



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Central de cobro automatizada (Robotina)

INFORME DE ACTIVIDADES PROFESIONALES

Que para obtener el título de
Ingeniero en Computación

P R E S E N T A

Diego Espinosa García

ASESORA DE INFORME

M.C. María Jaquelina López Barrientos



Ciudad Universitaria, Cd. Mx., 2024

Objetivo

- Mostrar mis conocimientos adquiridos a lo largo de mi trayectoria profesional hasta el momento.
- Constatar la importancia de los conocimientos adquiridos en la Facultad de Ingeniería para mi vida profesional.

Marco teórico

El desarrollo de las Tecnologías de la Información y la Comunicación (TIC) ha ayudado a transformar las economías y a la sociedad en general, creando nuevos modelos de colaboración para aumentar la eficiencia de los trabajadores, promoviendo la innovación y mejorando la educación y el acceso a servicios básicos [1]. El uso de las TIC ha llevado a que la información sea manejada a nivel planetario, transformando las esferas sociales, políticas y económicas [2].

Con el paso de los años, el aumento de la población en las ciudades y los avances tecnológicos, las empresas han sido obligadas a actualizar sus procesos y maquinaria para brindar un mejor servicio, aquellas que no han podido adaptarse a las nuevas necesidades de la época han tenido que cerrar.

Gracias a esto, cada vez es más común ver a las TIC en los procesos de casi todas las áreas: educación, medicina, servicios básicos, entre otras. Dejó de ser un lujo de las grandes empresas y productos Premium, para ser una realidad en los diversos estratos sociales y la podemos encontrar a diferentes escalas incluso en PyMEs.

La competencia entre las distintas empresas ha permitido a los usuarios elegir la que mejor se adapte a sus necesidades y, sobre todo, tener la oportunidad de disfrutar de las nuevas tecnologías al menor precio posible.

Ya sea en forma de hardware o software, la tecnología ha revolucionado la vida de los consumidores, en los procesos administrativos, industriales, de entretenimiento y los servicios básicos, por mencionar algunos.

Índice

Índice de figuras	4
Capítulo 1 ANTECEDENTES	5
1.1 La empresa.....	6
1.2 Mi experiencia profesional	9
Capítulo 2 CONTEXTO DE PARTICIPACIÓN PROFESIONAL	11
2.1 Actividades y roles dentro de la consultoría Inventronix.....	12
Capítulo 3 ANÁLISIS Y METODOLOGÍA EMPLEADA PARA LA RESOLUCIÓN DE PROBLEMAS	15
3.1 Metodologías utilizadas en Inventronix.....	16
3.1.1 Metodología Cascada	16
3.1.2 Metodología Prototipado	18
3.1.3 Metodología Incremental	18
3.1.4 Metodología Kanban	19
3.1.5 Metodología Scrum	19
3.1.6 Metodología Scrumban	20
Capítulo 4 PARTICIPACIÓN PROFESIONAL.....	21
4.1 Proyectos realizados	22
4.1.1 CRM App	22
4.1.2 SAM App	25
4.2 Central de cobro automatizada “Robotina”	28
4.2.1 Introducción	28
4.2.2 Definición del problema.....	29
4.2.3 Metodología empleada.....	29
4.2.4 Participación profesional	30
Capítulo 5 RESULTADOS OBTENIDOS.....	53
CONCLUSIONES	56
Glosario de términos.....	57
Referencias.....	59

Índice de figuras

Figura 1.1 Estructura general de la empresa.....	7
Figura 1.2 Jerarquía de la división de lavandería.....	8
Figura 1.3 Jerarquía de la división de tecnología.....	9
Figura 3.1 Pasos de la metodología cascada.....	17
Figura 4.1 Diagrama con los tiempos aproximados del proyecto CRM App.....	25
Figura 4.2 Diagrama con los tiempos aproximados del proyecto SAM App.....	27
Figura 4.3 Diagrama de etapas del proyecto.....	30
Figura 4.4 Diagrama de bloques de los componentes de la Central de Cobro.....	32
Figura 4.5 Comunicación con un solo bus.....	38
Figura 4.6 Comunicación con múltiples buses.....	38
Figura 4.7 Diagrama de conexión del SPO.....	39
Figura 4.8 Información básica del comando.....	40
Figura 4.9 Ejemplo de cadena transmitida.....	40
Figura 4.10 Formato de los parámetros del comando.....	41
Figura 4.11 Diseño de la versión 1.....	42
Figura 4.12 Pantalla principal v2.....	44
Figura 4.13 Elección de tamaño.....	45
Figura 4.14 Pantalla de selección.....	46
Figura 4.15 Pantalla de resumen.....	47
Figura 4.16 Diagrama con los tiempos aproximados de la primera versión del proyecto Robotina.....	51
Figura 4.17 Diagrama con los tiempos aproximados de la segunda versión del proyecto Robotina.....	52

Capítulo 1

ANTECEDENTES

En esta primera sección del informe doy a conocer el contexto de mi situación profesional, desde la presentación de la empresa para la cual estuve trabajando y en la cual realicé el proyecto que se describe a lo largo de este documento, como también de mi experiencia a lo largo de mis años con actividades profesionales y el papel que desarrollé a lo largo de los mismos. La importancia de esta sección radica en dar las bases para comprender las actividades que se describen posteriormente y constatar que el tiempo trabajado fue muy bien aprovechado por ambas partes (la empresa y yo) para el crecimiento de cada uno.

1.1 La empresa

Antes que nada, me permito hablar de la organización para la cual estuve trabajando, y que, por motivos de confidencialidad, desde este punto en adelante me referiré a ella como “Inventronix”.

Inventronix es una empresa que tiene tres divisiones principales:

1. La división de lavandería autoservicio, que cuenta con alianzas con otras empresas del mismo rubro y con más de 25 años de experiencia.
2. La división de importaciones, encargada de la importación de equipos de lavandería, partes, accesorios y *hardware* de distintos tipos.
3. Y la división de tecnología, división en la cual yo estuve trabajando.

Esta última es una consultoría de tecnología con fines de lucro, oficialmente inaugurada en el año 2015. Esta pequeña y aún joven división, está enfocada a ofrecer soluciones tecnológicas tanto a la división de lavandería como a cualquier agente externo que solicite sus servicios, de ser necesario, utiliza la división de importación a fin de traer los materiales necesarios para un proyecto.

La consultoría está dividida en 5 áreas principales:

- Área de desarrollo de escritorio: aquí se lleva a cabo la creación de nuevos sistemas o se les da mantenimiento a los ya existentes. Por ejemplo, Puntos de Venta (*POS*), y sistemas Gestión de Relación con los Clientes (*CRM*), desarrollos con un uso específico, entre otros. Es el área con más peso dentro de la empresa.
- Área de desarrollo web: en sus inicios, sus tareas eran básicas como la creación de *landing pages* con *frameworks* como WordPress, sin embargo, con los años fue evolucionando hasta la creación de aplicaciones web más avanzadas.
- Área de desarrollo móvil: es el área más pequeña y con menor peso de la empresa. Su existencia es temporal, ya que los proyectos que aquí se desarrollan son los menos frecuentes en la consultoría, por lo que los miembros de esta área sólo son contratados o prestados por otras áreas durante el desarrollo de los proyectos en puerta.
- Área de desarrollo de *backend*: es el área más nueva de la consultoría. Hasta el año 2019 aproximadamente, tanto el *frontend* como el *backend* de todas las demás áreas estaban integrados en el mismo proyecto, sin embargo, por diversas razones internas, se decidió abrir esta área para aquellos proyectos más complejos que requieran tener ambas partes por separado.
- Soporte técnico: esta es otra de las áreas más importantes en la consultoría, ya que gran parte de los ingresos vienen de aquí. Brinda apoyo tanto a la división de lavandería como a clientes externos a la organización, el soporte puede ser remoto, presencial, de *software*, *hardware*, etcétera. Esta área cuenta con miembros mixtos, es decir, puede haber miembros dedicados sólo en estas tareas, o hay miembros que realizan de forma paralela tareas tanto en esta área como en otras

En la figura 1.1 se muestran diagramas de la estructura de la organización y de la jerarquía de algunas de sus divisiones.

Para el caso de la División de Lavandería, únicamente se contemplan dentro de ella las sucursales pertenecientes o estrictamente con una relación de “socio” relativa a Inventronix, es decir, que el dueño de la sucursal es total o parcialmente el dueño de Inventronix, la sucursal pertenece a algún familiar o la sucursal es directamente una franquicia vendida por la marca.

En cuanto a la División de Tecnología, está conformada por diversas áreas, cada una con un fin específico descrito en la página anterior.

Finalmente, fundada en el año 2019, la División de Importación es oficialmente la más nueva y la más pequeña de todas. Dado que no trabajé directamente con dicha área, desconozco por completo su estructura interna, así como la jerarquía de su personal, sin embargo, al trabajar indirectamente en algunas ocasiones con ella, considero pertinente mencionarla.

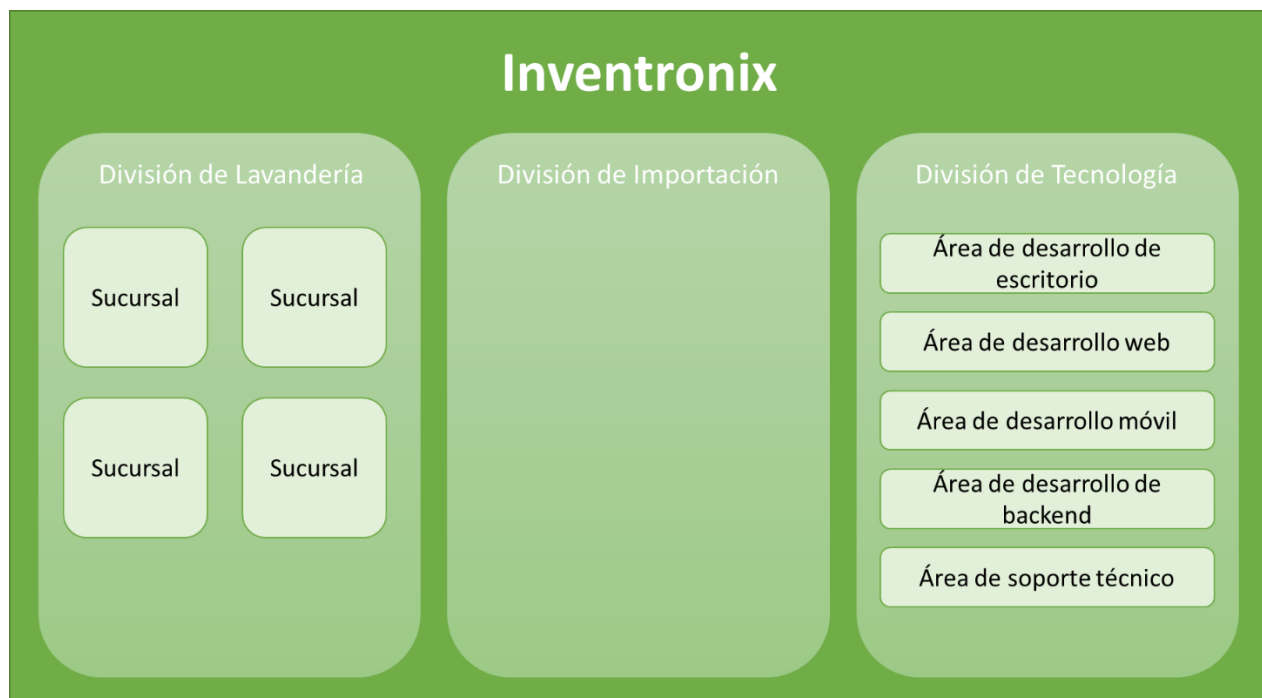


Figura 1.1 Estructura general de la empresa.

La figura 1.2 muestra la estructura jerárquica de la División de Lavandería. Comenzando con el director de la misma quien toma las decisiones con la información proporcionada por su equipo de gerentes, correspondientes a las áreas de operaciones, relaciones públicas y administración.

El Gerente de Operaciones coordina directamente las actividades de todas las sucursales con la ayuda de los supervisores de cada una de ellas, de modo que está al tanto de los aciertos y errores cometidos por los empleados y resuelve cualquier situación dentro de su jurisdicción.

El Gerente de Relaciones Públicas es el encargado de las alianzas con otras marcas o la venta de franquicias, haciendo cumplir las condiciones necesarias para la misma.

Finalmente, el Gerente Administrativo, coordina las tareas de facturación, nómina, depósitos y demás cuestiones económicas referentes a la operación de las sucursales.

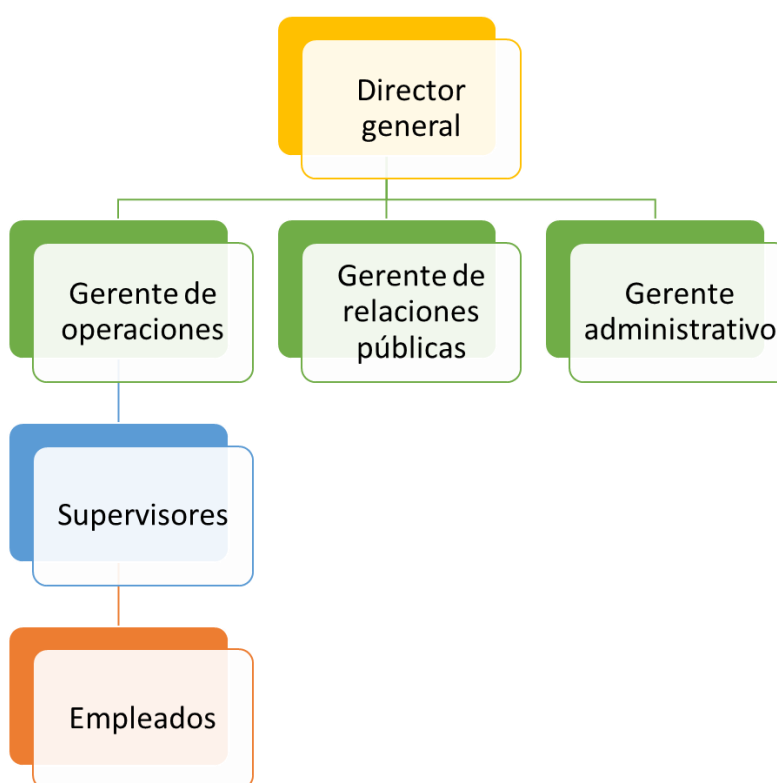


Figura 1.2 Jerarquía de la división de lavandería.

Finalmente, la figura 1.3 muestra la jerarquía del personal de la División de Tecnología. Comenzando con el Director quien recibe directamente las solicitudes de nuevos proyectos y los asigna dentro del área correspondiente, además es el encargado de revisar el progreso y administrar los recursos de la consultoría.

Los proyectos de desarrollo recaen en las manos de los líderes de proyecto, encargados de organizar (si es el caso) a su equipo de desarrolladores con las tareas a llevar a cabo, también revisan personalmente el

correcto funcionamiento del *software* y/o *hardware* en todas sus etapas. En ocasiones, el equipo de desarrollo puede incluir entre sus miembros a becarios con tareas menores y con el fin de entrenarlos y ver si tienen las capacidades necesarias para ser contratados formalmente por la empresa.

En cuanto a las tareas de soporte, son asignadas por el director en situaciones específicas, sin embargo, en su mayoría son recibidas directamente por el cliente y atendidas sin escalar a un nivel jerárquico superior. Esta área concentra la mayor parte de los becarios, ya que, a diferencia de las áreas de desarrollo, tras una breve capacitación, se pone directamente a los becarios a atender las situaciones cotidianas del área, teniendo como apoyo en caso de ser necesario, a un ejecutivo experimentado del área.

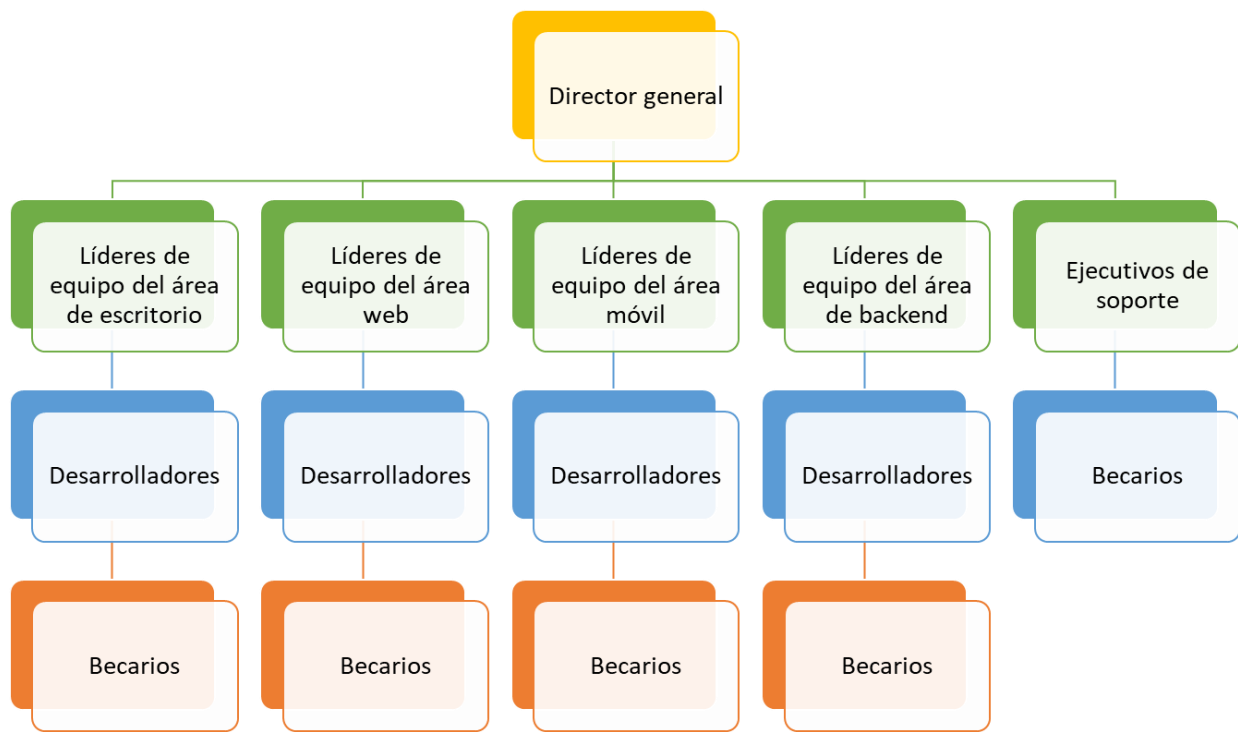


Figura 1.3 Jerarquía de la división de tecnología.

1.2 Mi experiencia profesional

Comencé a trabajar formalmente desde los 17 años en modo intermitente, es decir, únicamente trabajaba una temporada al año durante mis vacaciones escolares. Los empleos que fui tomando eran actividades básicas como *staff* en un parque de diversiones, *staff* de una cadena de cines o repartidor de plataformas de alimentos, que si bien no tiene nada que ver de forma directa con mi carrera, sí me dieron conocimientos importantes como responsabilidad, formas de tratar con clientes y compañeros, trabajo en

equipo, puntualidad, entre otros, conocimientos y valores que posteriormente me ayudarían a ingresar de forma más sencilla como becario en la consultoría “Inventronix”, y una vez dentro también me ayudarían a crecer dentro de la empresa y junto con eso, ayudar a la empresa a crecer.

Pese a que fue mi primer trabajo enfocado en mi ámbito profesional y mis conocimientos en la materia eran relativamente cortos ya que ingresé a trabajar aproximadamente en 5° semestre de la carrera, lo aprendido hasta el momento entre el bachillerato y mis semestres en la Facultad de Ingeniería me dieron las habilidades suficientes para desempeñar mi trabajo como becario y en conjunto con el apoyo que me brindó la empresa para expandir mis conocimientos, fui escalando en la jerarquía hasta llegar a líder de equipo de desarrollo. Los detalles de mi rol dentro de la empresa y más específicamente de las actividades que desarrollé en el proyecto “Robotina”, son descritos en las secciones posteriores.

A pesar de que tuve que salir de la empresa debido a situaciones ajenas a ambas partes (disminución de actividades laborales debido a la pandemia), los conocimientos adquiridos son invaluable. Posterior a mi salida trabajé como *FreeLancer* en algunos proyectos web y de *backend* y todo esto en conjunto me ayudó a ingresar a una reconocida institución como prestador de servicio social.

Capítulo 2

CONTEXTO DE PARTICIPACIÓN PROFESIONAL

En la presente sección brindo un contexto general de mis actividades dentro de la consultoría Inventronix. Partiendo desde mi ingreso a la misma hasta las últimas actividades que llevé a cabo en ella, explicando a grandes rasgos las tareas que desempeñé en cada uno de los roles que ocupé en la empresa, así como algunas tecnologías o lenguajes de programación que utilicé para la realización de las mismas. El objetivo de este capítulo es explicar las actividades más relevantes que desempeñé durante toda mi estadía en Inventronix. Es importante mencionar que mis tareas dentro de la empresa no se limitaron a las descritas en este informe.

2.1 Actividades y roles dentro de la consultoría Inventronix

Ingresé a trabajar a Inventronix en septiembre de 2017, en aquel momento iba aproximadamente en 5° semestre de la carrera de Ingeniería en Computación y no tenía nada de experiencia en este campo laboral, por lo tanto, al presentarse una oferta como becario para esta empresa no dudé en tomarla, ya que mi idea siempre había sido adquirir experiencia profesional antes de terminar la carrera, para tener mayores habilidades y mejores probabilidades de desarrollarme profesionalmente de forma exitosa al salir de la Facultad de Ingeniería.

Acudí a mi entrevista tranquilamente, ya que, si bien las actividades serían totalmente diferentes a lo que había hecho hasta el momento, el haber trabajado antes me forjó una seguridad para enfrentarme ante lo desconocido, el proceso no fue tan largo y a los pocos días ya estaba trabajando como becario.

Al ingresar me dijeron que yo no estaría fijo en un área, ya que como becario y al ser mi primer trabajo dentro de este campo laboral, me estarían rotando para ver en qué área me desempeñaba mejor y podría ser más útil para la empresa. Aquello me benefició demasiado, ya que gracias a eso ahora tengo diversos conocimientos en distintas áreas, que me han servido para proyectos y trabajos posteriores.

Las primeras actividades que realicé fueron para el Área de Desarrollo Web. Elaboré diversos sitios básicos, en general eran sitios web para presentar empresas o servicios, para dichas actividades se utilizaba WordPress. WordPress es un sistema de gestión de contenido de código abierto con licencia GPLv2 (*General Public License* versión 2) [3]. Es un *framework* que nos permite crear sitios web de forma rápida y sencilla, tiene una interfaz muy intuitiva para usuarios que, si bien ya tienen cierto conocimiento sobre el uso de una computadora, no saben nada de programación, otra ventaja es que existen muchos tutoriales y foros en internet que ayudan a manejar más fácilmente esta herramienta. Según w3techs [4], es usado en aproximadamente el 42.9% de todos los sitios web, esto podríamos atribuirlo a la gran cantidad de plantillas, extensiones, *plugins*, entre muchas otras características que integra este *framework*, lo que permite crear desde una simple *landing page*, hasta cosas más complejas como tiendas en línea, mismas que ocupan el 20.2% de los sitios hechos con WordPress.

Si bien WordPress tiene muchas ventajas, también tiene algunas desventajas y esas radican principalmente en la falta de flexibilidad con los diseños de la interfaz gráfica, ya que diseños más específicos ya requieren de conocimientos de programación y herramientas personalizadas, aunque la mayoría de los usuarios se limita a los elementos por defecto o los incluidos en las plantillas.

La razón por la cual se hacía uso de este *framework* en la empresa era que al inicio la consultoría no tenía tantos proyectos de este estilo y los clientes que tenía no buscaban cosas más elaboradas que una simple página para llevar su negocio al mundo digital. Al no haber tantos proyectos ni proyectos tan complejos, el personal de esta área no requería tener conocimientos tan avanzados y con las plantillas que había comprado la empresa, era más que suficiente para llevar a cabo las tareas.

Duré aproximadamente 4 meses en esta área y elaboré al menos 6 sitios web, en varios de ellos tuve que realizar modificaciones personalizadas con código, para lograr una interfaz gráfica tal como la deseaba el

cliente, pero gracias a que había tomado un curso de desarrollo web en la Facultad de Ingeniería, no representó un gran problema. Durante este lapso, aunque mis actividades principales eran las concernientes al Área de Desarrollo Web, de forma paralela me capacitaron para atender actividades del Área de Soporte Técnico y del Área de Desarrollo de Escritorio.

A partir de este punto y hasta mi salida de la empresa en abril del 2021, adicional a mis actividades del área en la que estuviera trabajando, siempre llevé de forma paralela actividades de soporte técnico, algunas de las cuales están descritas a continuación:

- Dar mantenimiento preventivo y correctivo a equipos de cómputo de todas las divisiones de la empresa y también de clientes externos que solicitaban el servicio. Respecto a *software*, estaba a cargo de la instalación, actualización, configuración y desinstalación de programas, formateo o restauración de los sistemas operativos, verificar que los equipos se encuentren libres de virus y en caso de encontrar uno, eliminarlo lo antes posible o en su defecto sustituir la computadora por otra del inventario, para no obstruir las actividades de operación mientras se solucionaba el problema en la oficina. En cuanto a *hardware*, me encargaba de la limpieza de los equipos, sustituir los discos duros que presentaban problemas y recuperar la mayor cantidad de información posible de ellos, la instalación o revisión del correcto funcionamiento de monitores, impresoras, escáneres RFID y demás periféricos que fueran necesarios.
- Con base en los diseños existentes de los circuitos de automatización de equipos de lavandería, me dedicaba a la construcción, reparación, pruebas e instalación de dichos elementos. Esta actividad era una de las más importantes porque está muy ligada al proyecto de la Robotina, ya que si bien la central es el cerebro y se encarga de ordenar a las terminales que realicen cierta tarea, estos circuitos son el medio que logra la comunicación entre los equipos de lavandería y la central. En este apartado llevé a cabo la automatización desde cero de 6 lavanderías con al menos 24 equipos cada una y el mantenimiento de la automatización de otras 7 lavanderías en 4 estados diferentes.
- Capacité al personal de las lavanderías donde llevé a cabo la automatización, así como el seguimiento de cualquier problema que llegara a surgir con el *hardware* o *software* de la misma. Además, capacité al personal de otros proyectos, algunos de ellos elaborados por mí mismo, como el caso de una aplicación móvil y la misma Robotina. Todo esto en 3 estados diferentes.
- En un principio monitoreaba y enviaba informes de errores de los sistemas (*hardware* y/o *software*) al área correspondiente para su corrección, sin embargo, tiempo después y al yo ser parte de las áreas de desarrollo, estaba encargado de corregir los errores reportados por los usuarios.
- Adicional a las actividades anteriores, apoyaba al área operacional y administrativa de las lavanderías.

La siguiente área en la que me capacitaron fue la de Desarrollo de Escritorio, misma donde trabajé la mayor parte de mi estadía en Inventronix. Cuando recién ingresé al área, todo desarrollo de escritorio era realizado con Delphi, debido a que los primeros sistemas que desarrolló la empresa y que aún en la

actualidad siguen existiendo se elaboraron con este lenguaje. Delphi es un lenguaje de programación de alto nivel, compilado y fuertemente tipado que admite el diseño estructurado y orientado a objetos, se creó con el propósito de agilizar la creación de *software* basándolo en una programación visual. En Delphi se utiliza una versión más actual de Pascal conocida como Object Pascal como lenguaje de programación [5]. La historia de Delphi data de 1995 cuando salió su primera versión a manos de Borland, el llamativo que tenía era su enfoque de programación orientada a objetos y basada en formularios, un compilador de código nativo extremadamente rápido y una estrecha integración con Windows [6], entre otros. Y pese a que ya no suele ser un lenguaje muy común, aún sigue siendo utilizado y continúan lanzándose versiones nuevas de él, con un IDE (*Integrated Development Environment*) cada vez más robusto, compatibilidad multiplataforma y con una comunidad sumamente activa.

Mis primeros días en el Área de Desarrollo de Escritorio me enseñaron los aspectos básicos para trabajar con este lenguaje, desde el uso del IDE hasta algunos aspectos básicos del lenguaje como la sintaxis, estructura del código y reglas de negocio para trabajar con él. No pasó mucho tiempo antes de que me asignaran mi primer proyecto, y a pesar de que estuve asesorado a lo largo del mismo yo era el responsable del resultado final, el *software* era un juego “memorama” que una empresa farmacéutica usaría en su *stand* durante una conferencia para dar a conocer sus productos.

Los siguientes 6 meses aproximadamente estuve realizando diversos proyectos como el anterior, que básicamente eran para empresas que tendrían *stand* en alguna exposición. Sin embargo, también realicé algunos programas con mayor relevancia, como uno que importaba o exportaba listas de productos de un archivo Excel hacia una base de datos relacional SQL server o viceversa, mismo que posteriormente se utilizaría como base para integrarlo de forma definitiva en los sistemas principales de la empresa (POS y CRM). Adicional a lo anterior, realicé varias modificaciones, optimizaciones y ayudé a ampliar las funcionalidades de los principales sistemas que la empresa tenía en este lenguaje.

Para este punto, mi estadía como becario ya llegaba a su fin y la empresa decidió contratarme formalmente como desarrollador, y aunque ya me quedaría fijo en esta área, fui prestado en diversas ocasiones al Área de Desarrollo de Aplicaciones Móviles. Como ya se mencionó, ésta es un área que sólo existe durante el desarrollo de algún proyecto, por lo cual estuve pocas veces en dicha área, los dos principales proyectos que llevé a cabo ahí son descritos en el capítulo 4.1 “Proyectos Realizados”.

Los últimos dos años de mi estadía en Inventronix, desempeñé varias veces el cargo de líder para el Área de Desarrollo de Escritorio y también tuve a mi cargo a varios becarios del Área de Soporte Técnico.

En este periodo tuve a mi cargo el proyecto más difícil y grande que haya desarrollado hasta ese momento en mi vida profesional, dicho proyecto es el punto central del presente reporte, ya que participé en todas las etapas de la vida del mismo, las actividades concernientes a dicho proyecto se encuentran descritas en el capítulo 4.2.

Capítulo 3

ANÁLISIS Y METODOLOGÍA EMPLEADA PARA LA RESOLUCIÓN DE PROBLEMAS

En el momento que se presenta algún problema o es necesario realizar la toma de alguna decisión con cierto impacto para el proyecto, el factor más importante siempre a tomar en cuenta es la “gravedad del suceso”, es decir, qué tan radical sería la implementación de cierta característica. Una de las ventajas más importantes de esta consultoría, es que siempre tomaban en cuenta las opiniones de los colaboradores (desarrolladores, diseñadores, *testers*, entre otros) al respecto de los proyectos y si alguna de nuestras ideas mejoraba al diseño original o si nuestras propuestas de solución eran óptimas, se llevaban a cabo, por lo tanto, si la complejidad de la característica era de mínima a media, yo decidía si llevaba a cabo el desarrollo de la implementación, cambio en la interfaz o resolución de un problema y a la hora de presentar los avances, exponía la razón por la cual consideré pertinentes las acciones realizadas. Para los casos en que la gravedad de los cambios era alta, directamente solicitaba la asesoría de mi jefe inmediato a quien le presentaba el problema y las propuestas al cliente o en su defecto para que él mismo decidiera el camino que se seguiría.

Esta forma de trabajar me resultó más que exitosa y me ayudó a adquirir habilidades de investigación, mejorar mi capacidad de análisis y sobre todo me ayudó a adquirir confianza con mi trabajo. Por parte de la empresa, me ayudó a destacar de los demás desarrolladores o incluso de otros líderes de proyecto, ya que filtraba al máximo los problemas o decisiones que iban surgiendo y únicamente acudía por asesoría a los jefes inmediatos en casos donde fuera estrictamente necesario. Debido a eso, cada vez tenía mayores responsabilidades y confianza por parte del personal con mayor jerarquía, razón por la cual mi estadía en esta empresa fue mayormente tranquila ya que tenía ciertas libertades a la hora de tomar decisiones o coordinar a mis subordinados.

3.1 Metodologías utilizadas en Inventronix

Aproximadamente cuando me encontraba cursando entre sexto y octavo semestre de la carrera de Ingeniería en Computación, y como parte de mi crecimiento dentro de la empresa, comencé a formar parte de los equipos de desarrollo de *software* y con ello a tener un contacto directo con las metodologías de desarrollo.

Hasta el momento, mi experiencia había sido en proyectos totalmente escolares, sin embargo, gracias a esos conocimientos adquiridos en la Facultad de Ingeniería, me fue fácil entender las distintas metodologías que utilizaba Inventronix como parte de sus proyectos, incluso, en la materia de Ingeniería de *Software* aprendí metodologías que aún no aplicaba la empresa y al notar que se adaptaban muy bien a ciertos proyectos de la misma, la sugerí a mi jefe inmediato y no sólo fue aceptada, sino que varias de las ideas que aporté gracias a lo que aprendí ese semestre, son las más usadas actualmente para los proyectos.

A continuación, me permitiré hablar de las metodologías más comunes dentro de Inventronix, de las cuales ya tenía conocimientos teóricos, pero fue hasta este punto de mi vida que pude ponerlas en práctica profesionalmente.

3.1.1 Metodología Cascada

Es una metodología tradicional en la que las etapas se organizan de arriba abajo, en ella existe un riguroso orden en la realización de las funciones, tal como se muestra en la figura 3.1. Antes de cada etapa se debe revisar el producto para ver si está listo para pasar a la siguiente fase. Los requisitos y especificaciones iniciales no están predispuestos para cambiarse [7]. Dentro de la empresa, esta metodología es usada principalmente para los proyectos de desarrollo de *landing pages*, ya que una vez presentado y aceptado el diseño del sitio, ya no será necesario ningún cambio y es fácil predecir y cumplir los tiempos de realización de cada tarea.



Figura 3.1 Pasos de la metodología cascada.

A continuación, se muestra una breve explicación de cada una de las fases de esta metodología [8].

Análisis

Comienza con la toma de requerimientos para el *software* y la creación de un estudio de viabilidad, que incluye los costos del proyecto, la rentabilidad y la factibilidad del mismo. El resultado de esta fase es una descripción general de los requisitos, un plan y estimación financiera del proyecto y una propuesta para el cliente. Posteriormente se hace una definición detallada de los requisitos, donde se describe la problemática como tal y las funciones y características que debe ofrecer el producto para cumplir con las exigencias del cliente. Finalmente, se hace un análisis de la definición de los requisitos en el que los problemas complejos se dividen en pequeñas tareas secundarias y se elaboran las correspondientes estrategias de resolución.

Diseño

Sirve para formular una solución específica en base a las exigencias, tareas y estrategias definidas en la fase anterior. En esta fase, los desarrolladores de *software* se encargan de diseñar la arquitectura de *software*, así como un plan de diseño detallado del mismo.

Implementación

Incluye la programación del *software*, la búsqueda de errores y las pruebas unitarias. Los diversos componentes se desarrollan por separado, se comprueban a través de las pruebas unitarias y se integran poco a poco en el producto final.

Prueba

La fase de prueba incluye la integración del *software* en el entorno seleccionado. Los productos de *software* se envían en primer lugar a los usuarios finales seleccionados en versión beta. Las pruebas de aceptación desarrolladas en la fase de análisis permiten determinar si el *software* cumple con las exigencias definidas con anterioridad. Aquellos productos de *software* que superan con éxito las pruebas beta están listos para su lanzamiento.

Servicio

Una vez que la fase de prueba ha concluido con éxito, se autoriza la aplicación en producción del *software*. La última fase del modelo en cascada incluye la entrega, el mantenimiento y la mejora del *software*.

3.1.2 Metodología Prototipado

Se basa en la construcción de un prototipo de *software* que se construye rápidamente para que los usuarios puedan probarlo y aportar *feedback*. Así, se puede arreglar lo que está mal e incluir otros requerimientos que puedan surgir [7]. Esta metodología como tal no se utiliza para los proyectos, más bien es una metodología auxiliar para aquellas situaciones donde el cliente no sabe con certeza lo que desea, por lo cual la consultora le presenta algunos prototipos basados en los requerimientos ambiguos del mismo y con base en sus comentarios se va dando forma al proyecto real. También es utilizado para probar funciones nuevas de los sistemas sin trabajar directamente sobre los mismos.

3.1.3 Metodología Incremental

En esta metodología de desarrollo de *software* se va construyendo el producto final de manera progresiva. En cada etapa incremental se agrega una nueva funcionalidad, lo que permite ver resultados de una forma más rápida en comparación con el modelo en cascada [7]. Esta metodología es de las más comunes en Inventronix y es utilizada principalmente para agregar o modificar funciones a los sistemas ya existentes, es especialmente útil cuando se requieren varias funciones en el menor tiempo posible, ya que

se trabaja una a una las funciones de forma independiente para hacer un *release* en cuanto esté probada una función y de esta forma los usuarios del *software* puedan utilizar dichas funciones mientras se trabaja en el resto.

La metodología incremental no necesariamente es rígida, es decir, que puede adaptarse a las características de cualquier tipo de proyecto, existen al menos 7 fases [9]:

1. **Requerimientos:** En esta fase se definen los objetivos centrales y específicos que persigue el proyecto.
2. **Definición de las tareas y las iteraciones:** El siguiente paso es hacer una lista de tareas y agruparlas en las iteraciones que tendrá el proyecto. Cada una debe perseguir objetivos específicos que la definan como tal.
3. **Diseño de los incrementos:** se necesita definir cuál será la evolución del producto en cada uno de ellos. Cada iteración debe superar a la que le ha precedido. Esto es lo que se denomina incremento.
4. **Desarrollo del incremento:** se realizan las tareas previstas y se desarrollan los incrementos establecidos en la etapa anterior.
5. **Validación de incrementos:** al término de cada iteración, los responsables de la gestión del proyecto deben validar los incrementos que cada una de ellas ha arrojado. Si no son los esperados o si ha habido algún retroceso, es necesario buscar las causas de ello.
6. **Integración de incrementos:** una vez son validados, los incrementos dan forma a lo que se denomina línea incremental o evolución del proyecto en su conjunto. Cada incremento ha contribuido al resultado final.
7. **Entrega del producto:** cuando el producto en su conjunto ha sido validado y se confirma su correspondencia con los objetivos iniciales, se procede a su entrega final.

3.1.4 Metodología Kanban

Consiste en dividir las tareas en porciones mínimas y organizarlas en un tablero de trabajo dividido en tareas pendientes, en curso y finalizadas [7]. Esta es una metodología ágil no muy usual dentro de los proyectos de la empresa, ya que no existe un control suficiente para los tiempos de desarrollo, es utilizada en proyectos de urgencia baja y que son realizados por los desarrolladores de forma simultánea con otro proyecto de igual o mayor prioridad.

3.1.5 Metodología Scrum

Se itera sobre bloques de tiempos cortos y fijos (entre dos y cuatro semanas) para conseguir un resultado completo en cada iteración. Las etapas son: planificación de la iteración (*planning sprint*), ejecución (*sprint*), reunión diaria (*daily meeting*) y demostración de resultados (*sprint review*). Cada iteración por estas etapas se denomina también *sprint* [7]. Es utilizada por Inventronix en proyectos de gran magnitud

donde la probabilidad de cambios es baja o media y donde el cliente esté disponible para revisar constantemente el desarrollo.

3.1.6 Metodología Scrumban

La premisa inicial de Scrumban es emplear lo mejor de Scrum y Kanban. El flujo de trabajo sigue siendo el mismo de Kanban (etapas relacionadas entre sí), aunque con la inclusión de algunos elementos de Scrum como las reuniones diarias de 15 minutos entre el grupo de trabajo y el gestor o los análisis retrospectivos para incorporar mejoras al proceso [10]. Esta metodología también es muy usada por la empresa para proyectos de diversas magnitudes, pero, sobre todo, para aquellos donde la cantidad de cambios a lo largo del desarrollo son grandes, tiene una gran flexibilidad con las tareas al mismo tiempo que limita su tiempo de realización, se adapta fácilmente a los cambios y mantiene un estatus preciso del proyecto. Como ya se ha mencionado, Scrumban combina aspectos tanto de Scrum como de Kanban, a continuación, se listarán algunas de las características que hacen de esta metodología una gran opción [11]:

Funciones de Scrum en Scrumban

- El trabajo se prioriza según la complejidad de la tarea y la demanda del producto.
- El equipo entero se alinea con respecto al significado de “trabajo terminado”. De este modo, los resultados finales quedan claramente plasmados y definidos.

Funciones de Kanban en Scrumban

- Se trabaja con un conjunto claro de tareas que se deben realizar. Una vez que un miembro del equipo empieza a trabajar con una tarea, la extrae del *backlog* y la ingresa a su propia carga de trabajo actual.
- Hay límites estrictos con respecto a la cantidad de tareas en progreso, para evitar el exceso de trabajo en el equipo.
- Por lo general, las tareas se representan con tarjetas que avanzan a través de diferentes etapas del proceso en un tablero Kanban.

Funciones exclusivas de Scrumban

- En Scrumban no hay jerarquías. Es decir, todos los integrantes del equipo de desarrolladores tienen las mismas oportunidades para elegir o tomar decisiones. Además, el grupo no tiene un líder claro. Más bien, el equipo se gestiona solo.
- Los proyectos Scrumban no tienen que tener necesariamente un vencimiento. Los *sprints*, por lo general, se trabajan en períodos incrementales de dos semanas, para que los miembros del equipo se puedan centrar en las tareas de cada *sprint* específico, hasta que llegue el momento de la revisión y de reiterar.

Capítulo 4

PARTICIPACIÓN PROFESIONAL

Este es el capítulo más importante del informe, ya que en él se detallan los principales proyectos que realicé en mi trayectoria profesional con Inventronix. Comenzando con proyectos que, si bien tienen una magnitud y dificultad considerables, presentaron un menor reto general contra el proyecto “Central de Cobro Automatizada”, el que ha sido mi más grande e importante proyecto hasta la fecha. Es importante mencionar que los proyectos secundarios sólo representan una pequeña parte de los proyectos totales que realicé en mi estadía con Inventronix, fueron seleccionados por tener una complejidad suficiente y a modo de introducción para el capítulo del proyecto principal que es la Robotina.

4.1 Proyectos realizados

A continuación, se describen dos proyectos realizados dentro de mi estancia en Inventronix, son proyectos que, si bien tuvieron cierto grado de dificultad, se eligieron como contenido complementario debido a que mi participación en ellos a nivel global fue menor que en el proyecto principal, pero lo suficientemente relevantes como para sobresalir de mis demás funciones en la empresa.

4.1.1 CRM App

Introducción

Esta es una aplicación para dispositivos Android, la aplicación funciona como un complemento al sistema de escritorio CRM, existente en la empresa desde antes de mi ingreso. Fue solicitada directamente por uno de los clientes que usa dicho sistema para cubrir necesidades específicas, el contexto es el siguiente: el cliente es una empresa de lavandería industrial presente en los estados de Yucatán y Quintana Roo, brindan atención principalmente a Hoteles, Moteles y Hostales, por lo cual diariamente todos sus choferes toman rutas específicas para visitar uno a uno a sus clientes, recolectando la mercancía en un inicio y más tarde entregando mercancía del día anterior.

El problema es que debido a los altos volúmenes que manejan, lo apresurada de las tareas de los choferes y lo complejo del proceso en las instalaciones, mantener los pedidos con notas escritas era muy difícil, ya que constantemente se perdían las notas, no había una forma de saber el estatus en tiempo real de los pedidos y los tiempos de comunicación entre el chofer y la planta eran muy tardados, es decir, la planta no sabía la cantidad de volumen que tendrían que atender sino hasta que el chofer terminaba toda la ruta, misma que podía tardar más de tres horas y al llegar los empleados en la planta no estaban preparados para manejar todo el volumen.

Participación profesional

Mi participación en este proyecto comenzó después de la toma de requerimientos al cliente, analizando dichos requerimientos y elaborando propuestas del diseño de la interfaz gráfica para la aplicación, tomando en cuenta que el *framework* utilizado por la empresa para el desarrollo de aplicaciones móviles es FireMonkey. FireMonkey es una plataforma de desarrollo incluida en el IDE RAD Studio, compatible con los lenguajes de programación C++ o Delphi y está diseñada para construir aplicaciones nativas multiplataforma [12]. Sus principales ventajas, sobre todo para la empresa es que es muy similar a programar en Delphi para escritorio, por lo cual, cualquiera de los desarrolladores de escritorio es capaz de realizar aplicaciones móviles con este *framework* y el código en su mayoría es compatible para ser ejecutado en Android, iOS o Windows salvo por algunas excepciones, para las cuales existen formas específicas de hacer que se ejecute determinado bloque de código según el sistema operativo.

Ya con el diseño aprobado, procedí a realizar el desarrollo de la aplicación, tomando en cuenta los dos tipos de usuarios definidos para ella con sus respectivos permisos cada uno (Administrador y chofer).

La aplicación debía utilizar la misma base de datos en la nube que existe para el CRM de escritorio, pero las consultas debían ser más rápidas y ser ejecutadas directamente por el servidor para compensar la baja potencia de un dispositivo móvil, por lo cual se utilizaron en su mayoría *Stored Procedures*.

Debido a que el cliente no pagó por el código, sino que iba a pagar rentas mensuales de la aplicación, Inventronix me solicitó que ideara una forma para desactivar la aplicación en caso de falta de pago, así como controlar los dispositivos que hacían uso de la misma, mi solución para ello fue hacer que el sistema se conectara a una base de datos diferente al abrir la aplicación, ya que ligué cada licencia al número de serie específico de cada dispositivo o en su defecto al IMEI, de forma que no se pudiera utilizar más de una vez cada licencia y sólo los dispositivos registrados pudieran hacer uso de la misma. Dicha solución que implementé cumplió a la perfección su trabajo y tuvo tan buen recibimiento por parte de la consultoría que posteriormente se utilizó el mismo método en otros sistemas.

Las funciones que pueden realizar cada tipo de usuario son:

- Chofer: únicamente puede abrir notas y registrar el contenido de las mismas (No le muestra el total de la nota), después de eso las notas ya no son visibles para él hasta después de terminar el proceso de lavado, donde las notas vuelven a estar disponibles. Para este punto el chofer utiliza la aplicación para cargar la camioneta con la mercancía de cada uno de los clientes en su ruta y notifica que ya comienza su trayecto, una vez que entrega la mercancía a cada cliente puede cambiar el estatus de la nota a entregado. Otra funcionalidad que se tiene es registrar algún problema con el pedido, como falta de prendas, prendas mal lavadas o rotas, entre otras.
- Administrador: tiene las mismas funciones del chofer, pero puede ver la nota en cualquier momento sin importar el estatus del pedido, es capaz de borrar o agregar productos aún después de tomada la nota, puede reimprimir *tickets*, ver un historial de pedidos, ver los reportes de problemas levantados por los choferes y además a estos usuarios si les aparece el total monetario de cada nota.

Además de las funciones del *software*, la aplicación debía conectarse a una impresora bluetooth externa para imprimir el *ticket* del pedido cada vez que el chofer recogía la mercancía del cliente, para que el cliente pudiera tener una garantía de las prendas que daba y el estado de las mismas y así notificar cualquier problema a la hora de la entrega.

Los dispositivos utilizados para este proyecto son:

- Tablet Samsung Galaxy Tab A6: se eligió este dispositivo por tener un tamaño ideal para los choferes, tanto para cargarla como para trabajar de forma óptima a la hora de levantar pedidos. Además, por tener la capacidad de llevar una tarjeta SIM y de esta forma poder realizar llamadas y acceder a internet vía datos y como último punto por su costo relativamente bajo [13].

- Impresora térmica portátil Bixolon SPP-R200III: esta impresora se seleccionó por su tamaño compacto, resistencia al polvo y al agua IP43, larga duración de la batería, varias opciones de conectividad inalámbrica y por su rápida velocidad de impresión, de forma que no interfiriera con los tiempos del chofer [14].
- Celular con impresora térmica integrada SUNMI V1 y V2: aproximadamente 4 meses después de la implementación de la aplicación, se decidió sustituir la Tablet y la impresora anterior por un solo dispositivo para reducir costos. Afortunadamente, no implicó cambios mayores en el *software*, únicamente se cambió el módulo de impresión ya que la impresora que tiene integrada este dispositivo utilizaba una forma de impresión más compleja y con menos funcionalidades, además de una velocidad de impresión menor [15].

Una vez terminada la aplicación, mis actividades profesionales en el proyecto continuaron con la capacitación presencial de los administrativos y los choferes en una de las sucursales que el cliente tiene en Quintana Roo. Posterior a ello realicé algunas modificaciones en las funcionalidades, o el diseño, y le agregué más características con base en las necesidades del cliente hasta que finalmente el proyecto ya no requirió atención.

Diagrama de Gantt

En la figura 4.1 se muestra una aproximación de los tiempos del proyecto, en todo el proceso no hubo complicaciones con el mismo.

Durante un par de semanas se analizaron los requerimientos del cliente, ya que el proyecto surgió por petición del cliente, pero el código sería completamente de la consultoría y estaría disponible para cualquier empresa que estuviera dispuesta a pagar una renta mensual por el uso de este. Una vez estudiados los requerimientos, me dispuse a reunir toda la información y a configurar los entornos de desarrollo necesarios para el mismo.

Las actividades ocurrieron de forma secuencial, diseño del *software*, maquetado y desarrollo, pues cada tarea dependía de su antecesora. A través de todo el tiempo de vida del proyecto se fueron realizando pruebas y corrigiendo errores, pero cerca de la conclusión del mismo, se intensificaron y diversificaron las pruebas mandando llamar a personal que no tuviera relación con el proyecto para tener una noción de la facilidad de entendimiento del mismo por parte de usuarios inexpertos.

Finalmente, el proyecto se implementó en producción entre los meses de junio y julio y se consideró como terminado el proyecto, sin embargo, ocasionalmente se siguió brindando mantenimiento menor al sistema, como cambio de logos, colores, funcionalidades adicionales, entre otras, según los análisis de uso y necesidades de las empresas contratantes.

CRM APP

Año. 2018

Periodo de realizacion. 4 meses

Proyecto sin contratiempos

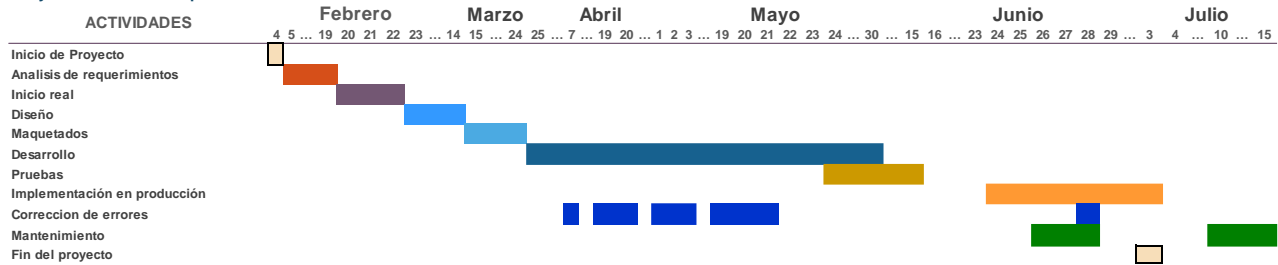


Figura 4.1 Diagrama con los tiempos aproximados del proyecto CRM App.

Conclusión

El proyecto fue entregado en tiempo y forma, la duración desde la toma de requerimientos hasta la capacitación del personal y su implementación fue de aproximadamente 4 meses. Después de un periodo de prueba en una sola sucursal, el cliente decidió extenderla al resto de sus sucursales y la consultoría lo vendió también a otra empresa en la Ciudad de México. En ambos casos hasta la fecha de mi salida en abril del 2021 el *software* seguía operando en su última versión, por lo cual, tanto personalmente como por parte de la empresa se consideró un éxito el resultado final del proyecto.

4.1.2 SAM App

Introducción

SAM es una aplicación para dispositivos Android, solicitada a la consultoría por un cliente dedicado al área de los moteles. En conjunto con una aplicación web forman un sistema cuyo fin es monitorear los ingresos de las sucursales de forma centralizada y verificar que los empleados cobren de forma correcta y no intenten robar a la empresa.

El problema principal por el cual la empresa acudió a nosotros era este último, nos comentó que sus empleados utilizaban las habitaciones sin entregar el dinero, aprovechando que no existía seguridad, o cobraban de forma irregular abusando que la persona encargada de vigilar cada sucursal estaba coludida con ellos.

Participación profesional

Antes de hablar de mi participación en el proyecto, me permitiré brindar un poco más de contexto, ya que como se mencionó, la aplicación está estrechamente ligada a la plataforma web, misma que fue realizada por otra área.

Las funciones que debe cubrir el sistema en conjunto son las siguientes:

- Registrar la renta de una habitación, especificando en la nota el número de habitación, si existe algún descuento aplicado y el tipo de servicio adquirido.
- Elaborar cerraduras especiales con sensores e implementarlas en las habitaciones, de forma que cada que se abra o cierre la habitación mandará una señal vía wifi, la información enviada por el sensor incluye la fecha, hora y habitación del suceso.
- Poder visualizar todo el historial de rentas y de eventos (entradas y salidas de la habitación).
- Poder consultar los videos de vigilancia de las cámaras afuera de cada habitación, para verificar cada uno de los sucesos que no correspondan con las ventas realizadas.

Mi trabajo fue desarrollar la aplicación móvil utilizando el *framework* FireMonkey. La aplicación debe considerar a dos tipos de usuarios con permisos diferentes:

- Vendedor: tras iniciar sesión con su cuenta, este usuario únicamente puede registrar las rentas de las habitaciones. En la nota se coloca la habitación asignada y con base en las políticas del establecimiento se elige el esquema de cobro para dicha habitación, es decir, tiene la posibilidad de elegir el precio regular o aplicar algún descuento disponible.
- Administrador: este usuario no puede registrar renta de habitaciones, lo que puede hacer es ver el historial de ventas y eventos en cada una de las sucursales que tiene asignadas. Existe una sección especial donde puede ver los eventos que requieren atención (esta información ya está filtrada por la aplicación web) y al revisar dichos eventos tiene a su disponibilidad pedazos específicos de video correspondiente a la habitación y el momento preciso en que ocurrió el evento y de esta forma verificar si se trata de un error (falso positivo), un evento válido (se abrió la puerta para recibir algún servicio a la habitación, salir al automóvil y volver, entre otros) o si efectivamente se está realizando algún ilícito.

Una vez terminado el desarrollo de la aplicación, participé en la fase de pruebas, misma que se realizó en conjunto con el Área de Desarrollo Web y para la cual se utilizaron los sensores de las puertas, tanto de forma simulada en la oficina como con las puertas reales en cada sucursal. Adicional a ello, acudí a la instalación de los sensores en cada una de las 30 habitaciones de las 3 sucursales donde se inició la implementación del proyecto.

Diagrama de Gantt

La figura 4.2 muestra una aproximación de la duración del proyecto. Cabe resaltar que durante el tiempo que laboré en Inventronix, jamás se dio por concluido el mismo, puesto que tuvo muchos contratiempos, desde dificultades en el *hardware* a utilizar, hasta los extensos tiempos muertos a causa de la poca atención que el cliente le brindó al proyecto.

La toma de requerimientos se realizó aproximadamente en febrero del 2019 y tras un análisis de alrededor de dos semanas, el inicio real del proyecto fue casi un mes después. Las tareas de análisis de *hardware*, diseño del *software* y el maquetado sucedieron casi de forma secuencial y sin retrasos.

Dado que el cliente solicitó que se le consultara cada movimiento dentro del proyecto para ser aprobado, a partir de este punto comenzaron los retrasos en el mismo, puesto que el primer prototipo de *hardware* presentado al cliente no fue de su agrado y se tuvo que realizar un nuevo análisis de requerimientos y otra búsqueda de *hardware*. El cliente tardó alrededor de medio año en aprobar el *hardware* y los diseños del *software*, por lo cual fue hasta los meses de mayo y junio del 2020, que se llevó a cabo el desarrollo de la aplicación móvil a mi cargo y el sistema web por parte de otro desarrollador, durante ese periodo, el *software* se probó y se corrigieron todos los errores hasta conseguir una versión estable y correcta del mismo.

Formalmente en ese punto concluyó mi participación en el proyecto, pues sólo me habían asignado al desarrollo de la aplicación móvil, sin embargo, en la recta final de mi estadía en Inventronix entre los meses de febrero y abril del 2021, solicitaron mi ayuda para la implementación y pruebas conjuntas de ambos *softwares* y el *hardware* ya instalado en las sucursales de prueba.

SAM APP

Año. 2019-2021

Periodo de realizacion. 2 años

Proyecto con contratiempos

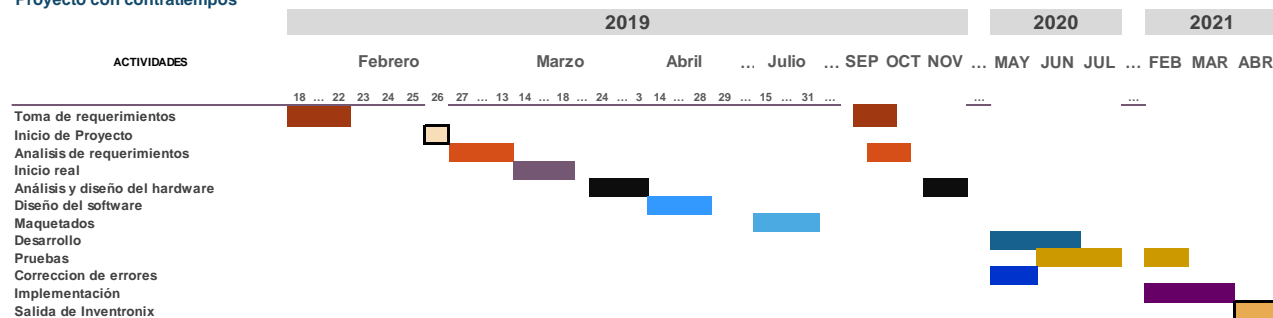


Figura 4.2 Diagrama con los tiempos aproximados del proyecto SAM App.

Conclusión

El proyecto demoró más de lo necesario, ya que a lo largo del mismo el cliente lo iba retrasando puesto que tardaba mucho tiempo en tomar las decisiones o revisar el avance del mismo y estábamos condicionados a esperar su aprobación antes de continuar con las siguientes tareas. Pese a que la aplicación móvil y la web ya se encontraban terminadas y testeadas, los sensores ya estaban instalados y probados en todas las sucursales iniciales y todo estaba listo para la puesta en marcha, debido a la pandemia, el cliente puso en pausa el proyecto dado que se vio en la necesidad de cerrar temporalmente los establecimientos por disposición oficial, casi al mismo tiempo se dio mi salida de Inventronix, por lo cual ya no pude saber si el proyecto continuó y tuvo éxito, tuvo problemas, o si fue cancelado por completo.

4.2 Central de cobro automatizada “Robotina”

Objetivo

- Diseñar y construir una máquina automatizada, capaz de recibir pagos con efectivo y tarjeta de débito y/o crédito para la activación vía puerto serial de equipos de lavandería.
- Diseñar y crear el sistema de *software* que permita a la máquina funcionar con una interfaz amigable e intuitiva para el usuario, deberá almacenar la información en una base de datos relacional (SQL Server) y funcionar por sí sola o en conjunto con un sistema de escritorio (POS).

4.2.1 Introducción

La competencia entre las distintas empresas ha permitido a los usuarios elegir la que mejor se adapte a sus necesidades y, sobre todo, tener la oportunidad de disfrutar de las nuevas tecnologías al menor precio posible.

Para el caso específico de las lavanderías, se ha pasado por pequeñas máquinas operadas de forma manual, el uso de monedas, *tokens* o tarjetas y más recientemente por medio de dispositivos conectados en red en cada equipo o mediante el envío de señales desde un equipo de cómputo en la sucursal. Sin embargo, no todos los propietarios de estos establecimientos tienen la confianza o los recursos para acceder a dichas tecnologías. Y aunque algunas de estas lavanderías ya cuentan con sistemas de escritorio para la administración o incluso la activación de equipos, el tener a un sólo cajero vuelve el proceso mucho más lento, además de la posibilidad del error humano o los altos costos que representa cada empleado adicional dentro de la sucursal.

Se necesita una nueva forma de operar estos establecimientos, algo que reduzca costos de forma significativa, prevenga lo más posible los errores humanos y al mismo tiempo que agilice el proceso tanto para los empleados y administrativos, como para los usuarios que hacen uso de dichos negocios.

4.2.2 Definición del problema

Algunas lavanderías, como es el caso de la empresa que por cuestiones de confidencialidad denominaré “Lavaland”, han tenido tal grado de éxito a lo largo de los años, que la innovación no ha parado. Pero más recientemente todos los intentos de mantener un buen servicio se han visto opacados por la alta demanda y a pesar de tener a dos empleados en un mismo turno, la carga de trabajo es tanta que se cometen errores cada vez con más frecuencia, hay problemas entre los usuarios por los tiempos de espera para ser atendidos y los cortes de caja generan un gran problema por su demora y tiempo perdido.

La empresa para la que trabajé es una consultoría de *software* que brinda servicio entre otros clientes, a lavanderías, el proyecto que se describe a lo largo del informe es el producto real de un desarrollo de *software* solicitado a la empresa por dichos clientes, cuyo fin es disminuir la carga de trabajo a los empleados, aumentar la velocidad de los procesos y generar mayor satisfacción a los clientes.

4.2.3 Metodología empleada

La metodología usada a lo largo de todo el proyecto fue Scrumban, ya que tuve la oportunidad de colaborar con el cliente en todo momento, la facilidad de administrar las tareas del proyecto por medio de un tablero y, sobre todo, debido a los cambios constantes en los requisitos de funcionalidad y de la interfaz gráfica por parte del cliente, por lo que dicha metodología quedó perfecta, además de ser la más usada en la empresa.

La forma como se iban colocando las tareas era progresiva, basada en los avances semanales, el resultado del análisis de cada junta y la disponibilidad de tiempo, personal y requisitos tanto de *hardware* como de *software*. Cada día había una junta pequeña (máximo 10 minutos) para hablar de los progresos, reportar algún problema o retraso y discutir el mejor momento para tener cada recurso. Adicional a ello, cada semana se llevaba a cabo una junta más extensa, para mostrar más a fondo los avances finales hasta el momento e ir presentando el trabajo al cliente para su aprobación, ya que, aunque el presupuesto fue aprobado desde el principio, la forma como se entregó fue parcial, ya que no se quería tener equipo sin utilizarse en la oficina y realizar el gasto desde el principio del proyecto.

El resultado fue más que satisfactorio, tanto el equipo de la consultoría como el cliente, siempre tuvimos conocimiento del estado real del proyecto, las tareas que se iban completando y los tiempos de entrega, lo que nos permitió lanzar el producto en el menor tiempo posible y con la mejor calidad, mitigando la mayoría de los errores antes de pasar a la etapa de pruebas.

4.2.4 Participación profesional

Estuve involucrado en casi todas las etapas del proyecto, mismas que se muestran en la figura 4.3. En ella se muestra el proceso seguido en ambas versiones, mismas en las que participé, podemos observar que los procesos se repiten dos veces por complicaciones de las que se hablará en capítulos posteriores. Colaboré desde el análisis de requerimientos y la elección de los componentes, hasta las pruebas finales y mantenimiento del *software*, posterior a la finalización del mismo. Cabe mencionar que tal como se observa en la figura, el proyecto no se dio por terminado sino hasta que se implementó la segunda versión, es por ello que se describirá el proceso de realización de ambas versiones, ya que en realidad conforman un solo proyecto.

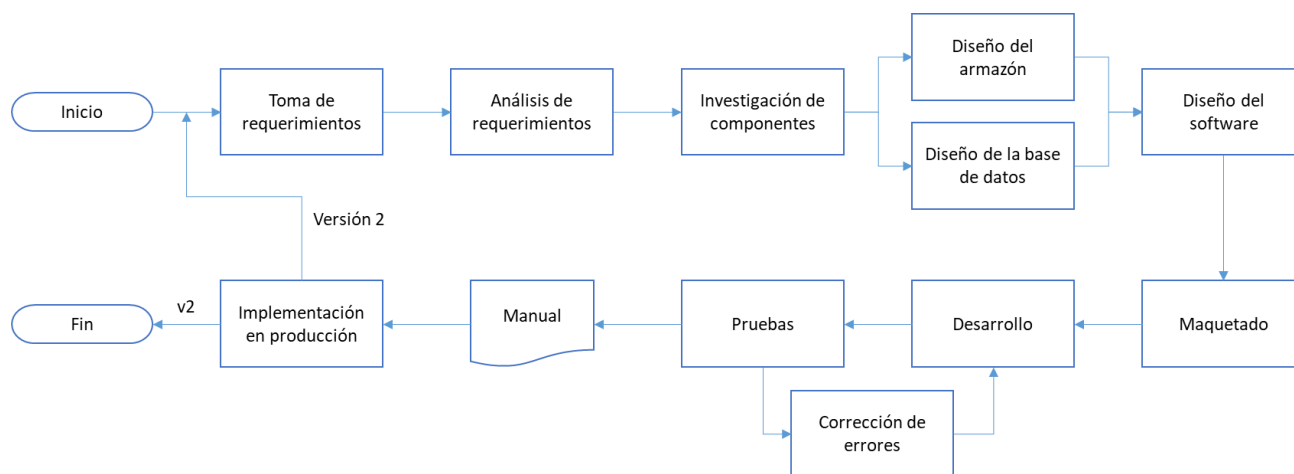


Figura 4.3 Diagrama de etapas del proyecto.

El concepto del proyecto es simple, la máquina es una central de cobro automatizada, capaz de recibir pago en efectivo, puntos de lealtad y tarjetas de débito y/o crédito. Desde ella se pueden elegir los equipos y productos que desea el cliente y posterior al cobro se manda señal a los mismos para su activación, todo esto con una interacción directa entre cliente-máquina que permite concentrarse a los empleados en otras tareas de la sucursal.

En cuanto la consultoría recibió el proyecto por parte del cliente, me pusieron totalmente a cargo del mismo, debido a que llevaba más de un año tratando temas similares, enfocado hacia las lavanderías y puntos de venta. Es importante mencionar que durante todo el proyecto estuve colaborando directamente con mi jefe inmediato, si bien, yo estaba a cargo del desarrollo de todo el sistema, mi jefe estaba a cargo de la construcción del armazón para la central de cobro y entre los dos diseñamos la base de datos. Por lo tanto, es él a quien reportaba mis resultados y con quien discutía problemas, cambios y sugerencias, mismas que en su mayoría fueron aceptadas pues yo estaba involucrado más que nadie en la investigación relacionada al proyecto.

A continuación, me permito hablar brevemente sobre mi jefe inmediato, la intención es dar contexto de la otra persona involucrada en este proyecto, así como dar a conocer las actividades que cada uno realizó.

Es ingeniero mecatrónico, en el ámbito de la empresa desempeña el puesto de Director General y es, además, el fundador de la consultoría Inventronix. Desde antes de graduarse ya estaba involucrado en el mundo de las lavanderías automatizadas, él desarrolló las primeras versiones del punto de venta que comercializa aún en la actualidad la empresa, mismo sistema que adicionalmente sentó las bases para la creación de la Central de Cobro Automatizada.

Los detalles de mis actividades dentro del proyecto se encuentran desglosadas en los siguientes subcapítulos del informe.

4.2.4.1 Investigación y elección de los componentes.

Lo primero que hicimos tras la toma de requerimientos que realizó mi jefe, fue hacer una lista a grandes rasgos de las funciones de la central de cobro. Con este análisis de requerimientos que realizamos, obtuvimos una lista de componentes, misma que utilicé para elegir el *hardware* necesario para armar la máquina y con la que ambos haríamos el diseño de la misma, considerando las medidas del *hardware* elegido, la seguridad de la caja y demás aspectos, por ejemplo, las condiciones en que estará expuesta y las necesidades de ventilación, entre otros.

Así, el *hardware* que desde un principio identificamos que sería necesario es el siguiente, en este punto aún no se habían definido tamaños, marcas ni características, sólo fue una lista de los componentes principales para comenzar la investigación.

- Pantalla: para la interacción con el usuario.
- Escáner RFID: se usa para leer las tarjetas de lealtad.
- Impresora de *tickets*: para entregar un comprobante de compra al cliente.
- Computadora: el cerebro de todo el sistema.
- Circuitos de automatización: para activar los equipos de lavandería. Estos circuitos fueron diseñados y son producidos por la consultoría y han sido usados por muchos años para automatizar equipos de lavandería de diversas marcas.
- Armazón: donde se monta todo, se eligió que fuera metálico por temas de seguridad. Es mandado a hacer con las medidas exactas de los componentes elegidos, por lo cual es único en el mercado.
- Monedero: para recibir monedas.
- Billetero: para recibir el pago con billetes.

Ya con la lista básica de componentes, lo siguiente fue definir las características mínimas que debían tener cada uno de ellos para ser una opción viable, con base en las características que se planeaban para el *software*, como el uso de hilos para el funcionamiento simultáneo de los aparatos de *vending*, la carga

gráfica de la interfaz, tener un tamaño suficiente para ser usado por todo tipo de público, entre otras consideraciones.

Después de eso comenzaría mi investigación entre diversas marcas, modelos y formas, siempre considerando un buen desempeño, costo acorde al presupuesto, puntos de seguridad, entre otras cualidades.

En la figura 4.4 se muestra en un diagrama de bloques la interconexión de todos los componentes de la Central de Cobro, cabe mencionar que, según la versión, algunos componentes no existen o son opcionales.

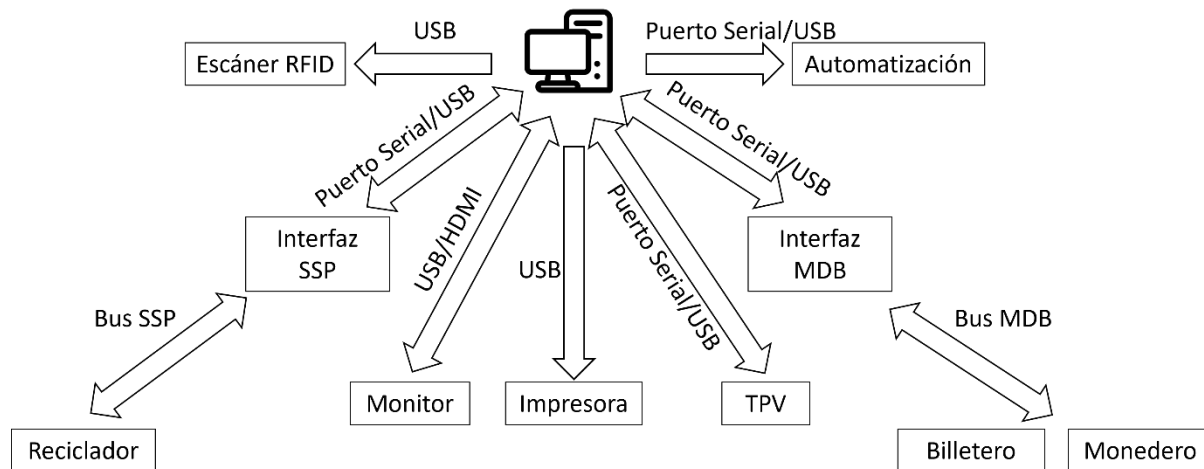


Figura 4.4 Diagrama de bloques de los componentes de la Central de Cobro.

Pantalla

Las cualidades que debe tener son:

- Que sea *touch*, ya que les permitirá a los usuarios interactuar fácilmente con el sistema.
- Alto rendimiento, para mantenerse encendida 24/7.
- Que sea de uso rudo, para tolerar todo tipo de trato por parte de los clientes.
- Gran tamaño para mostrar todo en buena proporción, debido a que una cantidad importante de la clientela de la cadena corresponde a personas de la tercera edad.
- Ligera, para facilitar su colocación vertical sobre muros.

Después de revisar varias opciones en un rango de 5-7 mil pesos, se decidió que la pantalla fuera una ViewSonic TD2220 con las siguientes características:

- Pantalla de 21.5”.
- Resolución *Full HD*.
- Conexión DVI y VGA.
- Bocinas integradas (En un principio se tenía considerado usar el audio).
- Costo aproximado en aquel momento de \$6500.00 MXN

Esta decisión se debió a que cumplía con todos los requisitos.

La pantalla funcionó a la perfección durante el desarrollo y pruebas sin montar (debido a que la fabricación del armazón fue de lo último en terminarse), por desgracia el problema vino en las últimas etapas de la versión 1, ya que nunca imaginamos que el armazón metálico de la Robotina, además del hecho de que se desensambló la pantalla para quitar todos los marcos y la estructura plástica, causaría una completa falla con el *touch* mientras la pantalla estuviera montada en la máquina, lo cual hizo imposible utilizar dicha pantalla en el proyecto.

Este fue el primer gran tropiezo en el proyecto y se debió principalmente a que, para disminuir costos, el cliente decidió optar por una pantalla *touch* ordinaria, en lugar de una pantalla especializada, y a pesar de que nosotros le dimos opciones de pantallas especializadas y argumentos del por qué debía elegir las, la decisión final fue suya, meses después se tuvo que cambiar la pantalla por una de dichas opciones.

La pantalla final fue una de la marca Elo, de 22” y un costo aproximado de \$12 000.00 MXN. Esta es una pantalla con una calidad muy superior, totalmente pensada para ser empotrada en algún armazón, y para un uso rudo como el que se tenía contemplado en el proyecto. Todas las Robotinas que fueron producidas mientras trabajé en Inventronix utilizaron esta pantalla y jamás recibimos reportes de problemas a causa de ellas.

Escáner RFID

Este componente no hubo necesidad de buscarlo, ya que optamos por un escáner que la consultoría venía implementando desde meses atrás con muy buenos resultados, un HID Omnikey 5427ck. Es un escáner que utiliza la tecnología de radio frecuencia baja, está pensado para leer principalmente tarjetas y *tags* de identificación a no más de 10 cm de distancia. La ventaja de la radio frecuencia baja es que funciona cerca de metales y superficies líquidas, además de que no es sensible a la interferencia de ondas de radio [16], lo que lo hace muy confiable y rápido, adicionalmente los *tags* son pasivos, de forma que no necesitan una batería para funcionar, las tarjetas que utilizamos son de sólo lectura y cuentan con un ID único de fábrica, aunque nosotros lo combinamos con otros caracteres para evitar los *tags* no autorizados.

Impresora de tickets

Para este componente se han utilizado varias opciones a lo largo del tiempo que lleva la Robotina en producción, en esta ocasión no se debe a algún problema o error, sino que existió la posibilidad de bajar costos sin perjudicar la calidad.

Las características buscadas por la consultoría para este componente son:

- Bajo costo sin perjudicar la calidad del aparato.
- Alta velocidad de impresión.
- Dimensiones reducidas.
- Con cambio sencillo de rollo de impresión.

La primera versión de Robotina utilizó una impresora térmica TM-T81 de la marca Epson con las siguientes características:

- Impresora de punto de venta para rollos de 80 mm.
- Velocidad de impresión de 130 mm/s.
- Alta confiabilidad y continuamente usada en las sucursales de Lavaland, hasta que fue discontinuada.
- De alto costo y en conjunto con que fue discontinuada, se optó por cambiarla.

La segunda versión de la central de cobro utilizó una impresora de kiosco Zebra KR403:

- Impresora especializada para este tipo de máquinas.
- Compatible con rollos térmicos de 80mm y de una marca con buena calidad.

El problema con esta impresora fue que el aparato como tal cuesta menos que la Epson, sin embargo, no incluye el kit de montaje, sin el cual se presentan continuamente problemas de impresión debido a que se atora el papel, ambos elementos en conjunto ya sobrepasaban el precio de la impresora Epson, por lo cual innegablemente debía ser sustituida.

La impresora que se utilizó de forma definitiva en el resto de las centrales de cobro es una impresora china de la marca Masung, cuyas características son:

- Acepta rollos de 80 mm y está hecha para kioscos y máquinas similares.
- Alta velocidad de impresión 200 mm/s (mucho más rápida que la Epson).
- Tiene la opción de desensamblarse para adaptarse a los espacios de la máquina sin dejar de funcionar.
- Gran calidad de impresión y liberación rápida en caso de atascos.
- Pero su mejor cualidad es que su costo es de la mitad que cualquiera de las otras dos opciones.

Computadora

Las especificaciones mínimas que debe contener la computadora son:

- Medidas reducidas para estar dentro de la central de cobro.
- Suficiente velocidad de procesamiento para ejecutar el sistema de forma fluida y rápida, además, debe ser capaz de tolerar operaciones paralelas, por ello se eligió como procesador mínimo un Intel Core i3 y una RAM base de 4 GB.
- Reducida capacidad de almacenamiento, la computadora únicamente contendrá los programas básicos para el funcionamiento del *software* y el sistema operativo, por ello se desea reducir el presupuesto para este rubro integrando un disco duro de 500 GB o menos.

La computadora elegida es una Dell Micro Optiplex 3070:

- Dimensiones de tan sólo 3.6 cm de ancho, 17.8 cm de profundidad, 18.2 cm de altura.
- Peso reducido de 1.18 kg [17]

Ya que cubre todos los requisitos y adicionalmente cuenta con dos tipos de salidas de video, múltiples puertos USB de alta velocidad, que son de gran importancia para los aparatos que se usan en ella y también incluye un soporte para montar que brinda una rápida liberación.

Esta computadora ha sido usada en todas las versiones de Robotina y jamás ha presentado problemas. En algunas centrales, el cliente eligió reemplazar el disco duro mecánico por uno de estado sólido para aumentar la velocidad de lectura y escritura de datos.

Monedero

Los últimos dos componentes fueron los “más complicados” de elegir, la razón de esto es que ni la consultoría ni yo habíamos trabajado nunca con *vending machines*, además muchos de estos aparatos son sumamente caros.

Tras unos días de investigación, descubrí que existen 3 tipos de monederos: aceptadores de monedas, dispensadores de monedas y recicladores de monedas, los cuales se describen a continuación:

- Aceptadores de monedas: la única función de estos dispositivos es revisar las monedas que pasan a través de ellos, si son válidas las “aceptan” y dejan salir por su dispensador, mandando una señal a la computadora con el valor de la moneda ingresada, en caso contrario son devueltas por la parte frontal inferior sin mandar señal o indicando que la moneda fue rechazada.
- Dispensadores de monedas: como su nombre lo indica su función es contar y/o proveer monedas de un origen externo al usuario, es decir, hay un recipiente ajeno al aparato y cuando recibe la señal de dar cambio deja pasar monedas a través de él según la cantidad indicada.

- Recicladores de monedas: estos aparatos incluyen las características de ambos tipos pasados. Su tarea inicia cuando entra una moneda a través de él, verifica que la moneda sea válida (según la divisa configurada y las monedas de dicha divisa que son aceptadas por el dueño del aparato), si no es válida la devuelve por el dispensador, si la moneda es válida revisa en sus tubos si tiene espacio para almacenar la moneda, en caso contrario la manda por el *cashbox*, en ambos casos manda una señal con la información de la moneda recibida, cuando se le solicita que de cambio (recicle) revisa sus tubos y ya sea de forma automática o manual devuelve el importe solicitado.

De éstos, se identificó que dos de ellos no nos eran útiles, debido a que los validadores no tenían la capacidad de dar cambio, únicamente registraban la moneda que se les introducía y lo que pasaba con ella después. Mientras que los dispensadores no son capaces de validar monedas, únicamente surten una cantidad determinada de ellas que se encuentran en un almacén externo, además del hecho de que sus precios eran elevados. Una vez que se identificó el tipo de dispositivo que se necesitaba, lo siguiente fue decidir la forma como el dispositivo se comunicaría, ya que existen diversos protocolos de comunicación en el mercado, cada uno con sus pros y contras, por ejemplo, *ccTalk*, *Smiley Secure Protocol (SSP)*, *Multi Drop Bus (MDB)*, entre otros.

El monedero elegido para la Robotina es el Mei CF7000, un reciclador de monedas cuyas características son:

- Tiene 5 tubos para almacenamiento de monedas.
- Acepta diversas divisas y es configurable en las denominaciones y la ruta que debe seguir cada moneda (el tubo donde se almacenará o si directamente será enviada al *cashbox*).
- Es compatible con el protocolo MDB.
- Tiene un precio aproximado de \$11 000.00 MXN.

Billetero

Al igual que el monedero, existen 3 tipos de billeteros en el mercado con diversos objetivos entre sí: validadores, dispensadores y recicladores de billetes.

El problema con este componente es que los tipos de billeteros están más bien orientados a diversas gamas de compradores, siendo los validadores los más accesibles y mayormente consumidos por los “pequeños negocios”.

Los otros dos tipos de billeteros están orientados a operaciones de gran magnitud en su mayoría (un flujo de \$30 000 pesos diarios o superiores), como cajeros automáticos, kioscos con muchas transacciones de alto valor, y otros similares. Por lo cual los costos son excesivamente altos en su mayoría para Pymes y existen muy pocas opciones que se puedan adaptar a una máquina de tan pequeña magnitud de transacciones como lo es Robotina. Es por ello que se decidió crear dos versiones de la central de cobro,

cuya diferencia radica principalmente en el billettero que usa, el tamaño del gabinete (a causa de las grandes dimensiones de los recicladores de billetes) y por consiguiente el costo de la misma.

- *Versión lite*: es aquella que únicamente da cambio con monedas, está pensada para negocios con mínimas transacciones y que no requiera dar altas cantidades de cambio, ya que está limitado por la capacidad de los tubos y las monedas introducidas en él. En esta versión se utiliza un validador de billetes Mei CXB2 con capacidad para almacenar hasta 300 billetes de diversas denominaciones, cuenta con un sistema para verificar que el billete introducido sea válido y que no sea falso, se comunica con el protocolo MDB y es totalmente compatible con el CF7000, por lo cual pueden compartir bus de datos, su precio aproximado es de \$11 000.00 MXN.
- *Versión full*: como su nombre lo dice, es la versión más completa al tener la posibilidad de dar cambio tanto en monedas como en billetes, está pensado para negocios con una mayor cantidad de transacciones al día, al poder combinar cambio de ambos tipos tiene una mayor autonomía antes de necesitar ser rellenado, aun cuando las cantidades de cambio sean grandes. En este caso la central monta un reciclador de billetes *SMART Payout* de la marca *Innovative Technology*, es capaz de reciclar hasta 70 billetes de diversas denominaciones (los enrolla en un tambor inteligente que conserva la información del billete para su posterior acceso), además tiene una capacidad de almacenamiento (no disponible para reciclar) de 500 billetes por defecto con un sistema de liberación básico, pero tiene la posibilidad de adquirir accesorios para imprimir *tickets*, ampliar el almacenamiento a más de 1000 billetes y para colocar un método de extracción cerrado con llave [18]. Su precio base es de aproximadamente \$21 000.00 MXN. Es configurable para diversos protocolos de comunicación entre los que se encuentran MDB, ccTalk, SSP, salida en pulsos, entre otros, por medio de interfaces. Debido a su alta velocidad de transmisión de datos (115200 bauds), su longitud de cadena de transmisión mayor y que usaremos el protocolo SSP en lugar de MDB para comunicarnos con él, no es posible compartir bus con el monedero CF7000.

MDB – Multi Drop Bus

El protocolo elegido para utilizar con el monedero fue MDB (*Multi Drop Bus*), es un protocolo utilizado en las máquinas de *vending* para interconectar diferentes módulos, como pueden ser: Monederos (*coin changers*), billetteros (*bill acceptors*) y lectores RFID.

El protocolo MDB transmite las señales por un bus serie que trabaja a 9600 baudios o más en una configuración maestro-esclavo, donde todos los periféricos son esclavos de un controlador maestro. Cada periférico tiene una dirección única y un conjunto de comandos [19]. En este tipo de configuración el maestro debe escanear el bus constantemente para comunicarse con los esclavos, se les hace una solicitud por medio de una cadena de información que puede contener la versión del protocolo, el código de la instrucción, un *checksum* y el dispositivo destino del mensaje (debido a que todos están en el mismo bus) entre otros parámetros según el dispositivo y la configuración del fabricante. Tras la respuesta de los esclavos, el maestro analiza la información y la devuelve a la computadora que se la solicitó, en caso de que no exista una respuesta, el maestro está configurado para repetir la solicitud determinada cantidad

de veces, tras las cuales en caso de persistir la ausencia se le notifica a la computadora que dicho esclavo no está transmitiendo.

Una de las principales razones por la cual elegimos este protocolo es que MDB es el “estándar” internacional, lo que significa que gran parte de los dispositivos *vending machine* en el mercado son compatibles con dicho protocolo y por lo tanto son mayormente compatibles entre sí, lo que permite integrar distintas marcas y tipos de aparatos en un solo sistema, como se muestra en la figura 4.5, además de ser un protocolo con demasiados años en el mercado [20], por lo cual es ampliamente confiable y está más que probada su efectividad y seguridad.

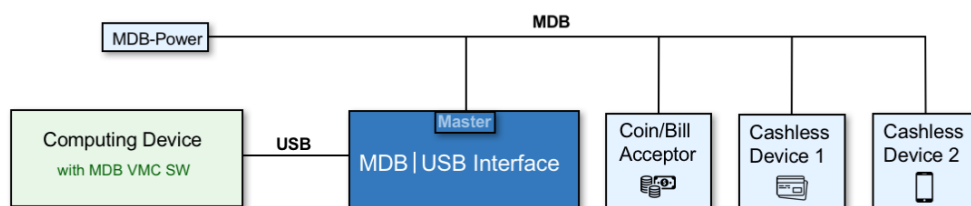


Figura 4.5 Comunicación con un solo bus.

Por desgracia, a pesar de ser un protocolo estándar, la longitud de las cadenas de datos en ocasiones varía, así como también lo hacen las velocidades de transmisión de datos, por lo cual en algunas ocasiones habrá que tener más de un bus de comunicación entre la computadora y los diversos aparatos, tal como se ejemplifica en la figura 4.6.

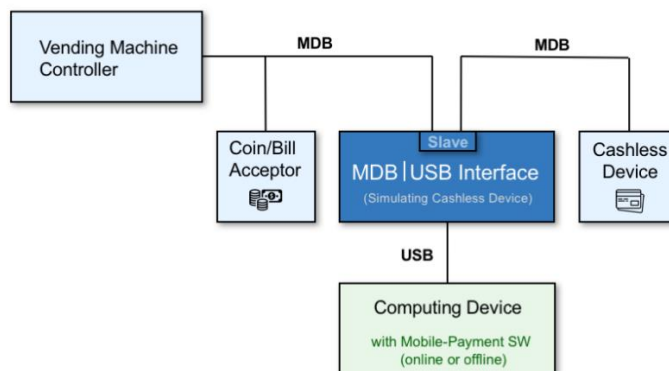


Figura 4.6 Comunicación con múltiples buses.

Otro problema con estas variaciones está relacionado con las características de la señal de transmisión, es que no siempre puede ser realizada de forma directa entre la computadora y el aparato *vending machine*,

en ocasiones es necesario integrar una interfaz MDB que se comunique entre un puerto estándar de la computadora y el aparato *vending machine*, en estas ocasiones dicha interfaz actuará como el maestro. Tal es el caso del monedero CF7000 y el billeteo CXB2 de la marca Mei, mismos que necesitan de una interfaz llamada EasyMDB Universal de la marca Mayavending para alimentarse y comunicarse con la computadora, por un lado el programa manda los comandos por medio del USB con señales serial, llegan a la interfaz que las convierte en instrucciones del protocolo MDB y las manda a los esclavos (monedero y billeteo), la respuesta nuevamente pasa por la interfaz MDB para ser convertida y finalmente recibida por la computadora.

SSP – Smiley Secure Protocol

Smiley Secure Protocol (SSP) es una interfaz segura diseñada específicamente por ITL (*Innovative technology*) que utiliza un modelo maestro esclavo, la máquina host es la maestra y los periféricos (billeteo o monedero) son los esclavos. La transferencia de datos se realiza a través de un *multi-drop bus* utilizando transmisión en serie asíncrona, la integridad de las transferencias de datos se garantiza mediante el uso de sumas de comprobación CRC de 16 bits en todos los paquetes. Cada dispositivo SSP tiene un número de serie único, este número se usa para validar cada dispositivo en la dirección de transferencia antes de que las transacciones puedan tomar lugar. Utiliza un sistema de cifrado para evitar fraudes a través del bus [21].

Para el caso del *SMART Payout (SPO)* también se necesita de una interfaz para comunicarse (viene incluida al comprarlo), sin embargo, requiere de una fuente de poder adicional para funcionar, tal como se muestra en el diagrama de conexión de la figura 4.7. El protocolo de comunicación elegido para este billeteo es SSP, pues está principalmente diseñado para funcionar con dicho protocolo y la comunicación por medio de MDB requiere de una interfaz adicional y los comandos no cambiarían significativamente.

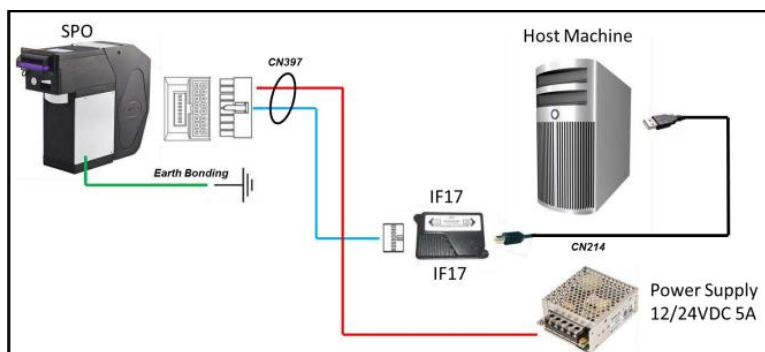


Figura 4.7 Diagrama de conexión del SPO.

Aunque la mayoría de las operaciones con las *vending machine* son fijas y sólo requiere enviarse una cadena igual en cada ocasión, hay casos en los que la cadena se va a construir de acuerdo con los requerimientos de la operación. Por ejemplo, para dar el cambio en billetes, se puede elegir la cantidad de cada denominación que se desea entregar, calculando dichas cantidades y configuraciones en la aplicación y posteriormente traduciendo los cálculos en arreglos de bytes para ser enviados al aparato, las figuras 4.8, 4.9 y 4.10 muestran un ejemplo de esta operación, extraída directamente del manual del fabricante [22]. Al desarrollar la aplicación es muy importante comprender toda la lógica de los comandos, desde la información básica como el nombre del comando y su código, hasta los parámetros completos que debe llevar el paquete, cada comando contiene al menos un ejemplo para comprender mejor este proceso.

Command	Code hex	Code decimal
Payout By Denomination	0x46	70

Implemented on	Encryption Required
NV22, SMART PAYOUT	🔒 yes

Description
A command to payout the requested quantity of individual denominations.

Figura 4.8 Información básica del comando.

Example - A hopper unit has stored 100 x 0.10 EUR, 50 x 0.20 EUR, 30 x 1.00 EUR, 10 x 1.00 GBP, 50 x 0.50 GBP and the host wishes to payout to 4 x 1.00 EUR, 5 x 0.10 EUR, 3 x 1.00 GBP and 2 x 0.50 GBP.

```

Host transmit: 7F 80 27 46 04 04 00 64 00 00 00 45 55 52 05 00 0A 00 00 00 45 55 52 03
              00 64 00 00 00 47 42 50 02 00 32 00 00 00 47 42 50 58 94 B7
ascii:       . . F . . . d . . . E U R . . . . . E U R .
              . d . . . G B P . . 2 . . . G B P X . .
Slave Reply: 7F 80 01 F0 23 80
    
```

Figura 4.9 Ejemplo de cadena transmitida.

Command format:

byte	function	size
0	the number of individual requests in this command (max 20)	1
1	the number to pay	2
3	the denomination value	4
7	the denomination ASCII country code	3
10	repeat block for each required denomination	
	The option byte (TEST_FLOAT_AMOUNT 0x19 or FLOAT_AMOUNT 0x58).	1

Figura 4.10 Formato de los parámetros del comando.

Donde la estructura de la cadena incluye entre otros datos: un indicador de inicio de la cadena, el comando de la instrucción a ejecutar, los parámetros para dicho comando, un *checksum*, el código del aparato que recibirá el mensaje y un indicador de fin de la cadena, entre otros elementos según sea el caso. Cabe mencionar que algunos comandos deben ir encriptados al transmitirse para poder ser aceptados por el aparato.

4.2.4.2 Diseño del almacén y de la base de datos

Ya elegidos los componentes, mi jefe buscó proveedores para la fabricación del almacén de la Robotina, además de supervisar en todo momento la construcción de la misma, por ello y dado que no requería atención especial de mi parte, tras haber obtenido las medidas de todo el *hardware* y diseñada la estructura de la caja, nos dispusimos a diseñar y crear la base de datos para el sistema.

La base de datos está hecha en SQL Server 2014 y se encuentra disponible únicamente de forma local dentro de la computadora. Nos basamos en gran medida en la que utiliza el actual punto de venta, sobre todo por el hecho de que se tenía contemplado que en algunas sucursales trabajaran en conjunto la Robotina y el POS sobre la misma base de datos. Únicamente agregué procedimientos almacenados, campos y registros de configuración para que la central de cobro pudiera trabajar sobre la misma base de datos, ya sea de forma independiente o en conjunto con el punto de venta.

4.2.4.3 Diseño y desarrollo del software

Han existido dos versiones del sistema, aunque oficialmente sólo salió una versión a la venta, la primera versión fue realizada con Delphi debido a que como ya se mencionó era el lenguaje preferido de la consultoría en ese tiempo, se elaboró el diseño de la interfaz gráfica en conjunto con el cliente y se le dio

prioridad a la velocidad de la transacción debido al alto volumen de personas que cada semana manejaba la lavandería. El diseño final fue similar al de las figuras 4.11 a 4.15, debajo de cada una se incluye la descripción de sus funciones.

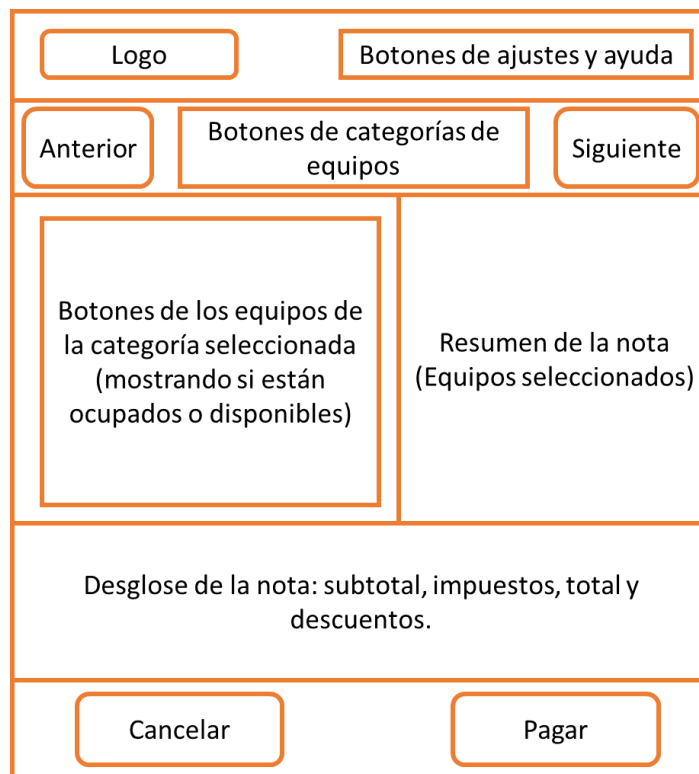


Figura 4.11 Diseño de la versión 1.

El sistema constaba de dos pantallas:

- Selección: donde estaba la mayor parte de la interacción con el usuario, en ella se elegían los equipos a activar (lavadoras y secadoras de los tamaños disponibles), se mostraba el resumen de los equipos seleccionados, así como el desglose de precios, al cancelar se borraba toda la selección.
- Pagar: al presionar pagar desplegaba una pantalla básica donde se mostraba el dinero recibido por los aparatos de *vending*, también mostraba un panel para canjear los puntos de las tarjetas de lealtad tras escanearlas.

El desarrollo de esta primera versión tardó alrededor de 5 meses, mostré avances de forma semanal a mi jefe y a su vez ambos presentábamos los avances cada que se consideraba pertinente al cliente, para hacer los cambios o agregar las funcionalidades que solicitaba casi en cada reunión.

La primera versión contemplaba funciones básicas para el funcionamiento de la Robotina: selección y activación de equipos, pago con efectivo y cambio en monedas, abono y cobro de puntos en tarjeta de lealtad e impresión de *tickets*. Para la recta final de desarrollo se estaba contemplando la posibilidad de aceptar pagos con tarjeta, sin embargo, no sería una funcionalidad que se desarrollaría inmediatamente al lanzar la versión a producción, sino más adelante según la aceptación de la Robotina por el público.

Cuando se terminó el desarrollo, el cliente quedó satisfecho con el resultado, por lo que decidió lanzarlo a producción sin sustituir el punto de venta aún hasta que las empleadas y sus clientes dominaran el uso del sistema. Por desgracia, pese a que la primera versión fue bien recibida por el público joven, las empleadas y el público mayor de 40 años lo consideró complicado de usar al tener todo en una sola pantalla, aun cuando el personal les explicaba en todo momento el uso del *software*. Estuvo en producción alrededor de 2 meses antes de que el cliente solicitara una reforma a la interfaz gráfica para abandonar la rapidez de transacción y hacerlo más amigable y entendible por todo el público.

Por parte de la oficina, aunque ya habíamos entregado el *release*, nunca dejamos de trabajar en el proyecto, ya que nos enfocamos en el módulo de pago con tarjeta, el cual fue complicado de arrancar debido a los trámites con el banco para que nos proporcionaran una terminar con ambiente de pruebas y toda la documentación necesaria para iniciar. Cuando el banco finalmente nos proporcionó lo necesario, nos indicó que la librería para la interacción con sus dispositivos no era compatible con Delphi.

Al recibir la solicitud del cliente para reformar por completo la interfaz gráfica, la consultoría volvió a ponerme a cargo del proyecto ya que mi trabajo anterior les había parecido excelente, además del hecho de que al haberlo desarrollado por completo, yo era quien más sabía del mismo, aproveché esta revolución en el sistema para sugerir una migración total al lenguaje de programación C#, ya que era uno de los lenguajes compatibles con la librería del banco y era muy similar a Delphi, por lo cual, a pesar de no estar familiarizado con dicho lenguaje, no presentaría una curva de aprendizaje tan compleja.

C# es un lenguaje de programación moderno y orientado a objetos desarrollado por Microsoft, permite a los desarrolladores crear muchos tipos de aplicaciones seguras y sólidas que se ejecutan en .NET. El lenguaje tiene sus raíces en la familia de lenguajes C y es inmediatamente familiar para los programadores de C, C++, Java y JavaScript. Varias características de C# ayudan a crear aplicaciones sólidas y duraderas. La recolección de basura recupera automáticamente la memoria ocupada por objetos inaccesibles no utilizados. El manejo de excepciones proporciona un enfoque estructurado y extensible para la detección y recuperación de errores. La sintaxis de *Language Integrated Query* (LINQ) crea un patrón común para trabajar con datos de cualquier fuente [23]. Además, al ser un lenguaje ampliamente aceptado a nivel mundial y al ser uno de los lenguajes más utilizados en la actualidad, cuenta con una comunidad activa y un sinnúmero de librerías que facilitan casi cualquier tarea.

Por su parte, .NET es un *framework* o una plataforma para el desarrollo de *software* que fue lanzada por Microsoft con la finalidad de fusionar su amplio catálogo de productos, que va desde sus múltiples sistemas operativos hasta herramientas de desarrollo. Permite realizar tanto desarrollos web, como programas de escritorio o aplicaciones para dispositivos móviles. Además, favorece el desarrollo en

multiplataforma [24]. Sus principales beneficios son la disminución de tiempo de desarrollo al poder utilizar funcionalidades prediseñadas en lugar de hacer todo desde cero, simplificación en el mantenimiento de los proyectos realizados en esta plataforma y por consiguiente una reducción en los costos.

La idea de migrar el sistema de Delphi a C# fue aceptada por la consultoría y por el cliente, entendiéndose que, aunque el desarrollo de esta segunda versión tardaría más que si conserváramos el mismo lenguaje, el resultado final sería notablemente mejor y con mayor compatibilidad para cambios futuros.

A diferencia de la versión anterior que se resumía a 2 ó 3 pantallas, el nuevo sistema para la Robotina estaría dividido en más de 6 pantallas para hacerlo más sencillo para el usuario, a continuación, muestro el diseño y una corta descripción de las principales pantallas del sistema.

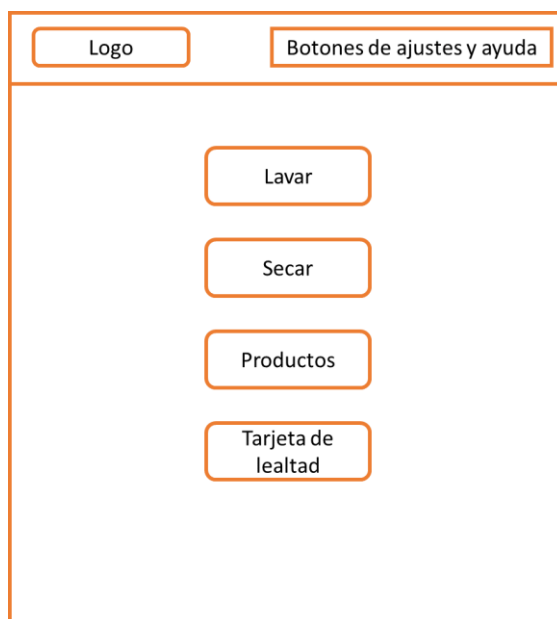


Figura 4.12 Pantalla principal v2.

La pantalla principal (véase la figura 4.12) contiene 4 botones, los 2 primeros llevan a la selección de los tamaños disponibles de la categoría elegida, el botón de “Productos” lleva directamente a la selección de los productos disponibles y el último botón abre una ventana para dar de alta o usar una tarjeta de lealtad para hacer uso de los puntos o aplicar en automático los descuentos disponibles. Hay varias opciones configurables, el botón de productos y el módulo completo de tarjetas de lealtad son unos de ellos, para el caso de “Productos” se puede deshabilitar el botón y la pantalla de selección de los mismos (si la lavandería no vende productos), mientras que para el módulo de tarjetas de lealtad las opciones son un tanto más complejas:

- Activar o desactivar generación de descuentos: en caso de activarlos se puede elegir si el descuento es sobre el total de la nota o si es por producto (se puede asignar un precio diferente por día).
- Activar o desactivar puntos: si se activa esta opción se puede elegir si sólo los equipos generan puntos o si es sobre el total de la nota, en esta segunda opción se puede asignar un porcentaje que se recuperará como puntos o qué monto genera un punto (Por ejemplo, cada 3 pesos = 1 punto).
- Activar o desactivar el uso de las tarjetas de lealtad por completo.



Figura 4.13 Elección de tamaño.

Siguiendo el flujo del programa, la siguiente pantalla (véase la figura 4.13) es la selección del tamaño de los equipos, esta pantalla es la misma para las lavadoras y las secadoras, su información se llena al cargar la pantalla para mostrar sólo los tamaños disponibles en la sucursal y el precio de cada tamaño, en caso de estar activa la función de descuento por equipo también se mostrará al usuario cuál es el precio de los equipos en caso que cuente con una tarjeta de lealtad, para fomentar la compra y uso de las mismas.

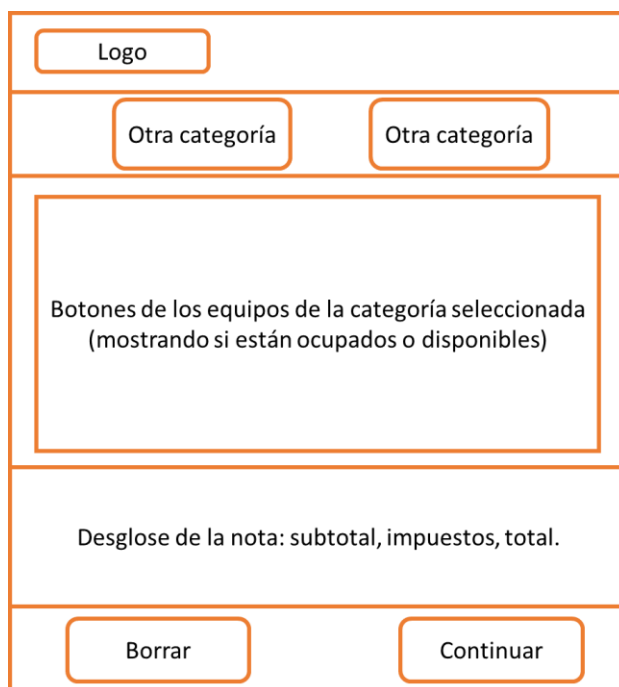


Figura 4.14 Pantalla de selección.

Una de las pantallas más importantes de todo el sistema es la selección de equipos (véase la figura 4.14), la sección central de dicha pantalla es donde se muestran todos los equipos de la categoría y tamaño seleccionados con anterioridad, los equipos que se encuentran ocupados tendrán su nombre en color rojo y los disponibles tienen el nombre verde, cada que presionamos un botón “reservar” el equipo para que nadie más (POS) lo pueda seleccionar en dos minutos mientras se termina la transacción, si no se termina en dicho tiempo, todos los equipos seleccionados serán liberados para ser usados por otro cliente. Para el caso de las secadoras, cada toque representa un periodo de tiempo previamente establecido por el dueño de la sucursal, de esta forma puede ofrecer los servicios de secado en múltiplos de 5 minutos, por ejemplo, y el tiempo acumulado se muestra junto al nombre del equipo para que el cliente sepa en todo momento cuánto tiempo va a pagar.

En la parte superior encontramos dos botones que nos permiten mover a otra categoría, por ejemplo, si nos encontramos en la selección de lavadoras, nos permitirá ir a la selección de productos o a la pantalla de tamaños de secadoras, de esta forma nosotros podemos interactuar entre los distintos servicios de la sucursal en una sola transacción.

En la parte inferior tenemos un pequeño panel para ver el desglose del saldo acumulado hasta el momento basado en nuestra selección, también hay dos botones que nos permitirán borrar el contenido de nuestra nota o continuar a la siguiente pantalla.

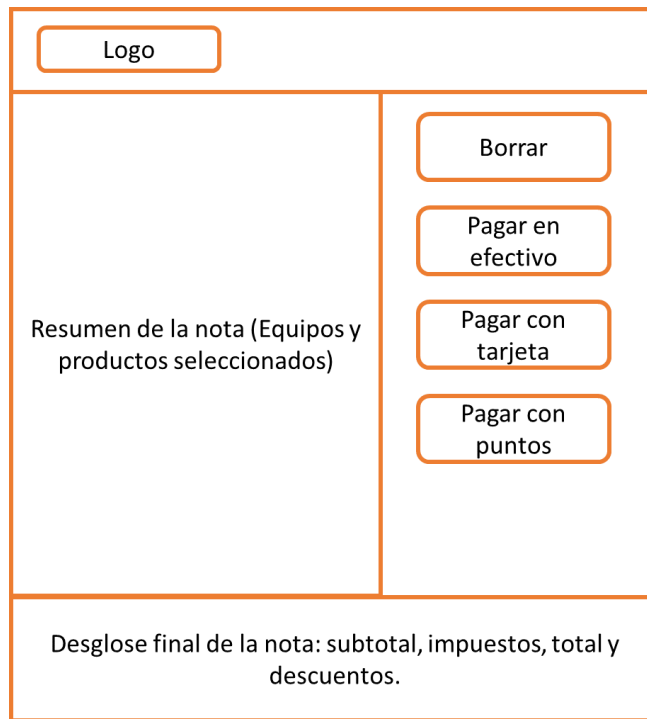


Figura 4.15 Pantalla de resumen.

Si continuamos con el proceso llegaríamos a la pantalla de resumen (véase la figura 4.15), en ella se muestra una lista de todos los equipos y productos seleccionados, en conjunto con el desglose de la nota, en caso de haber escaneado la tarjeta de lealtad en la pantalla principal, y haber un descuento disponible para la nota (precio especial o descuento sobre el total), este se aplicará de forma automática y se verá reflejado en el campo de descuento.

Si presionamos borrar nos regresará a la pantalla anterior y borrará el contenido de la nota (en caso de haber seleccionado equipos los liberará). Cabe mencionar que, si cancelamos la operación, la nota se vacía por completo en la base de datos, pero se reutiliza en la siguiente operación para no dejar notas en blanco o realizar más cancelaciones de las necesarias en la base de datos.

Finalmente nos presentará de una a tres opciones de pago:

- Efectivo: actualmente esta opción siempre está disponible y nos abrirá una pantalla con el saldo restante de la nota. La pantalla se cargará en un hilo independiente de ejecución para evitar que se congele, al mismo tiempo el hilo principal del programa se comunica con los aparatos de *vending machine*, mandando por medio del bus o de los buses existentes los paquetes necesarios para inicializar la recepción de dinero y mantener escaneando los buses para reflejar en la interfaz gráfica cada que se inserta dinero en la máquina, en cuanto el sistema detecta que se ha introducido el importe exacto o mayor al saldo restante, manda un paquete a los aparatos para

que detengan sus operaciones y ya no permitan introducir dinero. En ese momento se calcula el cambio y analiza la mejor forma de darlo, comenzando con las mayores denominaciones de monedas o billetes (para la versión *full*) y de esta forma asegurar una mayor autonomía, una vez que se calculó el cambio en relación al monto a entregar y la disponibilidad de efectivo, arma el paquete y lo envía al aparato correspondiente para dispensar el cambio.

- Puntos: este botón es configurable, basado en si el cliente solicita que la Robotina incluya el módulo de tarjetas y si es así, también dependerá de si él decide incorporar la funcionalidad de puntos en sus tarjetas de lealtad, de ser así, y en caso de que se haya escaneado la tarjeta desde el inicio, aparecerá el botón y al presionarlo nos abrirá una pantalla donde se mostrará el nombre del dueño de la tarjeta, sus puntos disponibles y un campo donde se deberá introducir la cantidad de puntos que queremos redimir.
- Tarjeta de débito o crédito: esta es una de las nuevas funcionalidades que se incorporaron en la versión 2 del sistema. El botón es configurable según se haya elegido incorporar el módulo de pago con tarjeta o no, si decidimos aceptar, al presionarlo el sistema mandará un paquete a la terminal por medio de la dll (*Dynamic Link Library*) proporcionada por el banco, dicho paquete será armado en base al saldo restante de la nota y hará que la terminal bancaria empotrada a la Robotina, solicite introducir la tarjeta, mostrando el monto que se cobrará al cliente y solicitando la firma electrónica en caso de ser necesaria, toda la información es confidencial y manejada directamente por el banco, la dll únicamente le manda a nuestro sistema la confirmación del pago, el monto cobrado y el tipo de tarjeta.

Por último, sin importar la forma de pago elegida por el cliente, al cubrir todo el saldo de la nota, el sistema preguntará si se desea imprimir el *ticket*, mostrará una pantalla de carga mientras se termina la transacción y mandará la señal a los equipos seleccionados, por medio de puerto serial a los circuitos de automatización de la lavandería. Una vez hecho eso, abrirá una nueva nota en blanco y volverá a la pantalla de inicio en espera de un nuevo cliente.

Ya con el diseño aprobado por el cliente y tras haber realizado el maquetado de todo el sistema, procedí a realizar la lógica del mismo, tanto el *frontend* como el *backend* están integrados en el mismo proyecto, por lo cual, fui trabajando a la par en ellos.

Las funcionalidades nuevamente fueron divididas en módulos y cada uno de estos módulos en tareas diferentes que se iban colocando en el tablero para saber el estatus de cada una de ellas, teníamos una columna de pendientes, en progreso, terminada sin probar, en pruebas y finalizada. Esta separación por módulos es la que nos permite hacer el sistema configurable en base a las necesidades de cada uno de nuestros clientes. Las primeras tareas que se debían realizar eran la conexión con la base de datos y la seguridad del sistema contra copias no autorizadas.

Entity Framework

Para llevar a cabo las conexiones con la base de datos, utilicé *Entity Framework*, que es un conjunto de tecnologías en ADO.NET que respaldan el desarrollo de aplicaciones de *software* orientadas a datos [25]. Permite a los desarrolladores convertir sus estructuras de datos en clases para poder trabajar con esa información usando objetos de .NET, reduciendo considerablemente la cantidad de código que sería necesario para realizar las tareas de acceso y gestión de datos en nuestra aplicación [26].

En resumen, esta herramienta, tras haber colocado las credenciales para conectarnos con nuestra base de datos, mapea las tablas, funciones, *stored procedures*, y demás elementos que nosotros elegimos de forma manual, de esta forma, cuando nosotros necesitemos conectar con la base de datos únicamente debemos llamar a nuestra entidad y solicitarle la acción requerida, todas las clases necesarias para recibir o manejar la información ya se encuentran hechas, por lo cual los desarrolladores sólo debemos preocuparnos por el uso de los datos y no por cómo obtenerlos o manejarlos.

Una vez establecida la forma de conexión con la base de datos, la siguiente tarea en la lista fue utilizar el mismo método que cree en Delphi para mantener seguras las licencias y evitar el acceso no autorizado a la aplicación. Este método consiste en ligar la licencia al número de serie del disco duro de la computadora que la dio de alta, de modo que, si algún usuario intenta utilizar el sistema en otra computadora sin adquirir una nueva licencia con anterioridad, la aplicación no los dejará ingresar.

Las tareas siguientes consistieron en su mayoría de migrar el código hecho en Delphi a C#, modificando la lógica para funcionar a lo largo de todas las pantallas mencionadas anteriormente en lugar de una pantalla unificada.

Como ya se mencionó, a lo largo de la aplicación se hace uso de uno o varios hilos de ejecución para asegurar la estabilidad de la aplicación y reducir los tiempos de espera mediante el paralelismo. Podemos encontrar dicha programación sobre todo en los módulos que implican una comunicación con los aparatos de *vending*.

El desarrollo de la nueva versión de Robotina, tomó aproximadamente otros 5 meses, debido a la curva de aprendizaje del nuevo lenguaje de programación utilizado, los periodos de espera obligados por parte del banco o del proveedor del *SMART Payout* para entregar las librerías y documentación necesarias, y por la integración de las nuevas funcionalidades que fueron surgiendo por parte del cliente, como los reportes o el corte de caja automatizado.

4.2.4.3 Pruebas al sistema e implementación en producción

Respecto a las pruebas, las fui realizando durante todo el desarrollo y antes de presentarlo a mi jefe en la junta semanal, para mitigar la mayoría de bugs antes de este punto. De igual manera ambos realizábamos pruebas exhaustivas y en ocasiones implicando a un tercero antes de presentar el avance al cliente.

Las pruebas previas a la versión final del *software* fueron llevadas a cabo de la siguiente forma:

- Durante el desarrollo realizaba pruebas siguiendo el “camino correcto” únicamente para asegurarme de que la nueva funcionalidad en la que trabajaba cumpliera con las expectativas.
- Una vez terminada la funcionalidad en curso, realizaba pruebas depurando el valor de las variables y las instrucciones condicionales tanto siguiendo el “camino correcto” como simulando ser un usuario inexperto. Todo esto con el fin de eliminar los errores o problemas más comunes.
- Antes de presentar el resultado en la junta semanal, hacía pruebas a mayor velocidad, tratando de presionar botones incorrectos, sin esperar a que la interfaz termine de cargar, sometiénola a un estrés un poco mayor, ya que se esperaba que el sistema fuera a prueba de cualquier tipo de usuario (experimentado, inexperto, pacientes, apresurados, entre otras cualidades) y trato de los mismos (uso rudo).

El periodo de pruebas finales se dividió en tres partes, cuando ya tuve una versión estable y casi todas las funcionalidades, llevamos a cabo pruebas internas dentro de la oficina de la consultoría, primeramente mi jefe y yo realizando pruebas basadas en un archivo que cree, contenía una lista de a mayoría de combinaciones posibles para el uso del sistema, por ejemplo, nota con tarjeta club + uso de lavadoras + pago en efectivo, nota con tarjeta club + uso de lavadoras + pago en efectivo + pago en tarjeta, nota sin tarjeta club + uso de secadoras + pago en puntos + pago con tarjeta + descuento con tarjeta club, entre muchas otras combinaciones posibles. Realizando dichas tareas primero como un usuario experimentado y luego tratando de emular a un usuario totalmente inexperto en la tecnología.

En la segunda parte de las pruebas, ya con la versión final del *software* y aún dentro de la oficina, recurrimos a miembros que no estaban involucrados en el proyecto, pero que tenían noción de cómo funcionaba el punto de venta, para probar si el sistema era intuitivo, tales como compañeros del área de soporte técnico, becarios y empleadas de una de las lavanderías.

Posteriormente iniciamos un breve periodo de pruebas directamente en la sucursal por parte de las empleadas de la lavandería y en ocasiones con la ayuda de clientes de la sucursal que desconocían por completo el proyecto. Esta vez su nivel de aceptación y entendimiento fue abismalmente mayor, por lo cual, tras un par de semanas de capacitación y simulaciones de ventas, nuevamente se lanzó la Robotina a producción y el cliente solicitó nuevamente mi presencia durante un par de semanas en su establecimiento, para corroborar que todo trabajara en forma y para ayudar a las empleadas en caso de que no recordaran algún punto de su capacitación.

En esta ocasión, tanto el cliente como la gran mayoría de sus visitantes quedaron satisfechos con el sistema, por lo tanto, se dio por terminado el proyecto. El cliente siguió utilizando la central de cobro aún a mi salida de Inventronix en abril del 2021, además, para ese punto ya habíamos implementado Robotinas en 5 lavanderías más, 4 de ellas esparcidas en la Ciudad de México siendo todas de distintos propietarios y/o marcas y una en la ciudad de Mérida en el estado de Yucatán, todas implementadas desde la automatización hasta la capacitación por mí, personalmente.

Diagrama de Gantt de la versión 1

La primera versión de Robotina inició con una breve toma de requerimientos, como se muestra en la figura 4.15, puesto que fue un proyecto interno que ya se venía discutiendo tiempo atrás en la consultoría. El proyecto se consideró sin contratiempos, ya que a pesar de que se tuvo que cambiar de pantalla puesto que presentaba problemas, no representó un impacto real en la duración del proyecto. Además, todas las tareas fluyeron de forma continua y sin interrupciones.

El proyecto se implementó en producción aproximadamente en el mes de junio del 2019, desafortunadamente sólo estuvo alrededor de un mes en uso real antes de que se retirara y se solicitara una nueva versión del mismo.

Robotina v1

Año. 2019

Periodo de realizacion. 1 año

Proyecto sin contratiempos

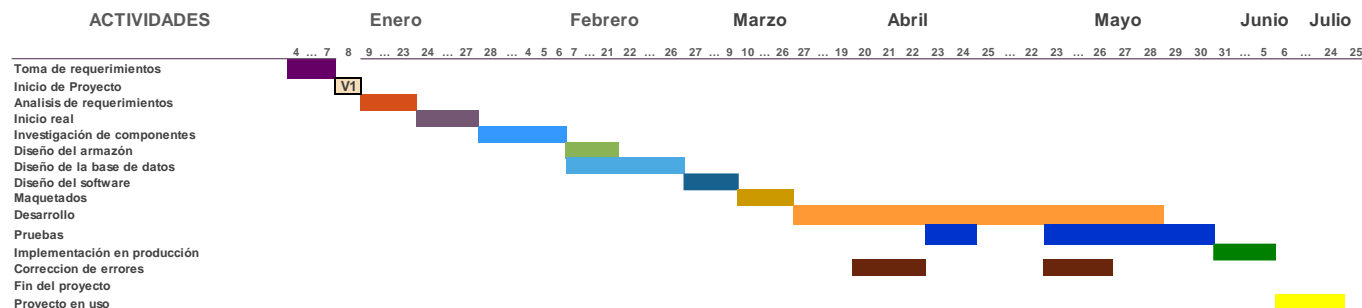


Figura 4.16 Diagrama con los tiempos aproximados de la primera versión del proyecto Robotina.

Diagrama de Gantt de la versión 2

La segunda versión de Robotina se basó en gran parte en el código de la primera versión, a pesar de ser en lenguajes de programación diferentes todo era muy similar, la mayor cantidad de tiempo invertido en el desarrollo del mismo se debió a las nuevas características de la central y al nuevo *hardware vending* que implementaba la versión *full*. Es por ello que los tiempos fueron considerablemente menores en la mayoría de las actividades, como se puede apreciar en la figura 4.17, aunque el tiempo total fue casi el mismo debido a trámites con proveedores o con el banco para el uso de sus tecnologías, y la curva de aprendizaje que su uso representó.

La segunda versión se implementó por primera vez en enero del 2019 y ese mismo mes se dio por concluido el proyecto, ya que la aceptación fue notablemente mayor y dicha versión siguió vigente al menos hasta mi salida de Inventronix en abril del 2021.

Robotina v2

Año. 2019

Periodo de realizacion. 1 año

Proyecto sin contratiempos

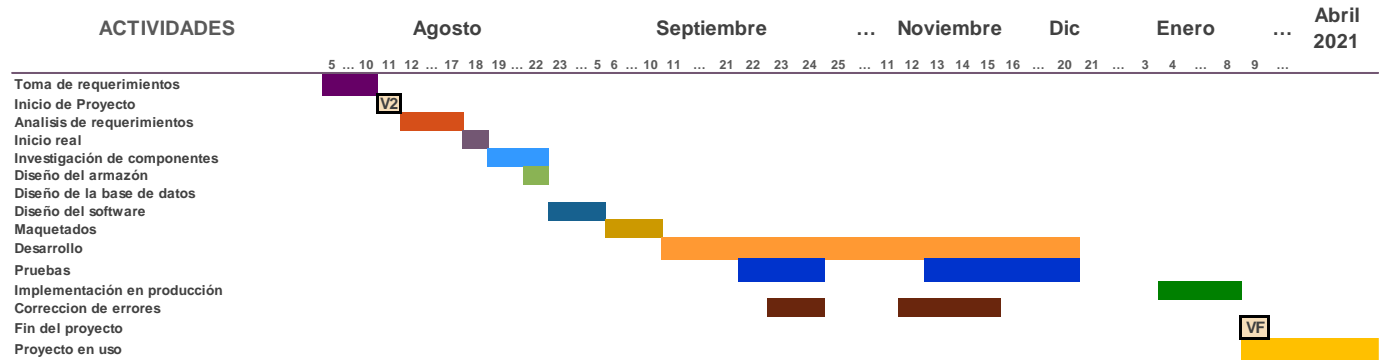


Figura 4.17 Diagrama con los tiempos aproximados de la segunda versión del proyecto Robotina.

Capítulo 5

RESULTADOS OBTENIDOS

La segunda versión de Robotina no sólo gustó a las empleadas de la lavandería, sino que tuvo un gran recibimiento por la mayoría de los clientes, las primeras semanas siempre solicitaban ayuda de las empleadas para activar sus equipos, sin embargo, después de este tiempo la mayoría de los usuarios llegaba directo a la central y sin solicitar ayuda se atendían por sí mismos, activaban sus equipos y ya no tenían necesidad de esperar a ser atendidos.

Aun así, hubo algunos clientes que se rehusaban a utilizar la máquina, por lo cual se realizaron algunas entrevistas entre los usuarios en general para ver el por qué, los resultados fueron que la población que no utilizaba la Robotina eran personas de edad avanzada y al preguntarles por qué no la usaban nos decían que preferían ser atendidos por una persona por que se sentían más cómodos. Mientras tanto, todos los demás usuarios nos decían que les gustaba la Robotina ya que muchas veces las empleadas estaban ocupadas con otras tareas y tenían que esperar no sólo a que se atendiera a los de adelante de la fila, sino a que las empleadas terminaran las demás tareas y ahora con la central ya no había ese problema, los usuarios llegaban directo a alquilar sus máquinas, las filas eran menores, había menor error humano y por consiguiente menor retraso y lo más importante, la máquina jamás los trataría mal, entonces a la mayoría les gustaba más esta forma de ser atendidos.

Por el lado del cliente, la puesta en operación de la Robotina le trajo grandes beneficios, ya que las empleadas ahora se podían centrar en la limpieza del local, los servicios por encargo y demás tareas por completo, mientras que los clientes se atendían solos, además los errores por equivocación se redujeron notablemente y las empleadas ya no necesitaban comunicarse todo el tiempo con soporte técnico para solucionar las equivocaciones que cometían. De aproximadamente dos llamadas por hora que hacía cada sucursal para solicitar soporte por fallas, en su mayoría de veces se redujo a una sola, y eso debido a que se continuaba usando al punto de venta a la par de la central. De igual forma, se redujo el tiempo de inactividad, ya que mientras un corte de caja normal tardaba aproximadamente 10 minutos, ahora se hacía en menos de 2 minutos, por lo cual no se acumulaban los clientes para ser atendidos y más allá de eso, no había molestia por parte de los mismos al tener que esperar para activar sus equipos.

Finalmente, fue tal el éxito de la “Central de Cobro” que la consultoría, al tener los derechos sobre el proyecto, decidió venderlo a otras cadenas de lavanderías, fue así como me dieron la tarea de implementar la central de cobro junto con la automatización de los equipos, capacitar y brindar soporte a 6 lavanderías diferentes en varias ubicaciones de la ciudad de México e incluso en Mérida, Yucatán.

La metodología usada para el proyecto, Scrumban, fue perfecta ya que nos ayudó a mantener el orden, al mismo tiempo que nos brindaba información del proyecto y a calcular tiempos aproximados de finalización tras cada junta semanal, contemplando los nuevos cambios o problemas que iban surgiendo.

En cuanto a los planes a futuro, se contemplaba cambiar la forma como la “Central de Cobro” enviaba la señal, ya que se quería implementar una automatización inalámbrica para los equipos, también se quería aumentar la cantidad de reportes y de ser posible bajar los costos, sobre todo se tenía pensado crear una Robotina que no ocupara una pantalla, sino que tuviera botones físicos, para funcionar en pequeñas lavanderías, como por ejemplo, las lavanderías dentro de algunos condominios que sólo tienen un par de equipos y donde no convenía tener personal ya que los ingresos no eran suficientes para cubrir un sueldo y tener una buena ganancia, además de usarla en otros entornos, como en moteles.

Por desgracia, para entonces la pandemia iba avanzando y se detuvieron por completo los planes ya que los clientes y la misma consultoría no tenían suficientes ingresos para continuar en el momento, incluso llegó un punto donde todas las áreas tuvieron que recortar personal y el área de escritorio desapareció por completo, ya que los proyectos venían principalmente de los clientes de las lavanderías y al detener sus operaciones también se detuvieron los ingresos, razón por la cual dejé de trabajar en Inventronix.

En resumen, se listan los pros y contras del proyecto “Central de cobro automatizada - Robotina”:

Pros

- Se cumplió cabalmente con los objetivos de crear una máquina capaz de activar equipos de lavandería de forma automática, recibiendo como pago efectivo, puntos de lealtad y tarjetas bancarias, además de poder dispensar cambio en monedas y en algunos casos hasta en billetes.
- Se mejoró el rendimiento de las empleadas al quitarles la tarea más demandante de la lavandería.
- Se redujeron los tiempos de espera de los clientes, así como la satisfacción en general.
- Se redujeron los errores humanos, tanto de operación como económicos, al menos en cuestiones de trato con el cliente y pagos.
- El sistema es fácilmente modificable para ser actualizado o usado incluso en otros entornos, debido a su programación modular.
- La central tiene buena seguridad en general, tanto físicamente por el armazón metálico, como por parte del *software* al mandar los paquetes cifrados a los aparatos y al no almacenar datos confidenciales de las transacciones con la terminal bancaria.
- El sistema permite monitorear las ganancias en tiempo real vía remota, así como verificar los niveles de dinero dentro de la máquina, avisando cuando tiene poco o mucho.

Contras

- Algunos de los componentes son sumamente caros, por lo cual los costos, aún de la versión *lite*, son altos, por lo que no es muy usual que se venda dicho sistema.
- Debido a malas prácticas de los clientes (pagar una nota de pequeño valor con billetes de alta denominación), el rendimiento del cambio fue menor de lo esperado, aun limitando la cantidad de cambio que puede dar la máquina (sólo permite a los clientes ingresar billetes cuyo cambio respecto a la nota, sea igual o menor al monto configurado por cada dueño). La versión *lite* requiere ser rellenada en promedio cada 3 días de cambio, dependiendo de la cantidad de transacciones al día que realice, mientras que la versión *full* tiene un promedio de una semana gracias al cambio mixto.
- La profundidad de la versión *full* es muy grande, por lo cual es compleja su instalación en locales pequeños y el desperdicio de espacio es notable.

CONCLUSIONES

Trabajé en Inventronix por más de tres años y medio, comencé siendo becario con tareas simples y bajo supervisión y terminé como ingeniero de *software*, en ocasiones incluso teniendo a mi cargo un pequeño grupo de personas, ya sea para tareas de soporte y mantenimiento o para desarrollo de *software*.

La ventaja de haber estado en casi todas las áreas de la consultoría es que obtuve conocimientos muy variados que aún a la fecha me siguen ayudando con proyectos profesionales. Los beneficios principales son los conocimientos adquiridos sobre programación y la facilidad de palabra a la hora de hacer alguna presentación o capacitar personal.

Además, trabajar me ayudó a implementar los conocimientos adquiridos en la Facultad de Ingeniería de una forma práctica directamente en el campo laboral, e incluso en algunos semestres los conocimientos que yo adquirí primero en el trabajo me ayudaron a tener un mejor entendimiento de los temas relacionados y por consiguiente a aprobar mis materias con más facilidad.

Definitivamente tanto este trabajo como el proyecto principal desarrollado en este informe me ayudaron a crecer mucho como profesional, ya que aprendí a trabajar y tomar decisiones bajo presión, a administrar tiempo y recursos y pensar en el mejor camino para llevar a cabo una tarea, siempre manteniendo la calidad y la seguridad para todas las partes.

Mi vida profesional apenas está iniciando, sin embargo, considero que ya he dado varios pasos en la dirección correcta para tener éxito en la vida, tanto personal como profesionalmente.

Glosario de términos

- Tecnologías de la Información y la Comunicación (TIC): son los recursos y herramientas que se utilizan para el proceso, administración y distribución de la información a través de elementos tecnológicos, como: ordenadores, teléfonos, televisores, etcétera [27].
- Puntos de Venta (*POS*): terminal de venta o sistema *POS* es una agrupación de diferentes software y hardware que, al combinarse, permiten a los clientes finalizar su compra. Un *POS* consta de un ordenador principal que está conectado a varios terminales de caja, móviles o fijos [28].
- Gestión de Relación con los Clientes (*CRM*): se refiere a todas las estrategias, técnicas, herramientas y tecnologías utilizadas por las empresas para desarrollar, retener y adquirir clientes [29].
- Landing Page: es una página dentro de un sitio web, desarrollada con el único objetivo de convertir los visitantes en Leads o prospectos de ventas por medio de una oferta determinada. Generalmente tiene un diseño más sencillo con pocos enlaces e informaciones básicas sobre la oferta, además de un formulario para realizar la conversión [30].
- General Public License (GPL): es una licencia de software libre *copyleft* publicada por la *Free Software Foundation*. Los usuarios de un programa con licencia GPL son libres para usarlo, acceder al código fuente, modificarlo y distribuir los cambios; siempre que redistribuyan el programa completo (modificado o no modificado) bajo la misma licencia [31].
- Framework: es un entorno de trabajo que tiene como objetivo facilitar la labor de programación ofreciendo una serie de características y funciones que aceleran el proceso, reducen los errores, favorecen el trabajo colaborativo y consiguen obtener un producto de mayor calidad [32].
- Integrated Development Environment (IDE): es un sistema de software para el diseño de aplicaciones que combina herramientas comunes para desarrolladores en una sola interfaz de usuario gráfica (*GUI*) [33].
- RFID: es un sistema de almacenamiento y recuperación de datos remoto que usa dispositivos denominados etiquetas, tarjetas, transpondedores o *tags* RFID. El propósito fundamental de la tecnología RFID es transmitir la identidad de un objeto (similar a un número de serie único) mediante ondas de radio [34].
- Vending Machine: es una máquina que proporciona aperitivos, bebidas, golosinas y otros productos a los consumidores. Se trata de vender sin la presencia de un dependiente para cobrar los artículos [35].
- Cashbox: salida del aparato tipo *vending machine*, que debe dirigirse hacia algún almacén externo de monedas o billetes para guardar los sobrantes sin posibilidad de reciclarlos.
- Baud: es el número de veces por segundo que una señal de comunicaciones serie cambia de estado; el estado puede ser un nivel de voltaje, una frecuencia o un ángulo de fase de frecuencia [36].

- Checksum: una suma de comprobación, es el resultado de la ejecución de un algoritmo dentro de un archivo único, función denominada *Cryptographic hash function*. Comparar el *checksum* que generas desde tu versión del archivo, junto al provisto por la fuente del mismo, representa una ayuda para asegurarte una copia genuina y libre de errores [37].
- Innovative technology (ITL): empresa con presencia internacional, proveedor líder de tecnología de manejo de efectivo [38].
- SMART Payout (SPO): reciclador de billetes fabricado por Innovative technology.
- Language Integrated Query (LINQ): conjunto de tecnologías basadas en la integración de capacidades de consulta directamente en el lenguaje C# [39].
- Dynamic Link Library (dll): es una biblioteca que contiene código y datos que pueden ser utilizados por más de un programa al mismo tiempo [40].

Referencias

- [1] A. Arellano y N. Cámara. (2017, ago.). La importancia de las TIC en las necesidades de la sociedad: una aproximación a través de la óptica de Maslow [Online]. Available: https://www.bbvaesearch.com/wp-content/uploads/2017/09/maslow_piramide.pdf
- [2] J. F. Grajales y Y. M. Osorno. (2019, Jul-Dic). La globalización y la importancia de las TIC en el desarrollo social [Online]. Available: <https://revistavirtual.ucn.edu.co/index.php/RevistaRyS/article/view/1133/1527>
- [3] Anónimo. (2021, Jun 1). ¿Qué es WordPress? Explicación para Principiantes [Online]. Available: <https://kinsta.com/es/base-de-conocimiento/que-es-wordpress/>
- [4] Anónimo. Usage statistics and market share of WordPress [Online]. Available: <https://w3techs.com/technologies/details/cm-wordpress>
- [5] Anónimo. LENGUAJE DE PROGRAMACIÓN DELPHI [Online]. Available: <http://www.larevistainformatica.com/Delphi.htm>
- [6] Anónimo. (2017, Mar 17). La historia de Delphi [Online]. Available: <https://www.greelane.com/es/ciencia-tecnolog%3%ada-matem%3%a1ticas/ciencias-de-la-computaci%3%b3n/history-of-delphi-1056847/>
- [7] Santander Universidades. (2020, Dic 21). Metodologías de desarrollo de software: ¿qué son? [Online]. Available: <https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html#:~:text=%C2%BFQu%C3%A9%20es%20una%20metodolog%C3%ADa%20de,dise%C3%B1ar%20soluciones%20de%20software%20inform%C3%A1tico>
- [8] Anónimo. (2019, Mar 21). El modelo en cascada: desarrollo secuencial de software [Online]. Available: <https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/el-modelo-en-cascada/>
- [9] A. Pérez. (2016, Ago 16). Características y fases del modelo incremental [Online]. Available: <https://www.obsbusiness.school/blog/caracteristicas-y-fases-del-modelo-incremental>
- [10] A. Pérez. (2014, Dic 23). La metodología Scrumban. Cuándo y por qué utilizarla [Online]. Available: <https://www.obsbusiness.school/blog/la-metodologia-scrumban-cuando-y-por-que-utilizarla>
- [11] S. Laoyan. (2022, Ene 28). Scrumban: lo mejor de dos metodologías ágiles [Online]. Available: <https://asana.com/es/resources/scrumban>
- [12] Anónimo. The FireMonkey Framework in RAD Studio [Online]. Available: <https://www.embarcadero.com/es/products/rad-studio/fm-application-platform>
- [13] Anónimo. Samsung Galaxy Tab A (2016) [Online]. Available: <https://computerhoy.com/fichas/samsung-galaxy-tab-2016>
- [14] Anónimo. SPP-R200III [Online]. Available: https://es.bixolon.com/product_view.php?id=213
- [15] Anónimo. SUNMI V2 [Online]. Available: <https://www.sunmi.com/es/V2/>
- [16] Anónimo. RFID – Tecnología de identificación por radiofrecuencia [Online]. Available: <https://www.kimaldi.com/rfid-tecnologia-de-identificacion-por-radiofrecuencia/>

- [17] Anónimo. Micro OptiPlex 3070 [Online]. Available: <https://www.dell.com/mx/empresas/p/optiplex-3070-micro/pd>
- [18] Anónimo. SMART Payout [Online]. Available: https://qualmex.com/265_recicladores+de+billetes_producto-detalle
- [19] P. Ballesteros. (2015, Sep 29). Introducción a la especificación MDB/ICP [Online]. Available: <https://patxiballesteros.wordpress.com/2015/09/29/introduccion-a-la-especificacion-mdbicp/>
- [20] Qibixx. ¿QUÉ ES UN MDB Y CÓMO SE INTEGRA EN LAS MÁQUINAS? [Online]. Available: <https://qibixx.com/es/que-es-un-mdb-y-como-se-integra-en-las-maquinas/>
- [21] Anónimo. SSP Communications Protocol Manual GA138 [Online]. Available: <https://www.innovative-technology.com/product-files/ssp-manuals/bv20-ssp-manual.pdf>
- [22] Anónimo. Protocol Manual SSP [Online]. Available: https://www.innovative-technology.com/images/pdocuments/manuals/SSP_Manual.pdf
- [23] B. Wagner. A tour of the C# language [Online]. Available: <https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/>
- [24] P. Pellicer. (2021, Oct 21). ¿Qué es el .NET? ¿Para qué sirve? [Online]. Available: <https://www.emagister.com/blog/que-es-el-net-para-que-sirve/>
- [25] G. Warren. (2021, Sep 15). Entity Framework overview [Online]. Available: <https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/ef/overview>
- [26] M. Romanos. (2019, Jul 9). Entity Framework y Linq [Online]. Available: <https://www.dreams.es/transformacion-digital/desarrolladores-paginas-web/entity-framework-y-linq>
- [27] Anónimo. Qué son las TIC y para qué sirven [Online]. Available: <https://www.ulatina.ac.cr/blog/qu3-son-las-tic-y-para-que-sirven>
- [28] Anónimo. Sistema POS (Terminal de venta) [Online]. Available: <https://www.sana-commerce.com/es/conceptos-de-comercio-electronico/que-es-terminal-punto-de-venta/>
- [29] Anónimo. ¿QUÉ ES UN CRM? [Online]. Available: <https://www.creatio.com/page/es/que-es-un-crm>
- [30] M. F. (2021, Oct 17). Landing Page: ¿Qué son las páginas de aterrizaje y cómo hacer una? [Online]. Available: <https://www.rdstation.com/es/blog/landing-page/>
- [31] Anónimo. (2022, Ene 12). GPL [Online]. Available: <https://developer.mozilla.org/es/docs/Glossary/GPL>
- [32] E. Rodríguez. (2020, Oct 22). Qué es un framework y por qué es conveniente usarlo [Online]. Available: <https://www.seoestudios.es/blog/que-es-un-framework/>
- [33] Anónimo. (2019, Ene 8). El concepto de IDE [Online]. Available: <https://www.redhat.com/es/topics/middleware/what-is-ide>
- [34] Anónimo. ¿Qué es el RFID? [Online]. Available: <https://www.ceupe.com/blog/que-es-el-rfid.html>
- [35] G. Nova. (2017, Feb 24). ¿Qué es Vending Machine? [Online]. Available: <https://medium.com/@practicante.cyber/qu%C3%A9-es-vending-machine-ab241e132e04>

- [36] Anónimo. Velocidad en baudios [Online]. Available: <https://www.ibm.com/docs/es/aix/7.2?topic=parameters-baud-rate>
- [37] C. ¿Qué es un checksum? [Online]. Available: <https://www.neoguias.com/que-es-checksum/>
- [38] Anónimo. What we do [Online]. Available: <https://innovative-technology.com/about-us/what-we-do>
- [39] B. Wagner y G. Warren. (2022, Ene 1). Language Integrated Query (LINQ) [Online]. Available: <https://docs.microsoft.com/en-us/dotnet/csharp/linq/>
- [40] H. Deland y H. Clu. (2021, Sep 23). What is a DLL [Online]. Available: <https://docs.microsoft.com/en-us/troubleshoot/windows-client/deployment/dynamic-link-library>
- [41]

Recursos

- Diagrama de conexión del SPO. Available: <https://www.innovative-technology.com/images/pdocuments/manuals/User Manual NV200 Range GA02031 v1.1.pdf>
-