta forma de colocar el dedo en el centro del lector de una manera plana y firme como se muestra en la figura 3.11.

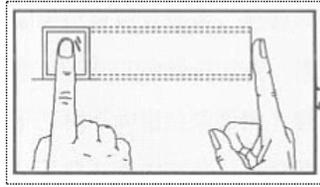


Figura 3.11: Forma ideal para colocar el dedo en el lector.

Es considerada una mala colocación del dedo en el lector, cuando el dedo no se encuentra completamente plano, no se encuentra en el centro del lector, ya sea que esté muy a la derecha, izquierda o abajo. O cuando el dedo no haga contacto la huella digital con el lector como se muestra en la figura 3.12.

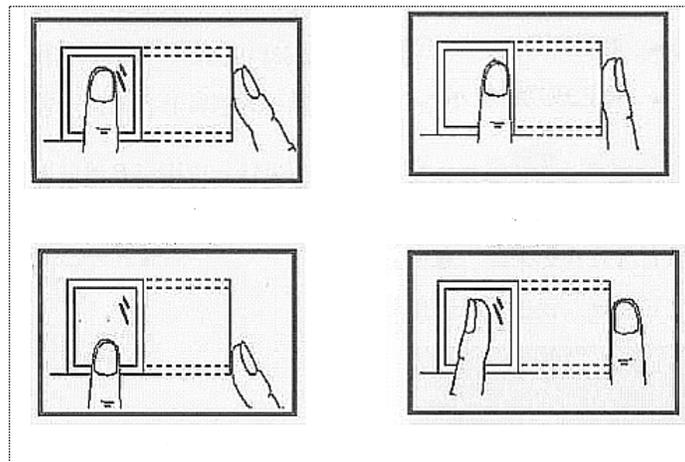


Figura 3.12: Forma incorrectas para colocar el dedo en el lector.

# Capítulo 4

## Resultados

La eliminación del impresor de auditoría trajo consigo muchos beneficios los cuales impactaron económicamente a la empresa. Una oficina *paperless* es aquella en donde el uso de papel se ha eliminado o su uso ha disminuido en gran medida. Con esta iniciativa se eliminó la compra de rollos de papel usados por el impresor, la adquisición y mantenimiento de impresores y cintas para éstos. El número de tickets en un día promedio por restaurante es de alrededor de trescientos, además de que todos éstos se deben imprimir para entregárselo al cliente, el impresor de auditoría también hacía una impresión simplificada de cada ticket.

Por cuestiones de auditoría de Hacienda, todos los rollos de papel con los tickets de auditoría se debían guardar por un periodo determinado de tiempo, lo que ocasionaba gastos de almacenaje y transporte.

Ahora la manera de consultar todos estos tickets es a través de un servidor web centralizado para la consulta por restaurante y por día. El ahorro anual de este proyecto es de alrededor MXN \$700 mil anuales.

Gracias al reporte de Estimado de Usos Ideales para KFC, la operación en el restaurante se encuentra más preparada para tener el nivel adecuado de inventario de los productos críticos en el restaurante. Así mismo, la merma se ha visto disminuida, ya que se puede saber con anticipación, por ejemplo la cantidad de papas a la francesa, cuántos kilos de col, cuántos paquetes de marinador, etc. se utilizarán. Una de las claves para el correcto funcionamiento de los usos ideales es que las recetas de los productos utilicen la cantidad exacta de ingredientes que se indican. Ya que en caso contrario el uso ideal puede tener cierto margen de error. Gracias a la ayuda que se le proporciona a la operación del restaurante, este reporte lo migré a Pizza Hut por solicitud del equipo operativo de la marca.

La captura de cierre de lote ha ayudado a que el staff gerencial del restaurante

esté al pendiente de realizar cada día al finalizar la operación, el cierre de lote en cada terminal bancaria. Esto con la finalidad de que el banco deposite el dinero al siguiente día hábil, agilizando las tareas contables y financieras de la empresa. Con el envío automático de un correo electrónico en caso de que la terminal tenga algún problema al realizar el cierre de lote, los soportes proporcionados por el proveedor se realizan de manera rápida y efectiva, evitando en mayor medida que el gerente tenga que llamar directamente al proveedor para levantar un ticket de servicio.

Actualmente el proyecto de control de asistencia mediante lector biométrico se encuentra en pruebas en dos restaurantes, uno de cada marca. El proyecto permite al departamento de Nóminas:

- Tener un control sistematizado de la asistencia en restaurantes.
- Justificaciones para el otorgamiento de prestaciones.
- Control de incapacidades y permisos del personal.
- Posibilidad de ampliar la emisión de nuevos reportes de labor.
- Asegurar al 100 % la integridad del pago de premio por asistencia.
- Control adecuado de absentismo para descontarse de las cuotas obrero patronales.

El beneficio que tiene el área de Operaciones junto con el gerente del restaurante:

- Visualización de la asistencia y staff gerencial.
- Checado diario de entrada/salida.
- Programación de descansos semanales.
- Control de retardos, faltas e incapacidades.
- Control de horarios trancos.

Para Recursos Humanos:

- Enrolamiento a partir del ingreso.
- Cambios de estructura / movilidad de personal.
- Incapacidades y absentismo.

- Excepciones de pago.

Uno de los mayores obstáculos que tuvo el proyecto en las primeras semanas de operación fue la renuencia por parte de los asociados de ingresar su entrada y/o salida a través del dispositivo biométrico. Poco a poco ha aumentado la disciplina de realizar esta tarea, ya que en caso de que olviden ingresar su salida, no hay manera de poder contabilizar el tiempo laborado para generar la nómina.

De acuerdo a cálculos preliminares realizados por parte del equipo de Nómina, se tiene contemplado un ahorro de MXN \$3 millones anuales en pago de cuotas obrero patronales en primera instancia, más el ahorro en la nómina gracias a tiempos reales de labor.

# Conclusiones

Las herramientas mostradas en este informe han cumplido a mejorar la eficiencia en la operación de los restaurantes, lo que conlleva a generar ahorros importantes para la empresa. Normalmente el equipo de Operaciones de cada marca, es con quien tengo juntas para conocer las necesidades y requerimientos, y de las cuales se tiene una lluvia de ideas para elaborar el sistema. El éxito de las herramientas ha propiciado que, por ejemplo, el equipo de Operaciones de KFC, solicité migrar la herramienta elaborada para PH, a sus restaurantes.

De los trabajos descritos en este informe de actividades, se puede concluir que éstos han logrado un gran ahorro de tiempo en las actividades de los gerentes en los restaurantes, lo que implica que puedan ocuparse en otras actividades pendientes, directas en la operación del restaurante. El adecuado uso de las tecnologías de la información permite que los datos proporcionados al momento y de manera veraz contribuyan a la toma de decisiones por parte del gerente de restaurante y del área de Operaciones. Por ejemplo, con el reporte de “Estimado de Usos Ideales” el gerente puede prever la cantidad de producto que se puede consumir, y estar preparado, solicitando a los proveedores más producto de acuerdo al inventario.

Mediante el uso del lector de huellas digitales, se espera un ahorro en el concepto de pago de nómina, ya que ésta se paga de acuerdo al número de horas trabajadas. Contabilizando de manera exacta y sin problemas de suplantación, las horas pagadas serán realmente las justas. Además de que la integración del sistema punto de venta con el dispositivo se realiza de manera transparente sin el uso de software propietario, lo que implica ahorro de dinero, al evitar el pago de licencias o desarrollos externos.

Como contratiempos y problemas que se han tenido en la puesta en marcha del lector de huellas digitales en los restaurantes es la costumbre de los asociados de que el gerente es quien ingresa las horas a pagar a través de un sistema en la intranet. Pudiendo, normalmente, beneficiarlos. La renuencia y olvido de los asociados a registrar, la entrada/salida a través del lector ha causado que no se puedan pagar todas las horas que el asociado indica que laboró. En caso de que el lector no se encuentre en funcionamiento por alguna razón, se tiene un proceso que constantemente revisa su estatus y en caso de fallo permite a los asociados poder registrar su entrada/salida a

través del sistema punto de venta, como normalmente lo deben hacer. Además de que se manda un correo electrónico avisando al restaurante, gerente de área y personal de Nóminas.

Esta aplicación puede ser mejorada con:

- Un sistema ligado con el de nómina y alta de asociados, para que cuando firmen contrato en las oficinas, también se registren sus huellas digitales y éstas sean mandadas al lector del restaurante correspondiente.
- Ligado al punto anterior, un proceso, que en caso de que el asociado no haya podido registrar sus huellas en oficinas, poder mandar el número de asociado al dispositivo para que únicamente el gerente busque el número de asociado y dé de alta únicamente las huellas para evitar errores al capturar el número de asociado.
- Un proceso que comparé al final del día los registros almacenados en el archivo del punto de venta y los que tiene el lector, para evitar discrepancias.

La experiencia obtenida al laborar en Premium Restaurant Brands desarrollando sistemas ha sido de gran valor para mi, ya que he desarrollado más mis habilidades de programación, así como también he adquirido conocimiento de la industria de restaurantes que antes desconocía. Los conocimientos obtenidos en la universidad, únicamente sirven de base para el desarrollo que uno tenga como profesionalista y es de particular importancia ampliar estos conocimientos y tenerlos siempre actualizados, ya que como sabemos, la tecnología avanza rápidamente y surgen nuevas herramientas, nuevo hardware, nuevos lenguajes de programación, etc.

El trabajo en equipo y de tipo multidisciplinario es fundamental para poder llevar a cabo los desarrollos requeridos, la colaboración implica alcanzar metas comunes y entender los requerimientos, es decir estructurar las necesidades del negocio. La comunicación constante, con las diversas áreas involucradas en el proyecto tendrá como resultado un proyecto exitoso y acorde con las necesidades del cliente.

Una de las características que he visto al laborar en el área de sistemas, es que existen varios proyectos de una o distintas áreas, incluida la misma de sistemas, que conlleva a desarrollar un alto nivel de responsabilidad y organización, para poder terminar los desarrollos en el tiempo establecido. Una de las áreas de oportunidad que tiene Sistemas en Premium Restaurant Brands es la implementación de un modelo de procesos que le ayude al desarrollo de los proyectos de una forma más estructurada. Sí es verdad que todos los proyectos se han llevado a cabo, hasta su finalización, el proceso de desarrollo se puede mejorar. La madurez de un proceso es el nivel al cual está explícitamente documentado, gestionado, medido, controlado y continuamente mejorado. Un modelo de procesos es un conjunto estructurado de elementos que

describen las características de procesos efectivos y de calidad, indicando “qué hacer”, no “cómo hacer” ni “quién hace”. Modelos de procesos como ISO o MoProSoft pueden ayudar a la mejora en el desarrollo y gestión de los proyectos a través de estandarización de prácticas, evaluación e integración de mejoras continuas. Las actividades que se deben desarrollar al implementar un modelo de gestión de procesos, puede resultar en una burocratización de éstas, lo cual es una desventaja. Por lo que puede convenir tomar ideas de estos modelos para obtener lo mejor de ellos, sin llegar al papeleo que puede acarrear.

# Apéndice A

## Cambios a libfprint

Realicé los siguientes cambios al archivo `data.c` de la librería `libfprint` para que pueda registrar y reconocer a más de un usuario. A continuación se muestra la salida del comando:

```
diff -u data.c.orig data.c
```

El cual sirve para poder distribuir únicamente el cambio sin necesidad de mandar el archivo completo. A esto se le llama un parche, el cual se aplica de la siguiente manera:

```
patch < data.patch
```

```
--- data.c.orig      2011-05-10 16:36:10.000000000 +0000
+++ data.c           2011-05-10 16:36:10.000000000 +0000
@@ -49,8 +49,9 @@

    static char *base_store = NULL;

-static void storage_setup(void)
+static void storage_setup(char *path, char *user)
    {
+    /*
        const char *homedir;

        homedir = g_getenv("HOME");
@@ -58,8 +59,9 @@
```

```

        homedir = g_get_home_dir();
    if (!homedir)
        return;
+   */

-   base_store = g_build_filename(homedir, ".fprint/prints", NULL);
+   base_store = g_build_filename(path, user, "fprint/prints", NULL);
    g_mkdir_with_parents(base_store, DIR_PERMS);
    /* FIXME handle failure */
}
@@ -225,7 +227,7 @@
    * \returns 0 on success, non-zero on error.
    */
API_EXPORTED int fp_print_data_save(struct fp_print_data *data,
-   enum fp_finger finger)
+   enum fp_finger finger, char *pathd, char *user)
{
    GError *err = NULL;
    char *path;
@@ -235,7 +237,9 @@
    int r;

    if (!base_store)
-       storage_setup();
+       storage_setup(pathd, user);
+
+   //printf("fp_print_data_save: %s\n", base_store);

    fp_dbg("save %s print from driver %04x", finger_num_to_str(finger),
        data->driver_id);
@@ -334,14 +338,17 @@
    * \returns 0 on success, non-zero on error
    */
API_EXPORTED int fp_print_data_load(struct fp_dev *dev,
-   enum fp_finger finger, struct fp_print_data **data)
+   enum fp_finger finger, struct fp_print_data **data,
+   char *pathd, char *user)
{
    gchar *path;
    struct fp_print_data *fdata;
    int r;

-   if (!base_store)
-       storage_setup();
+   //if (!base_store)
+       storage_setup(pathd, user);
+
+   //printf("fp_print_data_load: %s\n", base_store);

```

```

        path = get_path_to_print(dev, finger);
        r = load_from_file(path, &fdata);
@@ -551,7 +558,7 @@
    * \returns a NULL-terminated list of discovered prints, must be freed with
    * fp_dscv_prints_free() after use.
    */
-API_EXPORTED struct fp_dscv_print **fp_discover_prints(void)
+API_EXPORTED struct fp_dscv_print **fp_discover_prints(char *pathd, char *user)
{
    GDir *dir;
    const gchar *ent;
@@ -563,7 +570,9 @@
    unsigned int i;

    if (!base_store)
-        storage_setup();
+        storage_setup(pathd, user);
+
+    //printf("fp_discover_prints: %s\n", base_store);

    dir = g_dir_open(base_store, 0, &err);
    if (!dir) {

```

# Apéndice B

## Modificación a tabla pp\_emp\_check

Ya existía la tabla `pp_emp_check`, pero le realicé algunas modificaciones para soportar horas truncas, identificar si fue registro manual o mediante el lector biométrico.

```
--
-- PostgreSQL database dump
--

SET statement_timeout = 0;
SET client_encoding = 'LATIN1';
SET standard_conforming_strings = off;
SET check_function_bodies = false;
SET client_min_messages = warning;
SET escape_string_warning = off;

SET search_path = public, pg_catalog;

ALTER TABLE ONLY public.pp_emp_check DROP CONSTRAINT pp_emp_check_pkey;
DROP TABLE public.pp_emp_check;
SET search_path = public, pg_catalog;

SET default_tablespace = '';

SET default_with_oids = false;

--
-- Name: pp_emp_check; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
--

CREATE TABLE pp_emp_check (
    store_id smallint DEFAULT 0 NOT NULL,
    date_id date NOT NULL,
    emp_num character varying(6) NOT NULL,
    timein1 timestamp without time zone,
    timeout1 timestamp without time zone,
    timein2 timestamp without time zone,
    timeout2 timestamp without time zone,
    entry_type character varying(5)
);

ALTER TABLE public.pp_emp_check OWNER TO postgres;

--
-- Name: COLUMN pp_emp_check.entry_type; Type: COMMENT; Schema: public; Owner: postgres
--

COMMENT ON COLUMN pp_emp_check.entry_type IS 'Ingreso manual o con lector de huellas';

--
-- Name: pp_emp_check_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres; Tablespace:
--
```

```
ALTER TABLE ONLY pp_emp_check
  ADD CONSTRAINT pp_emp_check_pkey PRIMARY KEY (store_id, date_id, emp_num);
```

```
--
-- PostgreSQL database dump complete
--
```

# Apéndice C

## Procedimiento almacenado para cálculo de horas

Este procedimiento almacenado o *store procedure* lo realicé para el cálculo de las horas totales almacenadas desplegadas en el reporte de “Reporte de Asistencias en biométrico” en eReports.

```
CREATE OR REPLACE FUNCTION total_worked_hours(character varying, character varying, character varying, character varying)
RETURNS character varying AS
$BODY$
DECLARE
    timein1C ALIAS FOR $1;
    timeout1C ALIAS FOR $2;
    timein2C ALIAS FOR $3;
    timeout2C ALIAS FOR $4;
    timein1 timestamp without time zone;
    timein2 timestamp without time zone;
    timeout2 timestamp without time zone;
    timeout1 timestamp without time zone;
    result1 interval;
    result1C varchar(4);
    result2 interval;
    result2C varchar(4);
    result interval;
    finalresult varchar(20);
BEGIN

    IF timein1C != '' THEN
        IF timeout1C != '' THEN
            timein1 := CAST(timein1C as timestamp);
            timeout1 := CAST(timeout1C as timestamp);
            result1 := timeout1 - timein1;
        ELSE
            result1C := '0';
        END IF;
    ELSE
        result1C := '0';
    END IF;

    IF timein2C != '' THEN
        IF timeout2C != '' THEN
            timein2 := CAST(timein2C as timestamp);
            timeout2 := CAST(timeout2C as timestamp);
            result2 := timeout2 - timein2;
        ELSE
            result2C := '0';
        END IF;
    ELSE
        result2C := '0';
    END IF;

    IF result1C = '0' THEN
        IF result2C = '0' THEN
```

```
        finalresult := '';
    ELSE
        result := result2;
        finalresult := CAST(result as varchar);
    END IF;
ELSE
    IF result2C = '0' THEN
        result := result1;
        finalresult := CAST(result as varchar);
    ELSE
        result := result1 + result2;
        finalresult := CAST(result as varchar);
    END IF;
END IF;

    return finalresult;
END;$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;
ALTER FUNCTION total_worked_hours(character varying, character varying, character varying, character varying)
OWNER TO postgres;
```