UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA

DEPARTAMENTO DE INFORMÁTICA

VIÑA DEL MAR – CHILE

"SISTEMA DE APOYO PARA EL CONTROL DE VOLUNTARIOS EN SONRISAS SIN FRONTERAS"

MIGUEL ÁNGEL MONZÓN MARTÍNEZ
CARLOS RODRIGO MELLADO CONCHA

TRABAJO DE TITULACIÓN PARA OPTAR AL TÍTULO DE TÉCNICO EN INFORMÁTICA

PROFESOR GUÍA: **RODRIGO HERNÁN PINOCHET PONCE**

AGOSTO – 2018

RESUMEN

En el presente informe se recopila y muestra la información imprescindible para generar un mantenedor de registro de datos cuya función será la de otorgar una solución definitiva a varios problemas detectados en la organización social Sonrisas Sin Fronteras, un voluntariado de jóvenes comprometidos con el trabajo humanitario aplicado en niños en situación vulnerable. Para lograr este cometido, se utilizan técnicas de procesamiento de información con el fin de registrar las entidades y extraer sus características, relacionándose constantemente con las tablas creadas en una base de datos. Esta base de datos es de vital importancia para el almacenamiento de datos y el procesamiento de ellos en el programa, comprendiéndose esto como una entidad que permite tareas bidireccionales, vale decir, otorgando funciones de consulta como así también de respuesta.

La información necesaria para el resultado íntegro de la aplicación de escritorio ha sido recopilada a través de páginas online, documentos y entrevistas facilitadas por la administración de Sonrisas Sin Fronteras, con la finalidad de mostrar el desarrollo de este sistema computacional. Más adelante se detallará el procesamiento de los datos facilitados por la organización entregando el posterior análisis del estudio preliminar de la situación en la cual se encuentra el voluntariado de Quillota.

La importancia de desarrollar este proyecto responde a la necesidad que tiene la organización de mejorar los tiempos de respuesta a las solicitudes de los usuarios al momento de solicitar alguna operación del programa informático, como por ejemplo, registrar entidades, consultar información y generar documentos imprimibles que almacenen los datos del tipo de solicitud de una forma compacta y precisa. Es por ello que se ha decidido realizar los pasos sustanciales para forjar una nueva herramienta para la organización e introducir un modelo que automatice todas las tareas en contraste de las bitácoras físicas que se trabajan actualmente.

Finalmente, en el proceso de puesta en marcha del software por parte de la organización, se les ha notado un gran interés y conformidad, dado que se provee de una herramienta de fácil utilización y amistosa, la que permitirá facilitar las tareas que se desarrollan en el área de voluntarios.

Palabras Clave: organización, voluntarios, procesamiento de información, base de datos, automatización.

TABLA DE CONTENIDO

CAPÍTULO I: ASPECTOS RELEVANTES DEL DISEÑO LÓGICO 3

1.1 DESCRIPCIÓN DE LA ORGANIZACIÓN 4
1.1.1 HISTORIA DE LA ORGANIZACIÓN *5*
1.1.2 VISIÓN *6*
1.1.3 MISIÓN *6*
1.1.4 ORGANIGRAMA INSTITUCIONAL *6*
1.2 DESCRIPCIÓN DE LA SITUACIÓN ACTUAL 7
1.3 PROBLEMAS DETECTADOS 7
1.4 DESCRIPCIÓN DEL SISTEMA PROPUESTO 9
1.4.1 Objetivo general *9*
1.4.2 Objetivos específicos *9*
1.4.3 Beneficios *10*
1.4.4 Descripción general de la solución *11*
1.4.5 Diagrama de flujo administrativo *11*
1.4.6 Estructura Funcional del Sistema *14*
1.4.7 Información que se manejará *15*
1.4.8 Entradas del Sistema *15*
1.4.9 Salidas del Sistema *16*
1.4.10 Entidades de información *16*
1.4.11 Modelo lógico de datos *18*
1.4.12 Estructura de códigos *19*

CAPÍTULO II: MEDIO AMBIENTE COMPUTACIONAL Y DESCRIPCIÓN DE ARCHIVOS 20

2.1 DESCRIPCIÓN DEL RECURSO COMPUTACIONAL 21
2.1.1 Configuración del sistema *21*
2.1.2 Software Utilizado *22*
2.2 DESCRIPCIÓN DE ARCHIVOS 24
2.2.1 Comunidad *24*
2.2.2 Voluntario *24*
2.2.3 Colonia *25*
2.2.4 Col_Vol *26*
2.2.5 Rol *27*
2.2.6 Producto *28*
2.2.7 Niño *28*
2.2.8 Apoderado *29*
2.2.9 Nin_Col *30*
2.2.10 Usuario *31*

CAPÍTULO III: DESCRIPCIÓN DE PROGRAMAS **32**

3.1 DIAGRAMA DE MENÚS 33
3.1.1 DIAGRAMA DE MENÚ – INICIO DE SESIÓN *34*
3.1.2 DIAGRAMA DE MENÚ – MENÚ *34*
3.1.3 DIAGRAMA DE MENÚ – ADMINISTRADOR *34*
3.1.4 DIAGRAMA DE MENÚ – MANTENEDORES *35*
3.1.5 DIAGRAMA DE MENÚ – LISTADOS *35*
3.2 DIAGRAMA MODULAR 36
3.3 LISTA DE PROGRAMAS DESARROLLADOS 37
3.4 ESPECIFICACIÓN DE PROGRAMAS 38
3.4.1 PROGRAMA: MENÚ PRINCIPAL *38*
3.4.2 PROGRAMA: LOGIN *39*
3.4.3 PROGRAMA: NUEVO USUARIO *41*
3.4.4 PROGRAMA: NUEVA COMUNIDAD *43*
3.4.5 PROGRAMA: CREAR NUEVA COLONIA *45*
3.4.6 PROGRAMA: AGREGAR UN VOLUNTARIO A UNA COLONIA *47*
3.4.7 PROGRAMA: NUEVO APODERADO *49*
3.4.8 PROGRAMA: NUEVO NIÑO *51*
3.4.9 PROGRAMA: NUEVO VOLUNTARIO *53*
3.4.10 PROGRAMA: NUEVO PRODUCTO *55*
3.4.11 PROGRAMA: LISTADO DE VOLUNTARIOS POR COLONIA *57*

CONCLUSIÓN **60**

BIBLIOGRAFÍA Y LINKOGRAFÍA **61**

ANEXO: DOCUMENTACIÓN DE CÓDIGO **62**

ANEXO 1: CÓDIGO FUENTE DEL PROGRAMA MENÚ PRINCIPAL *62*
ANEXO 2: CÓDIGO FUENTE DEL PROGRAMA LOGIN *65*
ANEXO 3: CÓDIGO FUENTE DEL PROGRAMA NUEVO USUARIO *68*
ANEXO 4: CÓDIGO FUENTE DEL PROGRAMA NUEVA COMUNIDAD *71*
ANEXO 5: CÓDIGO FUENTE DEL PROGRAMA NUEVA COLONIA *73*
ANEXO 6: PROGRAMA FUENTE DEL PROGRAMA AGREGAR VOLUNTARIO A UNA COLONIA *76*
ANEXO 7: CÓDIGO FUENTE DEL PROGRAMA NUEVO APODERADO *79*
ANEXO 8: CÓDIGO FUENTE DEL PROGRAMA NUEVO NIÑO *82*
ANEXO 9: CÓDIGO FUENTE DEL PROGRAMA NUEVO VOLUNTARIO *88*
ANEXO 10: CÓDIGO FUENTE DEL PROGRAMA NUEVO PRODUCTO *94*
ANEXO 11: CÓDIGO FUENTE DEL PROGRAMA LISTADO DE VOLUNTARIOS POR COLONIA *96*

INTRODUCCIÓN

En la actualidad, las organizaciones luchan por obtener un lugar en la competencia de mercado y para ello se complementan con diversas herramientas para mejorar y optimizar sus procesos internos. Sin embargo, existe un grupo de personas disperso por la Quinta Región, quienes trabajan para una organización totalmente sin fines de lucro, sin desear competencia alguna, y que, sin hacer uso de tecnologías, se disponen a servir con el único fin de ver, permanentemente, sonrisas en los niños. Tal organismo recibe el nombre de ***Sonrisas Sin Fronteras***.

Sonrisas Sin Fronteras es un Voluntariado Juvenil que trabaja hace más de nueve años en la región de Valparaíso, realizando diversas actividades, proyectos y programas sociales enfocados en el desarrollo integral de niñas, niños y adolescentes, principalmente aquellos más vulnerables y, al ser una organización con naturaleza de emprendimiento único, el éxito de la misma está íntimamente ligado a la calidad y disposición de los recursos humanos (o entidades asociadas) que intervienen internamente, como también externamente.

Como organización consideran imperante la necesidad de definir aquellos valores que los representan, tanto en su trabajo con los niños, como en la manera en que se enfrentan a la sociedad en la cual están inmersos. Es por esto que han definido una serie de valores en base a sus propias palabras y nociones, las cuales de aquí en adelante les servirán de piedra angular para generar un trabajo serio, responsable y comprometido con la comunidad.

Como bien se sabe, la avanzada (o progreso) tecnológica/o, consecuencia de los descubrimientos y trabajos a nivel informático, el diseño y la fabricación asistida por computador, supone una profunda transformación del sistema productivo de cualquier organización, lo que ayuda, exponencialmente, en el crecimiento y supervivencia de esta misma. Hoy en día es fundamental hacer uso de herramientas tecnológicas con el fin de automatizar y estandarizar los procesos administrativos, lo que significa una entrega rápida y oportuna de la información requerida: esta es un elemento fundamental en la toma de decisiones, puesto que ella ordena, orienta y ayuda a verificar las metas y objetivos planteados.

Es por ello que el organismo de ***Sonrisas Sin Fronteras*** ha decidido sumarse y converger en la idea de comenzar a utilizar las herramientas tecnológicas que hoy se encuentran disponibles.

En este trabajo de título se busca generar sustancialmente una herramienta computacional, vale decir, crear un sistema de escritorio que cumpla con las características y funciones de un mantenedor: que entregue apoyo a las labores de recursos humanos para así poder mejorar y agilizar los procesos relacionados con la información de los voluntarios y sus proyectos actuales o venideros. Tomando en cuenta que en cada verano (temporada de trabajo más fuerte del voluntariado) participan alrededor de 250 voluntarios por cada colonia activa, y que, además, la cantidad de tareas aumenta proporcionalmente con la cantidad de participantes, se ha vuelto urgente apoyar, tanto en eficacia como en eficiencia, todo proceso comunicativo, entre los cuales se consideran: redes sociales, formato de reuniones, creación de comisiones, generación de documentos, etc. En este sentido, en respuesta a la incógnita de cómo hacer un uso ergonómico de los datos a futuro, es que el presente informe tiene por objetivo revelar el funcionamiento de un nuevo proyecto, en donde se dejará especificado el diseño e implementación de una solución computacional de escritorio útil, permitiendo al jefe administrativo a cargo, apoyar y agilizar los procesos relacionados con el área del voluntariado.

CAPÍTULO I: ASPECTOS RELEVANTES DEL DISEÑO LÓGICO

ASPECTOS RELEVANTES DEL DISEÑO LÓGICO

1.1 DESCRIPCIÓN DE LA ORGANIZACIÓN

Sonrisas Sin Fronteras es un Voluntariado Juvenil que trabaja hace más de nueve años en la región de Valparaíso, realizando diversas actividades, proyectos y programas sociales enfocados en el desarrollo integral de niñas, niños y adolescentes, principalmente aquellos más vulnerables; en un contexto de educación en valores, principalmente a través del juego y la experiencia.

Es una agrupación de voluntarios que, con gran dedicación y energía, trabajan para sacar adelante sus proyectos. En su mayoría estudiantes y con edades entre los 15 y 27 años, destacando también el apoyo incondicional que siempre se recibe por parte de sus ex voluntarios.

Sonrisas Sin Fronteras, al ser un voluntariado compuesto principalmente por la participación de jóvenes estudiantes secundarios y universitarios, durante sus diez años de existencia, ha tenido una maduración constante en miras a su misión y visión que hace referencia a su lucha por garantizar un desarrollo íntegro basado en los derechos y deberes de los beneficiarios del voluntariado que son los niños, niñas y adolescentes de diferentes situaciones sociales. Este proceso de crecimiento, lógicamente, ha contado con innumerables aciertos y errores que han llevado al voluntariado a un punto clave que manifiestan sus directores: transformarse en fundación asociada a un proceso importante de reorganización de la estructura en que se ha trabajado los últimos años.

Este voluntariado está implementando una sectorización de los territorios en los cuales se ha estado trabajando a través de la creación de “Comunidades”. La idea de este proceso es generar la identificación de cada voluntario con un territorio específico, con el fin de trabajar en base a un enfoque comunitario en el cual se buscará armar redes con juntas de vecinos, apoderados respectivos de cada niño, niña o adolescente y toda otra organización local que persiga motivos complementarios.

Toda acción de Sonrisas Sin Fronteras busca la promoción y defensa de los derechos del niño, niña y adolescente; desarrollando en los beneficiarios su carácter, su creatividad, su sociabilidad, su corporalidad, su espiritualidad y su afectividad. El éxito del método "Sonrisas" se puede entender como el logro de virtudes mediante la aplicación articulada y equilibrada de ciertos elementos: preocupación especial por los niños, niñas y adolescentes; comunidades de trabajo, esfuerzo de corazón y diversos componentes educativos y recreativos, de distinto origen o naturaleza. Si se logra un equilibrio en la aplicación de los elementos del método, se producirá el clima educativo que la agrupación busca en sus diferentes actividades y proyectos.

1.1.1 HISTORIA DE LA ORGANIZACIÓN

Sonrisas Sin Fronteras inicia sus actividades el verano del año 2007. Es en este que un grupo de ex alumnos del Instituto Rafael Ariztía, con la motivación y apoyo del hermano Marista Luis Cornejo, deciden servir a la comunidad, principalmente a los niños, niñas y adolescentes más desatendidos de la zona.

Es así que se genera la primera iniciativa del voluntariado y se realiza la Colonia de Verano Colmo 2007, que consistió en entregar una instancia de vacaciones entretenidas a los niños y niñas del sector, a través de una propuesta programática didáctica y juvenil. De esta forma se da inicio al voluntariado *Sonrisas Sin Fronteras*.

En estos cerca de 10 años de trabajo, se han realizado más de 50 proyectos en diversos territorios de la Quinta Región (Quillota, La Palma, Limache, Olmué, Colmo y La Calera), permitiendo acompañar a miles de niños, niñas y adolescentes en su crecimiento. Asimismo, ha crecido el número de jóvenes voluntarios que se adhieren al trabajo de *Sonrisas Sin Fronteras*.

Hoy, a casi 10 años de aquel verano, el sueño está más vivo que nunca y cada vez son más los jóvenes que creen en un mundo mejor y se suman al trabajo de tal organismo.

1.1.2 VISIÓN

Potenciar el desarrollo de niños, niñas y jóvenes, especialmente a aquellos más vulnerables, para que logren un desarrollo íntegro, de manera que sean forjadores de una nueva sociedad basada en valores, la alegría y el amor.

1.1.3 MISIÓN

Crear, gestionar y apoyar la ejecución de proyectos sociales, para niños, niñas y adolescentes enmarcados en la afectividad, desarrollo de habilidades sociales, entrega de valores, y promoción de los derechos del niño, niña y adolescente; principalmente a través del juego, incorporando a las familias y sus comunidades. En *Sonrisas Sin Fronteras*, incluso, los voluntarios son beneficiarios; su forma de trabajo es alegre y lo promueven en las comunidades como una forma de vida.

1.1.4 ORGANIGRAMA INSTITUCIONAL

Figura 1.1: Directorio institucional de Sonrisas Sin Fronteras

1.2 DESCRIPCIÓN DE LA SITUACIÓN ACTUAL

Hoy en día, el trabajo comunitario va creciendo exponencialmente. Cada vez son más los jóvenes que se interesan en participar en variadas actividades en su tiempo libre, y una de las áreas más demandadas es el trabajo constante con niños, niñas y adolescentes. En este sentido, *Sonrisas Sin Fronteras* ofrece participar como voluntario durante los períodos de vacaciones (temporada de invierno y verano) de cada año.

El método educativo de Sonrisas, se define como un sistema de educación no formal y participativo, complementario a otros agentes educativos (familia, amigos, escuela, etc.), que se fundamenta en la interacción de diversos componentes interrelacionados entre sí, y que pretende que niñas, niños y jóvenes se conviertan en el principal agente de su desarrollo, llegando a ser una persona comprometida con su entorno social, responsable de sí misma y el entorno, y principalmente feliz.

Además, *Sonrisas Sin Fronteras* funciona en base a la rotación constante de voluntarios en más de 10 cargos. Para alcanzar un cargo, la idea es que cada voluntario esté preparado o tenga las herramientas que lo hagan compatible a la función.

Finalmente, al ser un organismo sin fines de lucro y sustentado solo con aportes o donaciones de los mismos voluntarios, no ha podido contar con un sistema profesional para la ayuda y organización del registro del voluntariado y participantes de esta fundación.

1.3 PROBLEMAS DETECTADOS

a) Los recursos humanos con los que trabaja *Sonrisas Sin Fronteras* se encuentran escritos y adjuntos en diferentes bitácoras, lo cual hace demorar la búsqueda de información importante. Además, los datos anotados aquí no se presentan en una forma ordenada ni detallada, produciendo poca exactitud al momento de decisiones sustanciales.

b) Como el trabajo y labor es constantemente con niños, el cuidado de ellos es sumamente importante, y en caso de alguna negligencia, se debe contar con un sistema que contenga datos relevantes a la hora de actuar frente a una situación de emergencia, como el nombre del voluntario a cargo del grupo y contacto del tutor.

c) Actualmente, la tarea de responder a cualquier urgencia sigue el patrón de buscar en una base de datos física (hojas, block de apuntes, etc.) para llegar recién al resultado del niño y familiar relacionado a él. Al no presentarse lo anteriormente mencionado de forma ordenada, el socorro se ve demorado.

d) En *Sonrisas Sin Fronteras* el trabajo es jerárquico. Es así como se designan padrinos o madrinas en base a la experiencia del voluntariado. Para llegar a una decisión, el coordinador general deba revisar en su bitácora en cuantos proyectos ha participado cada persona y si además posee alguna sanción y cuan grave es esta. La falta de precisión en la información ha provocado que cada vez resulte más difícil la elección de cargos. Además, se han encontrado muchas veces con errores basados en la inexperiencia y falta de capacitación. En síntesis, tal organismo no posee un registro de voluntarios que esté siempre a la mano desde una plataforma virtual, para así poder contar con evaluaciones previas de cada voluntario.

e) Falta de información objetiva y oficial a la hora de tener que emitir certificados a voluntarios. Además, no se puede realizar un cómputo oficial de cuantos voluntarios son ni con cuantos pueden contar realmente a la hora de embarcarse en algún proyecto.

f) Poca transparencia y carencia de bases de datos reales con respecto al dinero recaudado y sobrante que se destina para realizar compras de productos para el inventario, como, por ejemplo: cloro, jabón, cepillos dentales, etc.

1.4 DESCRIPCIÓN DEL SISTEMA PROPUESTO

1.4.1 Objetivo general

El fin es desarrollar un sistema informático que facilite la gestión y el control de los recursos *(*comunidades, colonias, personas y productos) en el voluntariado, además de generar una solución definitiva a los problemas detectados y funcionar como respuesta a los requerimientos de Sonrisas Sin Fronteras. Tal aplicación de escritorio colaborará al manejo y control de datos varios, a través de una interfaz de fácil manejo, así mismo, apoyar las diferentes labores que el administrador o encargado de voluntarios se disponga a realizar, como el registro y mantención de los datos de jóvenes inscritos, generación de documentos imprimibles o tomas de decisiones en función a la información que proporcione el sistema.

1.4.2 Objetivos específicos

a) Crear un mantenedor de formularios para recopilar la información escrita en las bitácoras físicas y registrar los nuevos datos de ingreso.
b) Generar menús y vistas de fácil manejo, acceso y con interfaces amistosas para los usuarios del programa (miembros autorizados de *Sonrisas Sin Fronteras.*).
c) Otorgar seguridad al programa informático, a través de un formulario Login (inicio de sesión en función a un usuario y contraseña.).
d) Generación de informes varios de manera automática, conteniendo los datos de las consultas solicitadas (información de voluntario especificado, colonia o comunidad requerida e información del niño solicitado.).
e) Crear una base de datos para evitar el uso de bitácoras por cada voluntario y ente asociado de las colonias, además de los datos de ellas mismas y las comunidades.
f) Registrar, dentro del sistema, miembros de la organización Sonrisas Sin Fronteras, voluntarios y niños que participarán en las diferentes actividades, las actividades

mismas, tutores de los niños, materiales con los que se cuenta para el trabajo constante y diario por colonia, y las colonias y comunidades activas.

1.4.3 Beneficios

Entre los beneficios otorgados por el programa vale destacar:

a) Simplificación del uso y manejo del sistema: El programa contará con múltiples vistas, las cuales contienen claros mensajes en sus botones de acciones o barra de menú, dispuestos ordenadamente.
b) Procesos automatizados para la comodidad del usuario: Los datos sujetos a cada entidad procesable (Comunidad, Colonia, Apoderados, Niños, Voluntarios y Productos) serán ordenados y desplegados automáticamente en su menú respectivo con la finalidad del manejo a gusto del usuario, como, por ejemplo, modificar, eliminar o añadir una nueva instancia de la entidad.
c) Consistencia y coherencia en los datos: Para un buen desempeño de la aplicación y dar fin a las bitácoras físicas, se decidió montar una base de datos para su almacenamiento y/o lectura de ellos.
d) Orden y rápida visualización de la información requerida: Ya que la información ahora se despliega en una bitácora (o plataforma) computacional, el período de tiempo entre una solicitud y respuesta se reduce considerablemente. Consecuentemente, se pueden realizar varias operaciones en plazos cortos.
e) Transparencia en los recursos disponibles: Como el programa será manejado por el administrador o gente autorizada por él, los datos contables, materiales o intangibles, serán manejados con total transparencia y solo siendo modificados a través del tiempo por usuarios permitidos.
f) Seguridad absoluta de los datos: El acceso al programa principal estará condicionado por una ventana Login, que capturará los datos de usuario y contraseña de la persona a cargo del programa, siendo estos comparados con los ya registrados en la base de datos

y retornando una coincidencia y post funcionamiento del programa, o dando como resultado una autenticación inválida.

g) Acceso a través de múltiples cuentas: El sistema computacional otorgará a la organización el acceso a él mediante una cuenta inicial, pero desde la cual se podrán generar otras cuentas para usuarios seleccionados por el administrador.

1.4.4 Descripción general de la solución

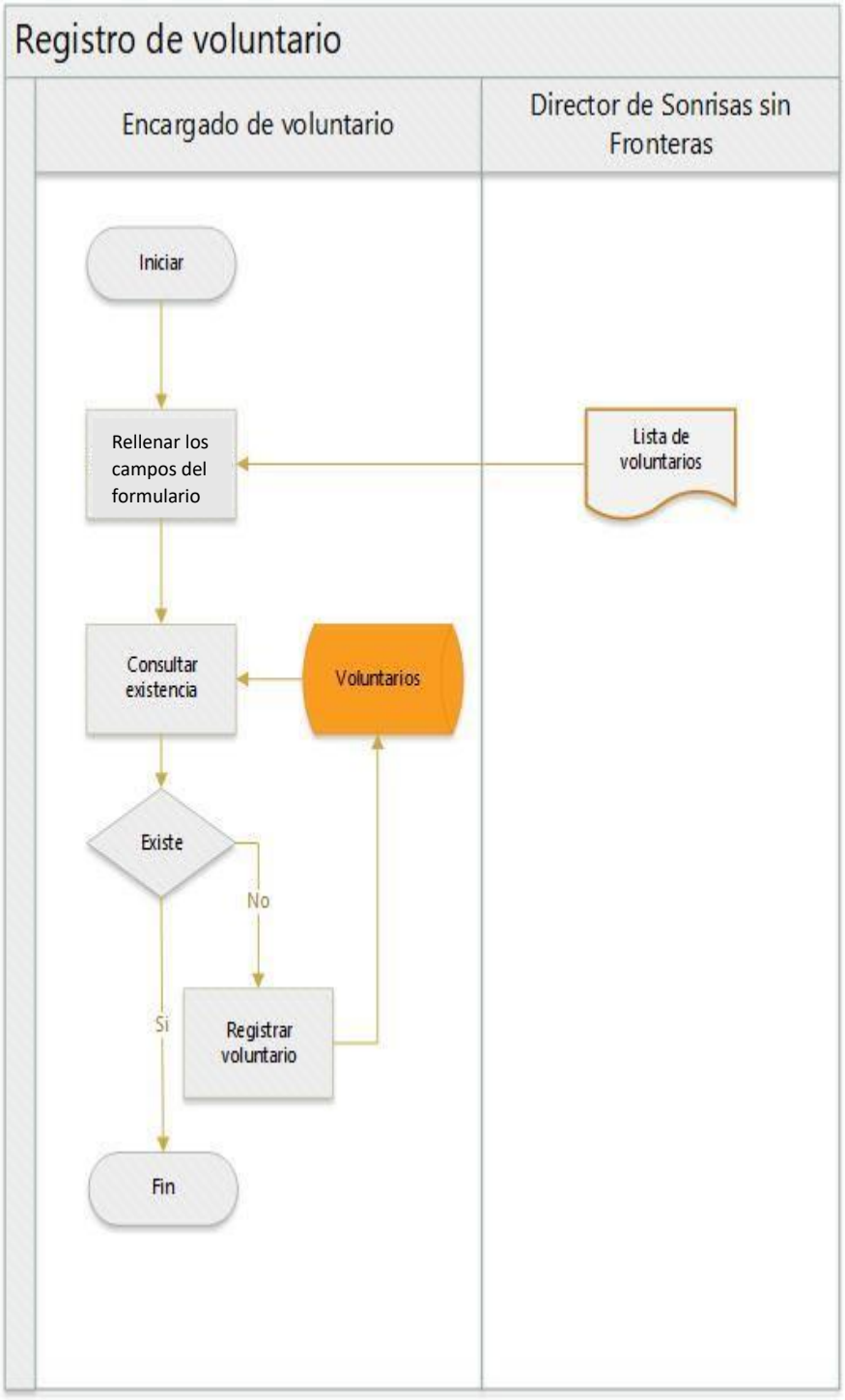
Sonrisas Sin Fronteras es una organización que cada año recluta cada vez a más niños-adolescentes, como también a voluntarios, lo cual demuestra el desempeño deslumbrante de sus actividades solidarias. Es por ello que se trabajará arduamente con el propósito de entregar una solución definitiva a sus problemáticas y que continúen con el curso de su trabajo de la manera más óptima. La solución propuesta para el voluntariado *Sonrisas Sin Fronteras* consiste en agilizar su gestión de captura de voluntarios a través de mantenedores integrados con formularios para el ingreso de datos sustanciales (nombres, apellidos, rut, fecha de nacimiento, etc.). Como también se tiene en cuenta la flexibilidad de los datos, así como el margen de error humano, se podrá navegar fácilmente en los mantenedores, permitiendo la modificación (o eliminación) de los datos.

La captura de datos de un voluntario solicitado es de vital importancia al momento de evaluar sus características o cualidades/aptitudes para la inscripción de este en una actividad determinada, por lo que se llevará cada respuesta ingresada en el formulario a una base de datos para futuras consultas. Lo anterior reducirá los tiempos de búsqueda y respuesta de la información requerida.

1.4.5 Diagrama de flujo administrativo

A continuación, se presentan los diagramas de flujo administrativo, con su respectiva explicación:

1.4.5.1 Procedimiento Administrativo: Solicitud de Registro



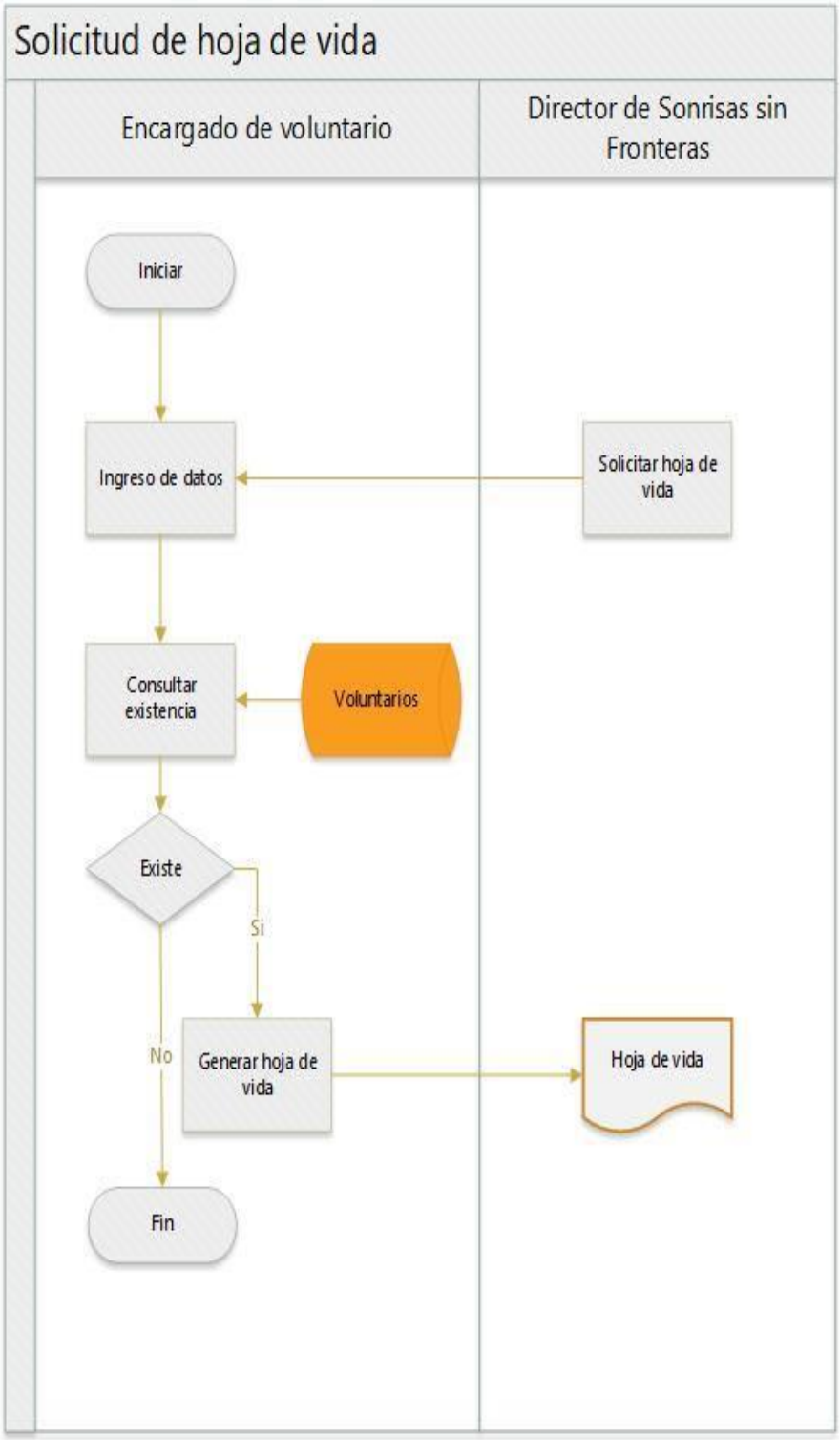
1. El Encargado de voluntarios recibe una solicitud de registro e inmediatamente ingresa los datos al formulario.

Posteriormente, chequea si éste se encuentra registrado en el sistema.

2. En caso de no existir el voluntario, el encargado lo registra en el formulario del mantenedor correspondiente.

3. Al momento de registrar sus datos satisfactoriamente en el sistema informático, estos son grabados en la base de datos correspondiente.

1.4.5.2 Procedimiento Administrativo: Solicitud de Hoja de Vida



1. El encargado del programa recibe la petición de generar la hoja de vida de un voluntario.

2. El encargado del programa ingresa el nombre del voluntario para verificar su existencia.

3. En caso que no exista una instancia del voluntario en la base de datos, se da por finalizado el proceso. En caso contrario se exporta automáticamente los datos de él hacia un archivo pdf imprimible.

1.4.6 Estructura Funcional del Sistema

El sistema a desarrollar corresponde a una aplicación basada en escritorio la cual permite apoyar el control de voluntarios en la asociación *Sonrisas Sin Fronteras* de diversas maneras (ingresar, modificar, eliminar, buscar, exportar, etc.) facilitando y agilizando las distintas labores en sus diferentes comunidades.

a) Mantenedor de Colonias y Comunidades: el sistema permitirá al encargado del programa administrar las diferentes colonias y comunidades establecidas en la región, es decir, Quillota, La Palma, Limache, Olmué, Colmo y La Calera. Se implementará un gestor para llevar a cabo el registro de cada una de las diferentes colonias ubicadas en alguna de las ciudades anteriormente nombradas, dando pie, posteriormente, a la mantención de las distintas colonias, permitiendo modificar, eliminar, ingresar ubicación, llenado de actividades durante el período activo, etc.
b) Gestor del Voluntariado: el sistema permitirá al encargado del sistema gestionar los diferentes datos de los voluntarios, esto significa, ingresar los datos de una persona, modificar los datos de una persona previamente registrada o eliminar los datos de una persona previamente registrada en la base de datos. Estas funciones, finalmente, entregarán una solución flexible a una de las necesidades críticas de *Sonrisas Sin Fronteras*, capturar a los voluntarios con experiencia y liderazgo.
c) Gestor de Tutores: el sistema permitirá la implementación de una gestión para el apoyo del control y cuidado de cada niño participante en las diferentes colonias y comunidades. En este, se dejará un registro ordenado de cada tutor o familiar por niño, concediendo al encargado del sistema las funciones de ingresar, modificar o eliminar datos de ellos.
d) Gestor de Niños: se diseñará e implementará en el programa un gestor que permita apoyar y agilizar los procesos relacionados a la función administrativa de cada persona asociada a los voluntarios, es decir, los niños; esto con el propósito de facilitar y

organizar el trabajo al voluntariado. Por cada niño participante en las diferentes comunidades de la zona, se hará posible su ingreso de datos, modificación de datos previamente anotados y eliminación de estos en la base de datos del programa, etc.

e) Mantenedor de Productos: el sistema posibilitará un mantenedor de inventario bastante familiar y sencillo para el encargado del sistema. Esto con el fin de poder llevar un orden y registro de los productos o bienes con los que cuenta cada colonia en su respectiva comunidad, como por ejemplo: tipos de lápices, blocks, pinturas, escobillones, etc.

f) El programa estará sujeto a la seguridad mediante la autenticación de credenciales o login para evitar el uso indebido del software. Esta ventana posibilitará la creación de un nuevo usuario y la recuperación de datos para el acceso en caso de olvido.

1.4.7 Información que se manejará

El organismo de *Sonrisas Sin Fronteras* no cuenta con un ningún sistema computacional actualmente, por lo que no tienen predefinido un formato de pantalla. Debito a lo anterior, se realizará un formato de pantalla sencillo que permita el fácil control del sistema a implementar. A continuación, se detallan las entradas y salidas del programa.

1.4.8 Entradas del Sistema

El programa debe ser capaz de recibir por teclado los siguientes datos:

- Datos de login: Ingreso del Administrador o Encargado de los voluntarios a través de un nombre de usuario y contraseña, asignando e identificando los permisos diferentes para ellos.
- Datos del voluntario: Nombre, rut, fecha de nacimiento, sexo, fecha ingreso al voluntariado.
- Datos de niños: Nombre, rut, fecha de nacimiento, sexo, dirección, fecha de ingreso.

- Datos de la colonia: Dirección, nombre de las personas que pertenezcan a la coordinación general, fecha de inicio, fecha de término, cantidad de niños participantes, presupuesto de la colonia y proyectos o actividades a realizar durante la colonia.
- Datos de la comunidad: Nombre de la comunidad, voluntarios que participen de la comunidad y actividades realizadas como comunidad.
- Datos del tutor: Nombre del tutor, rut del tutor, rut del niño a cargo, fecha en la cual fue asignado como tutor, sexo, fecha nacimiento, número de celular y dirección.

1.4.9 Salidas del Sistema

Las salidas del sistema son las siguientes:

- Listados de voluntarios por comunidad: Informe de voluntarios participantes en las distintas comunidades de la zona según el parámetro especificado.
- Listado de niños: Informe de niños que asisten a las Colonias según parámetro especificado.
- Informe sobre las distintas colonias: Da información acerca de los nombres de los voluntarios, presupuesto y actividades o proyectos realizados en la Colonia consultada.
- Hoja de vida: Informe que recauda y permite mostrar los datos del voluntario o tutor solicitado.

1.4.10 Entidades de información

A continuación, se detallan las entidades de información del modelo relacional:

1) **Tabla Voluntario:** Almacena los datos respectivos de los voluntarios participantes en la organización, siendo esta tabla, principalmente, accesada a través del Rut.

2) **Tabla Comunidad:** Corresponde al almacenamiento de la información con respecto a las comunidades activas y disponibles para la realización de actividades en tiempo de vacaciones de verano, invierno o más allá de esta calendarización. Esta tabla será ordenada según el Codigo_Comunidad.

3) **Tabla Col_Vol:** Tabla de intersección la cual contendrá información respecto al voluntario y su directa participación en cierta Colonia; permitiendo el acceso a ella a través de los campos Rut (voluntario) y Codigo_Colonia.

4) **Tabla Colonia:** Corresponde al almacenamiento de información en relación a una Colonia activa durante el período de vacaciones y ciertas dependencias de ella. Esta tabla será ordenada por el Codigo_Colonia (Codigo_comunidad + campo ID de autonumeración).

5) **Tabla Rol:** Se almacenan los diferentes roles o cargos de los voluntarios participantes en Sonrisas Sin Fronteras, siendo esta tabla ordenada por Cod_Rol.

6) **Tabla Producto:** Corresponde al registro, almacenamiento y stock de productos con los que cuenta cierta Colonia; ordenada por Cod_Colonia.

7) **Tabla Niño:** Se almacena la información con respecto a la inscripción del niño participante en Sonrisas Sin Fronteras, ordenada por Rut.

8) **Tabla Apoderado:** Se almacena la información con respecto al familiar a cargo del niño inscrito en Sonrisas Sin Fronteras, ordenada por Rut.

9) **Tabla Nin_Col:** Tabla de intersección de Niño y Colonia. En ella se almacena el registro del niño en una colonia determinada. Esta tabla será ordenada por Rut (Niño) y Codigo_Colonia (Colonia).

10) Tabla Usuario: Tabla independiente en donde se guardarán las credenciales de los usuarios registrados en el sistema. En el caso de la contraseña, esta se grabará de manera codificada. La tabla es ordenada por el campo auto numérico ID.

1.4.11 Modelo lógico de datos

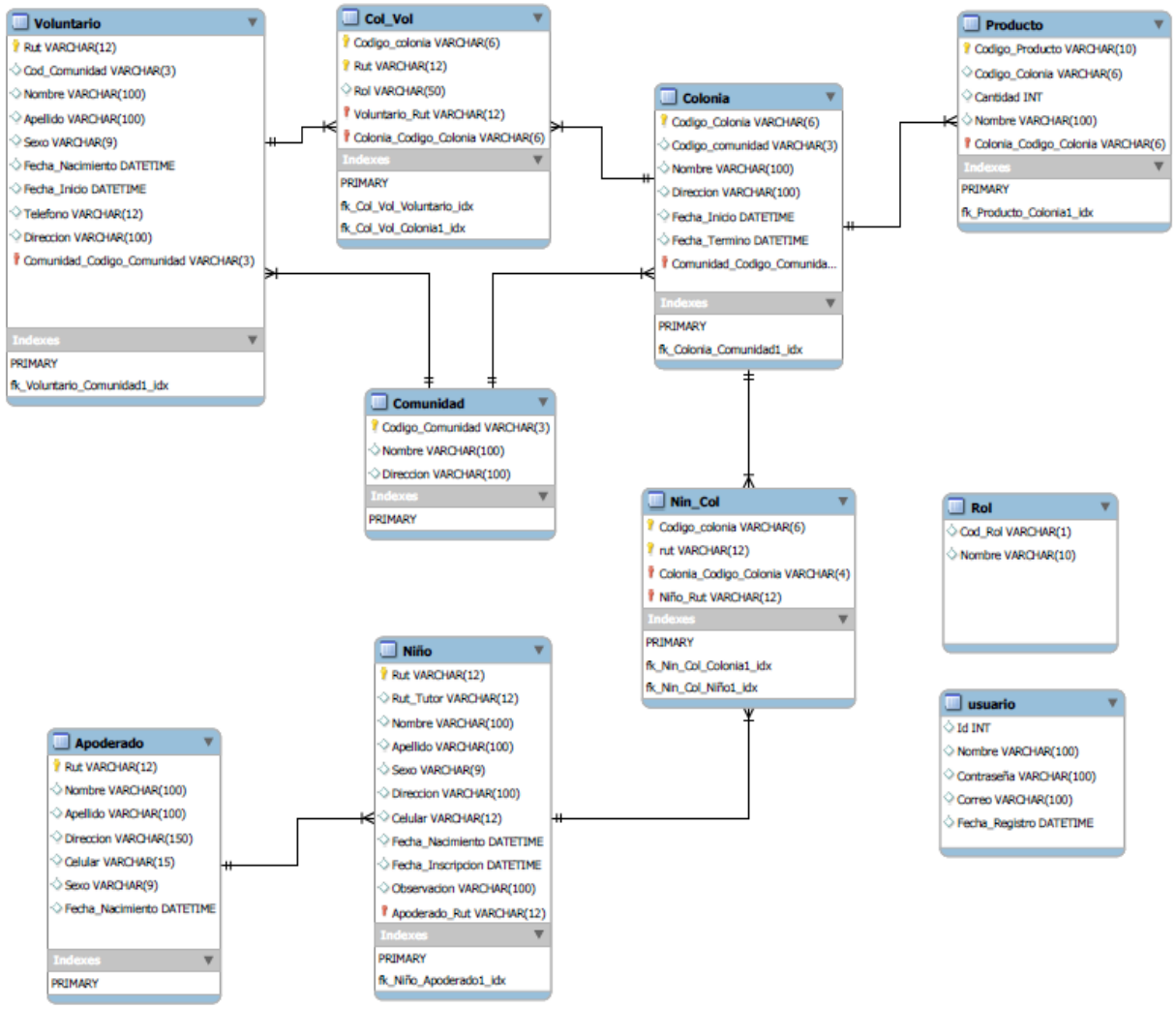


Figura 1.2: Tablas y relaciones del modelo lógico de datos.

1.4.12 Estructura de códigos

A continuación, se detallan algunos de los códigos a utilizar dentro del sistema informático:

- Cod_colonia: código compuesto por 5 caracteres en total, en donde los tres primeros corresponderán al código de la comunidad a la cual esta pertenece, y los dos siguientes a los dígitos de la colonia, por ejemplo, CAL01 (comunidad Calera).
- Rut: es un conjunto de caracteres alfanuméricos que conforman el identificador único e irrepetible de la entidad a procesar. A través de una función programada se verifica la verdadera existencia de la persona mediante el uso incluido por parámetro de este dato.
- Fecha_Nacimiento: corresponde a una cadena de caracteres procesada desde la opción seleccionada del calendario desplegado en el formulario del mantenedor del programa. A través de varias funciones programadas se valida la real existencia de la persona mediante el uso de este dato utilizado por parámetro.

CAPÍTULO II: MEDIO AMBIENTE COMPUTACIONAL Y DESCRIPCIÓN DE ARCHIVOS

MEDIO AMBIENTE COMPUTACIONAL Y DESCRIPCIÓN DE ARCHIVOS

2.1 DESCRIPCIÓN DEL RECURSO COMPUTACIONAL

Para poder desarrollar un sistema se necesitan recursos tanto materiales (computacionales) como humanos para el desarrollo íntegro y satisfactorio del sistema; los cuales se detallan a continuación.

2.1.1 Configuración del sistema

Para la puesta en marcha de la programación, implementación y ejecución del sistema basado en secuencias de algoritmos, se requiere de equipos que cumplan condiciones óptimas para asegurar un resultado positivo.

A continuación, se especifica el ambiente computacional (hardware) en donde se desarrollará e implementará el sistema propuesto.

Para el desarrollo del proyecto se utilizará un equipo con las siguientes características:

1. **Equipo Desarrollo 1**

   Computador portátil Lenovo Z40-70

   Procesador: Intel Core i5 4210U (2700 MHz)

   Memoria: 8 GB RAM

   Disco Duro: 1TB

   Sistema Operativo: Microsoft Windows 10 Home

2. **Equipo Desarrollo 2**

   Computador de escritorio

   Procesador: AMD A6 3670 (2700 MHz)

   Memoria: 8 GB RAM

   Disco Duro: 500 GB

   Sistema Operativo: Microsoft Windows 10 Professional.

El estándar de hardware existente en la organización, es una máquina que cuenta con las siguientes características:

1. **Equipo Implementación del Programa**

   Computador portátil

   Procesador: Intel Core i3 3110M (2400 MHz)

   Memoria: 2 GB RAM

   Disco Duro: 500 GB

   Sistema Operativo: Microsoft Windows 10 Professional

2.1.2 Software Utilizado

1. **Sistema Operativo**: El sistema operativo instalado en el equipo del administrador de Sonrisas Sin Fronteras es Windows 7 Home (arquitectura de 64 bits). Mientras tanto, ambos equipos desarrolladores cuentan con Windows 10 Home (arquitectura de 64 bits).

2. **Descripción del lenguaje a utilizar:** El lenguaje informático a utilizar para la creación del sistema será Visual Basic .NET.

- Visual Basic .NET (VB.NET) es un lenguaje de programación orientado a objetos que se puede considerar una evolución de Visual Basic implementada sobre el framework .NET.

  Se publicó en 2002 y fue la primera versión de Visual Studio en introducir el framework .NET. Esta versión de Visual Studio introdujo, junto con el Framework .NET tres nuevos lenguajes de programación, Visual C#, VB.NET y Visual J#. En esta primera versión de Visual Studio .NET se podían programar aplicaciones Windows.Forms (aplicaciones de escritorio) y aplicaciones ASP.NET (Aplicaciones Web).

  La sintaxis que utiliza este lenguaje de programación proviene de su predecesor, pero completada con comandos y códigos de otros lenguajes más modernos e

incorporando un apartado dedicado a la POO (programación orientada a objetos). Además, es un lenguaje muy apropiado para el manejo de bases de datos, ya que incorpora un asistente que, tras una serie de sencillos pasos, establece la conexión local con el repositorio a utilizar, en este caso, una base de datos Access. Esta combinación proporciona una potente herramienta que se acomoda a las necesidades particulares del proyecto.

3. **Descripción del motor de base de datos:** El motor de la base de datos encargado del funcionamiento del proyecto y la administración de las consultas requeridas es Microsoft Jet. Sus características se mencionan a continuación.

- Un motor de base de datos es el componente subyacente de una base de datos, una colección de información almacenada en una computadora de manera sistemática. La primera versión de Jet fue desarrollada en 1992, consistiendo en tres módulos que se podrían utilizar para manipular una base de datos.

  Actualmente es una implementación de base de datos ampliamente utilizada por las aplicaciones de Microsoft Office. Permite la conectividad entre distintas aplicaciones, sin importar su tipo ni ubicación, y puede ser utilizado por Microsoft Access, Microsoft Visual Basic, etc.

  El motor de base de datos Microsoft Jet proporciona un modelo relacional basado en el Lenguaje de Consulta Estructurado (SQL) estándar, utilizándose fundamentalmente para ejecutar consultas de manipulación y definición de datos mediante los objetos correspondientes de las bibliotecas de DAO y ADO.

2.2 DESCRIPCIÓN DE ARCHIVOS

2.2.1 Comunidad

Nombre Lógico: Comunidad

Nombre Físico: Comunidad

Descripción: Entidad que almacenará los principales datos de las comunidades activas.

Clave Principal: Codigo_Comunidad

Clave(s) Foránea(s): -

Tabla 2-1. Descripción de registro Comunidad

Nombre	Tipo	Longitud	Descripción
Codigo_Comunidad	Varchar	3	Código respectivo de la comunidad.
Nombre	Varchar	100	Nombre de la comunidad.
Direccion	Varchar	100	Ubicación física de la comunidad en su respectiva zona.

2.2.2 Voluntario

Nombre Lógico: Voluntario

Nombre Físico: Voluntario

Descripción: Entidad que almacenará los datos de los diferentes voluntarios inscritos en Sonrisas Sin Fronteras.

Clave Principal: Rut

Clave(s) Foránea(s): Cod_Comunidad (referencia a tabla Comunidad).

Tabla 2-2. Descripción de registro de Voluntario

Nombres	**Tipo**	**Longitud**	**Descripción**
Rut	Varchar	12	Rut de voluntario.
Cod_Comunidad	Varchar	3	Código de referencia a la tabla comunidad.
Nombre	Varchar	100	Nombre del voluntario.
Apellido	Varchar	100	Apellidos del voluntario
Sexo	Varchar	9	Sexo del voluntario.
Fecha_nacimiento	Date		Fecha de nacimiento del voluntario.
Fecha_inicio	Date		Fecha de ingreso del voluntario.
Telefono	Varchar	12	Teléfono del voluntario
Dirección	Varchar	100	Dirección del voluntario

2.2.3 Colonia

Nombre Lógico: Colonia

Nombre Físico: Colonia

Descripción: Entidad que almacenará los principales datos de las colonias activas.

Clave Principal: Codigo_Colonia

Clave(s) Foránea(s): Codigo_comunidad (referencia a tabla Comunidad).

Tabla 2-3. Descripción de registro de Colonia

Nombres	Tipo	Longitud	Descripción
Codigo_Colonia	Varchar	6	Código respectivo de la colonia.
Codigo_comunidad	Varchar	3	Código de referencia a la tabla Comunidad.
Nombre	Varchar	100	Nombre de la colonia.
Fecha_inicio	Date		Fecha del comienzo de temporada de colonia.
Fecha_termino	Date		Fecha de término de temporada de colonia.
Direccion	Varchar	100	Dirección física o ubicación de la colonia.

2.2.4 Col_Vol

Nombre Lógico: Voluntarios asignados en Colonias

Nombre Físico: Col_Vol

Descripción: Entidad de intersección que almacenará voluntarios inscritos y colonias activas.

Clave Principal: Codigo_colonia (colonia) + Rut (voluntario)

Clave(s) Foránea(s): FK1: Rut (referencia a tabla Voluntario)

FK2: Codigo_colonia (referencia a tabla Colonia)

Tabla 2-4. Descripción de registro de intersección Col_Vol

Nombres	Tipo	Longitud	Descripción
Rut	Varchar	12	Rut de referencia a la tabla Voluntario.
Codigo_colonia	Varchar	6	Código de referencia a la tabla Colonia.
Rol	Varchar	50	Nombre respectivo al rol o cargo.

2.2.5 Rol

Nombre Lógico: Rol

Nombre Físico: Rol

Descripción: Entidad que almacenará los diferentes roles de los voluntarios.

Clave Principal: Cod_Rol

Clave(s) Foránea(s): Sin clave foránea

Tabla 2-5. Descripción de registro Rol

Nombres	Tipo	Longitud	Descripción
Cod_Rol	Varchar	1 (auto numérico de uno a siete)	Código respectivo al rol o cargo.
Nombre	Varchar	50	Nombre respectivo al rol o cargo.

2.2.6 Producto

Nombre Lógico: Producto

Nombre Físico: Producto

Descripción: Entidad que almacenará los diferentes productos adquiridos y con los que cuenta la colonia.

Clave Principal: Codigo_Producto

Clave(s) Foránea(s): Codigo_Colonia (referencia a tabla Colonia.).

Tabla 2-6. Descripción de registro Producto

Nombres	Tipo	Longitud	Descripción
Codigo_Producto	Varchar	10	Código respectivo del producto.
Codigo_Colonia	Varchar	6	Código respectivo de la colonia perteneciente en la tabla Colonia.
Nombre	Varchar	100	Nombre del producto.
Cantidad	INT		Cantidad del producto.

2.2.7 Niño

Nombre Lógico: Niño

Nombre Físico: Niño

Descripción: Entidad que almacenará los diferentes datos de los niños participantes en la colonia.

Clave Principal: Rut

Clave(s) Foránea(s): Rut_Tutor (referencia a tabla Apoderado).

Tabla 2-7. Descripción de registro Niño

Nombres	Tipo	Longitud	Descripción
Rut	Varchar	12	Rut del niño.
Rut_Tutor	Varchar	12	Rut del tutor de referencia a la tabla Tutor.
Nombre	Varchar	100	Nombre del niño.
Apellido	Varchar	100	Apellido del niño.
Fecha_Nacimiento	Date		Fecha de nacimiento del niño.
Sexo	Varchar	9	Sexo del niño.
Fecha_Inscripcion	Date		Fecha de ingreso del niño.
Direccion	Varchar	100	Dirección de vivienda del niño.
Celular	Varchar	12	Teléfono del niño.
Observacion	Varchar	100	Discapacidades o enfermedades que presente el niño.

2.2.8 Apoderado

Nombre Lógico: Apoderado

Nombre Físico: Apoderado

Descripción: Entidad que almacenará los diferentes tutores (familiar a cargo) de los niños participantes en una colonia.

Clave Principal: Rut

Clave(s) Foránea(s): Sin clave foránea.

Tabla 2-8. Descripción de registro Apoderado

Nombres	Tipo	Longitud	Descripción
Rut	Varchar	12	Rut del tutor.
Nombre	Varchar	100	Nombre del tutor.
Apellido	Varchar	100	Apellidos del tutor.
Fecha_Nacimiento	Date		Fecha de nacimiento del tutor.
Sexo	Varchar	9	Sexo del tutor.
Celular	Varchar	15	Teléfono de contacto del tutor.
Direccion	Varchar	150	Dirección de vivienda del tutor.

2.2.9 Nin_Col

Nombre Lógico: Niños asignados en Colonias

Nombre Físico: Nin_Col

Descripción: Entidad que almacenará los diferentes niños inscritos en una colonia activa.

Clave Principal: Codigo_colonia (colonia) + rut (niño)

Clave(s) Foránea(s): FK1: Codigo_colonia (referencia a la tabla Colonia)

FK2: rut (referencia a la tabla Niño)

Tabla 2-9. Descripción de registro Nin_Col

Nombres	Tipo	Longitud	Descripción
Codigo_colonia	Varchar	6	Codigo respectivo de la colonia perteneciente a la tabla Colonia.
rut	Varchar	12	Rut del niño perteneciente a la tabla Niño.

2.2.10 Usuario

Nombre Lógico: Usuario

Nombre Físico: usuario

Descripción: Entidad que almacenará las diferentes credenciales de los usuarios registrados en el sistema.

Clave Principal: ID (valor autonumérico)

Clave(s) Foránea(s): Sin clave foránea.

Tabla 2-10. Descripción de registro Usuario

Nombres	Tipo	Longitud	Descripción
Id	Int		Valor autonumérico para diferenciar a cada usuario.
Nombre	Varchar	100	Nombre del usuario.
Contraseña	Varchar	100	Contraseña del usuario.
Correo	Varchar	100	Correo electrónico del usuario.
Fecha_Registro	Date		Fecha correspondiente al grabado de datos del usuario.

CAPÍTULO III: DESCRIPCIÓN DE PROGRAMAS

DESCRIPCIÓN DE PROGRAMAS

3.1 DIAGRAMA DE MENÚS

A continuación, se presenta el diagrama de menú para visualizar el sistema computacional en términos generales, contextualizando sus funcionalidades de manera sencilla a través de la figura.

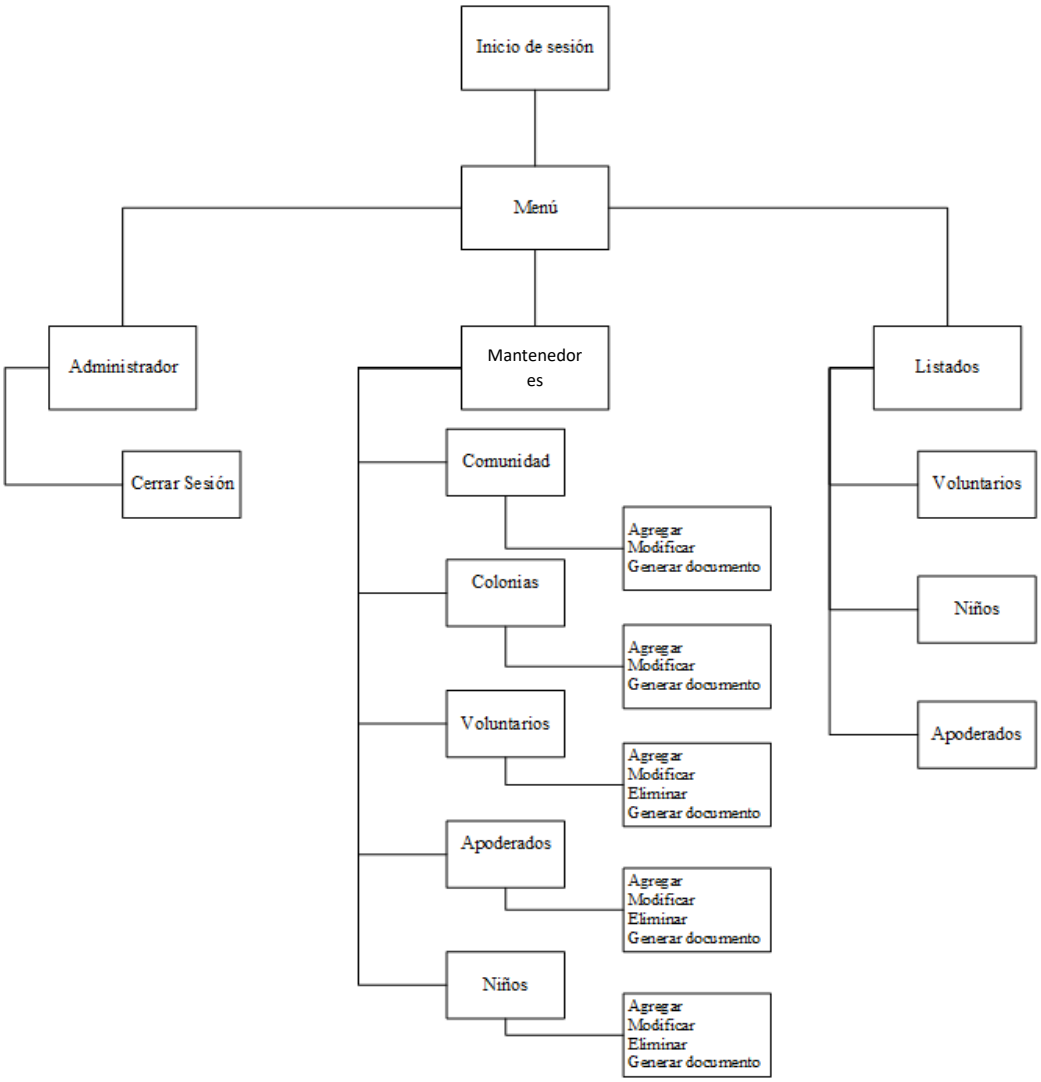


Figura 3.1: Diagrama de menú: sistema de apoyo para el control de voluntarios.

3.1.1 DIAGRAMA DE MENÚ – INICIO DE SESIÓN

Para un uso seguro del programa es que se decidió diseñar un Menú Login o Inicio de sesión, el cual permitirá ingresar (por campos separados) el usuario y la contraseña de la persona que esté accediendo al sistema de escritorio, únicamente la persona a cargo del programa entregado a Sonrisas Sin Fronteras. Para un acceso satisfactorio es obligatorio que la persona cuente con sus credenciales grabadas con anterioridad en la base de datos para que la vista posterior sea desplegada.

3.1.2 DIAGRAMA DE MENÚ – MENÚ

Se ha decidido crear una vista menú que sirva de apoyo para el manejo ordenado y simplificado de las diferentes funcionalidades otorgadas por el sistema informático. En esta vista se implementarán pestañas navegables, proyectando en cada una de ellas las diferentes utilidades para ser accesadas mediante esta vía: operar con el registro de usuarios y/o credenciales para el uso del programa, realizar consultas o manipular datos en los formularios incluidos en los mantenedores (voluntarios, niños, apoderados, colonias, comunidades, etc).

3.1.3 DIAGRAMA DE MENÚ – ADMINISTRADOR

Una de las pestañas navegables, correspondiente al menú principal del sistema, es la llamada Administrador, la cual desplegará diferentes opciones seleccionables, en donde una de ellas, redireccionará hacia un menú a la persona a cargo, otorgando el poder al mismo de generar (o crear) otros usuarios que puedan acceder y, posteriormente, intervenir en el programa, realizando las mismas acciones todos por igual (sin distinción de privilegios y personas que manejen el mismo ordenador en donde estará ejecutándose el programa) en caso de que él no se encuentre disponible. Estas credenciales quedarán registradas en la base de datos para la futura captura y así lograr un exitoso inicio de sesión.

3.1.4 DIAGRAMA DE MENÚ – MANTENEDORES

Desde el menú principal también será posible acceder a los diferentes mantenedores provistos para el control de operaciones específicas a través de botones con la instrucción clara indicando qué tarea realizará cada uno de ellos, como el registro y eliminación de entidades pertenecientes a la organización.

Cada uno de los mantenedores (comunidad, colonia, voluntarios, apoderado, niño y producto) facilitará a la persona a cargo del software la particular función de agregar, modificar o eliminar la entidad según corresponda. Además, en cada uno de ellos, se obtendrá una tabla ordenada y separada por diversos campos los cuales se actualizarán inmediatamente tras sufrir alguna alteración de sus registros guardados en la base de datos.

3.1.5 DIAGRAMA DE MENÚ – LISTADOS

Esta opción del menú navegable, permitirá al usuario poder listar según sea el requerimiento o parámetro a partir de su criterio, en las diferentes tablas de los mantenedores de voluntarios o niños.

3.2 DIAGRAMA MODULAR

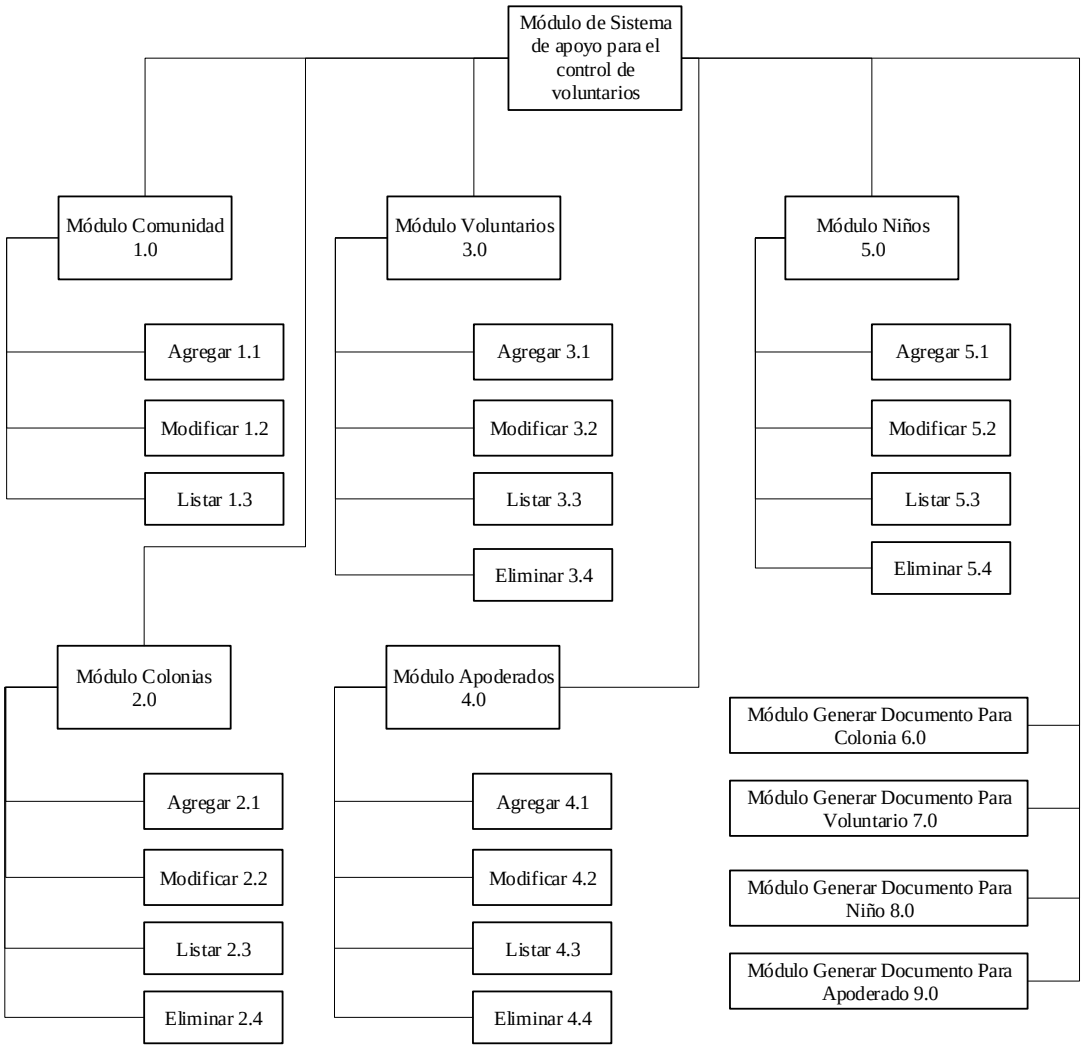


Figura 3.2: Módulos del sistema: mantenedores y funcionalidades varias.

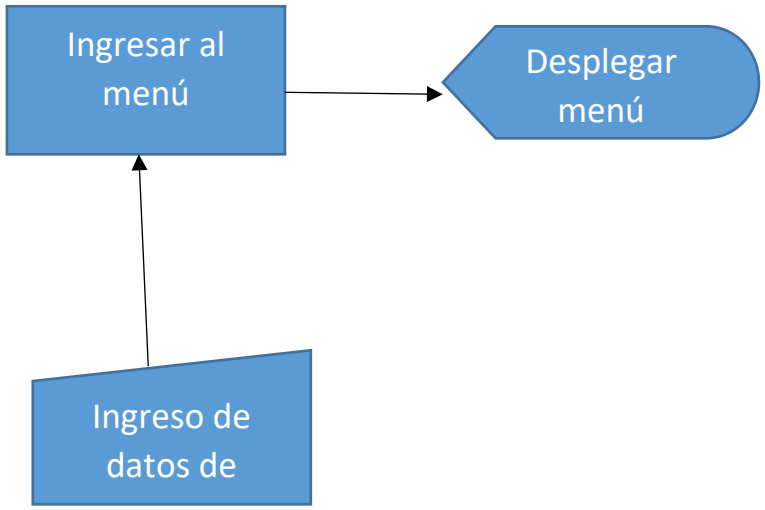
3.3 LISTA DE PROGRAMAS DESARROLLADOS

A continuación, se presenta la lista de programas que fueron desarrollados para llevar a cabo el proyecto "Sistema de apoyo para el control de voluntarios en SSF" en su totalidad. Posteriormente, se realizará la especificación de una parte de ellos, considerando los programas más relevantes del software.

Nombre del programa	Objetivos
FrmLogin.vb*	Menú inicial del sistema. Permite la autenticación de datos de un usuario previamente registrado.
FrmCrearUsuario.vb*	Permite el ingreso de datos para el registro de un nuevo usuario en el sistema.
FrmListarUsiarios.vb	Permite navegar visualizando los datos de los usuarios registrados. Otorga eliminaciones de ellos opcionalmente.
FrmModificarUsuario.vb	Permite modificar los datos del usuario que se encuentra haciendo uso del programa.
FrmRecuperarUsuario.vb	Permite el ingreso de datos para reestablecer la cuenta de un usuario previamente registrado.
Menu.vb*	Menú principal del sistema en donde se encuentran las operaciones principales para controlar el voluntariado.
FrmComunidad.vb	Menú principal de las comunidades. Permite la visualización de ellas y otorga funciones varias.
AgregarComunidad.vb*	Permite el ingreso de datos para la incorporación de una nueva comunidad en el sistema.
ModificarComunidad.vb	Posibilita la modificación de datos de una comunidad previamente registrada.
FrmColonia.vb	Menú principal de las colonias. Permite la visualización de ellas y operaciones varias.
AgregarColonia.vb*	Menú principal de las colonias. Permite la visualización de ellas y funcionalidades varias.
ModificarColonia.vb	Permite la modificación de datos de una colonia previamente registrada y activa en una zona.
AgregarNiñoAColonia.vb	Posibilita la visualización de los niños registrados en el sistema y la posterior inscripción de ellos en una determinada colonia.
AgregarVoluntarioAColonia.vb*	Posibilita la visualización de los voluntarios registrados en el sistema e inscripción de ellos en una determinada colonia.
FrmApoderados.vb	Menú principal de los apoderados. Permite la visualización de ellos y operaciones varias.
AgregarApoderado.vb*	Permite el ingreso de datos de un apoderado para registrarlo en el sistema.
ModificarApoderados.vb	Permite la modificación de datos de un apoderado anteriormente inscrito en el sistema.
FrmNiños.vb	Menú principal de los niños. Permite la visualización de ellos y operaciones varias.
AgregarNiño.vb*	Permite el ingreso de datos de un niño para registrarlo en el sistema.
ModificarNiño.vb	Permite la modificación de datos de un niño anteriormente inscrito en el sistema.
FrmVoluntario.vb	Menú principal de los voluntarios. Permite la visualización de ellos y operaciones varias.
AgregarVoluntario.vb*	Permite el ingreso de datos de un voluntario para registrarlo en el sistema.
ModificarVoluntario.vb	Permite la modificación de datos de un voluntario anteriormente inscrito en el sistema.
Hojadevida.vb	Permite navegar entre los voluntarios inscritos y generar un documento imprimible adjuntando información relacionada de uno de ellos.
MenuProducto.vb	Menú principal de los productos. Permite la visualización de ellos y operaciones varias.
AgregarProducto.vb	Permite el ingreso de datos de un nuevo producto para ser registrado en el sistema.
ModificarProducto.vb*	Permite la modificación de datos de un producto anteriormente registrado en el sistema.
ListadoNiños.vb	Menú principal del listado de la entidad Niños. Permite la selección entre ciertos criterios.
ListadoNiñosColonia.vb	Permite listar la totalidad de niños inscritos en una determinada colonia.
ListadoNiñosSexo.vb	Permite listar la totalidad de niños registrados en el sistema según su sexo.
ListadoVoluntarios.vb	Menú principal del listado de la entidad Voluntarios. Permite la selección entre ciertos criterios.
ListadoVoluntariosColonia.vb*	Permite listar la totalidad de voluntarios inscritos en una determinada colonia.
ListadoVoluntariosComunidad.vb	Permite listar la totalidad de voluntarios registrados en el sistema según la comunidad.
ListadoVoluntariosNombre.vb	Permite listar la totalidad de voluntarios registrados en el sistema según su nombre o apellido.
ListadoVoluntariosSexo.vb	Permite listar la totalidad de voluntarios registrados en el sistema según su sexo.

3.4 ESPECIFICACIÓN DE PROGRAMAS

3.4.1 PROGRAMA: MENÚ PRINCIPAL

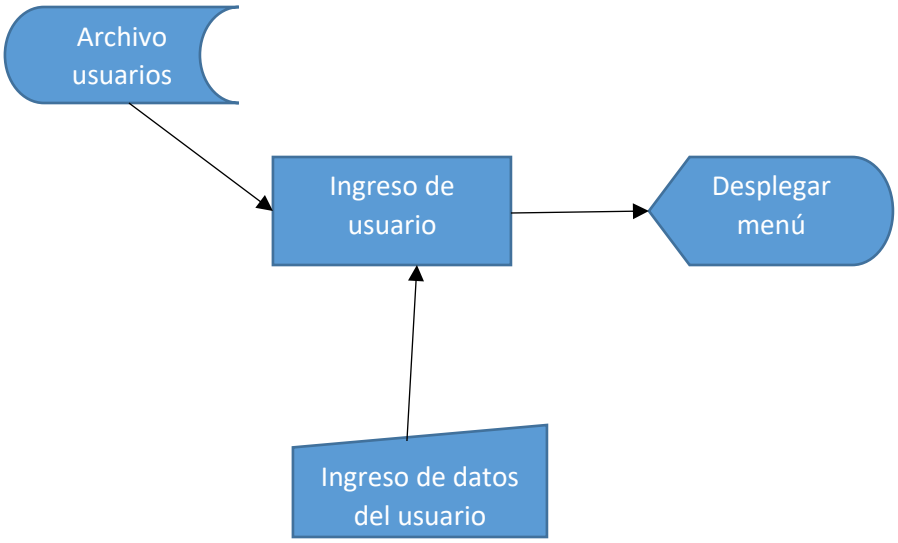
Menú principal del sistema		
	Nombre del programa	Menu.vb
	Objetivo	Interfaz que otorga la interacción con las principales funcionalidades del sistema, es decir, operar con los mantenedores de las entidades del sistema.
	Reglas del proceso	Tras validarse el login se efectúa la carga de diferentes funcionalidades en una barra de navegación ubicada en la parte superior de la ventana, de las cuales solo una puede ser seleccionada. En efecto, la elección influirá directamente en el resultado de la salida del proceso, invocando al mantenedor respectivo y ocultando este menú.
	Documentación del código	Páginas 61 - 63
	Diagrama de bloque	

3.4.1.1 DISEÑO DE PANTALLAS: MENÚ PRINCIPAL

Figura 3.3: Diseño de la pantalla correspondiente al menú principal del programa.

3.4.2 PROGRAMA: LOGIN

Acceso por login		
	Nombre del programa	FrmLogin.vb
	Objetivo	Interfaz que permite el ingreso de datos para el acceso de un usuario al sistema, otorgando seguridad a este último.
	Reglas del proceso	Mediante el ingreso de los datos de acceso de la persona (usuario y contraseña) compara estos con los ya existentes en la tabla Usuario, obteniendo un acceso satisfactorio o inválido. En caso de dejar los campos en blanco se despliega un mensaje notificando el error, así mismo para cuando el formulario esté completo pero sin coincidencias con los registros de la tabla. Este programa quedará oculto y tendrá como salida el menú principal al momento en que el usuario acceda con datos válidos, siendo este notificado a través de un mensaje de bienvenida.
	Documentación del código	Páginas 64 - 66

	Diagrama de bloque	

3.4.2.1 DISEÑO DE PANTALLAS: LOGIN

Figura 3.4: Diseño de la pantalla correspondiente al formulario login del programa.

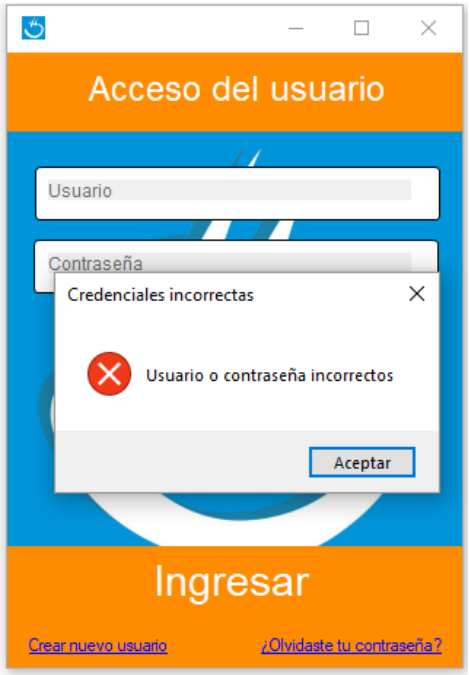
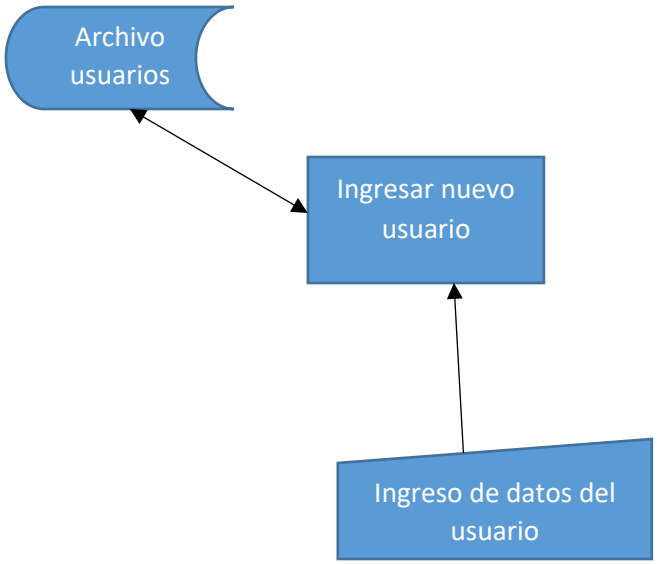


Figura 3.5: Diseño de la pantalla correspondiente al formulario login del programa (autenticación inválida)

3.4.3 PROGRAMA: NUEVO USUARIO

Crear un nuevo usuario	
Nombre del programa	FrmCrearUsuario.vb
Objetivo	Otorga un formulario para el ingreso de datos de un nuevo usuario próximo a utilizar el sistema.
Reglas del proceso	La verificación o comparación de los datos requiere consultar la tabla Usuario, actuando como entrada del programa. Cabe destacar que un exitoso registro se lleva a cabo cuando: todos los campos están rellenados, el nombre de usuario escrito no exista anteriormente (esto se logra comparándolo con los ya almacenados en la tabla Usuario), las contraseñas escritas coincidan entre ellas, el correo electrónico tenga un formato válido y sea equivalente al ingresado posteriormente.
Documentación del código	Páginas 67 - 69

	Diagrama de bloque	

3.4.3.1 DISEÑO DE PANTALLAS: NUEVO USUARIO

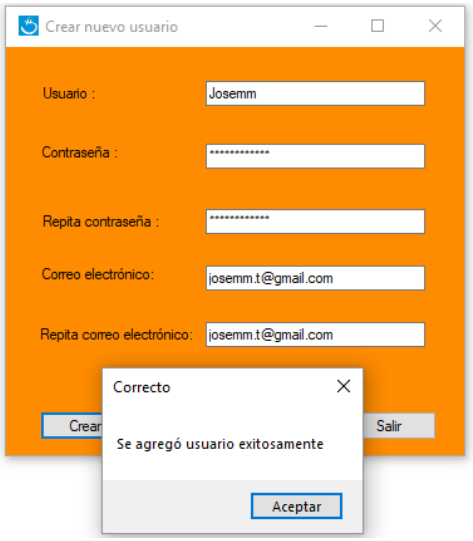


Figura 3.6: Diseño de la pantalla correspondiente a la creación de un nuevo usuario (caso exitoso).

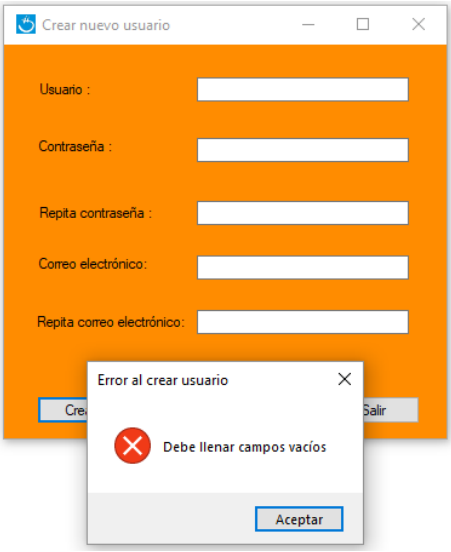
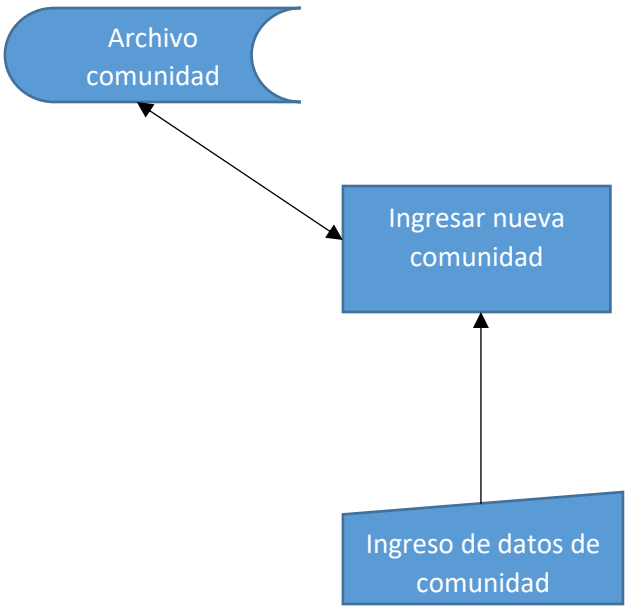


Figura 3.7: Diseño de la pantalla correspondiente a la creación de un nuevo usuario (caso sin éxito)

3.4.4 PROGRAMA: NUEVA COMUNIDAD

Crear una nueva comunidad	
Nombre del programa	AgregarComunidad.vb
Objetivo	Otorga un formulario para ingresar los datos de una nueva comunidad asociada al proyecto.
Reglas del proceso	Inicialmente, el formulario interactuará con la tabla Comunidad recuperando los códigos de aquellas entidades registradas con anterioridad para ser desplegadas en pantalla. Para que el registro sea exitoso y grabado en la tabla Comunidad, debe considerarse llenar todos los campos. En caso contrario, el usuario será notificado a través de un cuadro de mensaje con el error correspondiente. No permite duplicaciones, es decir, se invalida la creación de una nueva entidad si llegase a repetirse un código de comunidad previamente grabado en la tabla Comunidad.
Documentación del código	Páginas 70 - 71

	Diagrama de bloque	

3.4.4.1 DISEÑO DE PANTALLAS: NUEVA COMUNIDAD

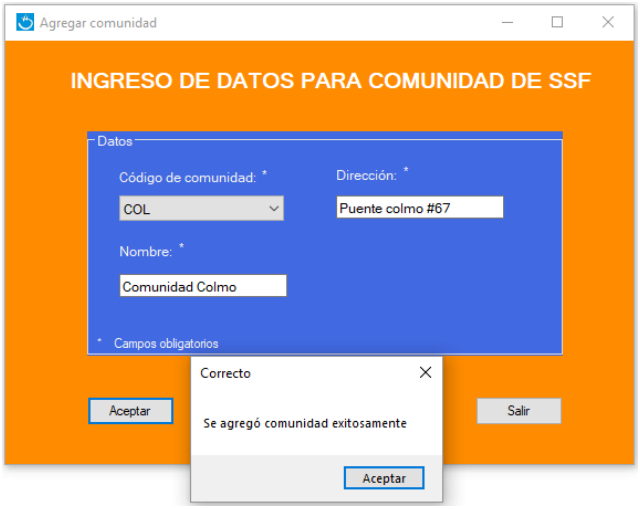


Figura 3.8: Diseño de la pantalla correspondiente a la creación de una nueva comunidad (caso exitoso).

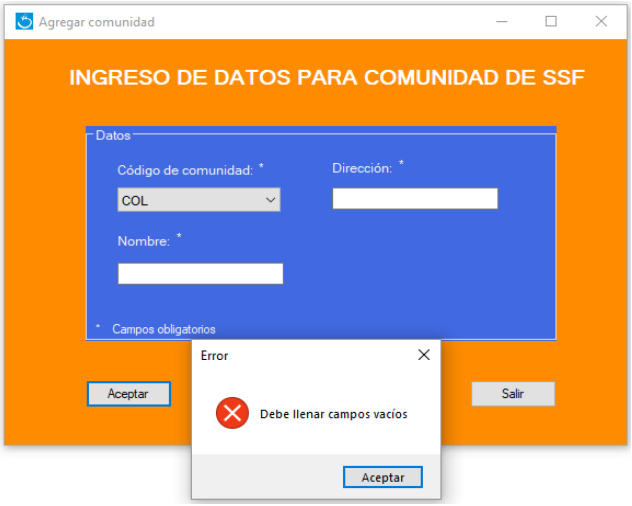
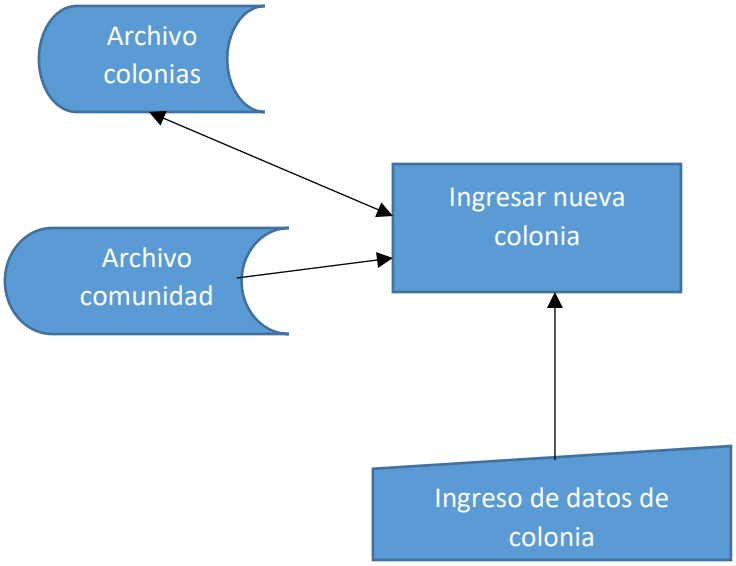


Figura 3.9: Diseño de la pantalla correspondiente a la creación de una nueva comunidad (caso sin éxito).

3.4.5 PROGRAMA: CREAR NUEVA COLONIA

Crear una nueva colonia	
Nombre del programa	AgregarColonia.vb
Objetivo	Otorga un formulario para ingresar los datos de una nueva colonia asociada al proyecto.
Reglas del proceso	Inicialmente, el formulario interactuará con la tabla Comunidad para recuperar los códigos de todas aquellas entidades registradas con anterioridad, posibilitando el despliegue de ellas en la pantalla. Para que el registro sea exitoso y grabado en la tabla Colonia, debe considerarse llenar todos los campos marcados como obligatorios, que la fecha inicial corresponda a la actual o una futura y que la fecha de término no exceda en más de siete días a la de inicio. En caso contrario, el usuario será notificado a través de un cuadro de mensaje indicando el error correspondiente.
Documentación del código	Páginas 72 - 74

	Diagrama de bloque	

3.4.5.1 DISEÑO DE PANTALLAS: NUEVA COLONIA

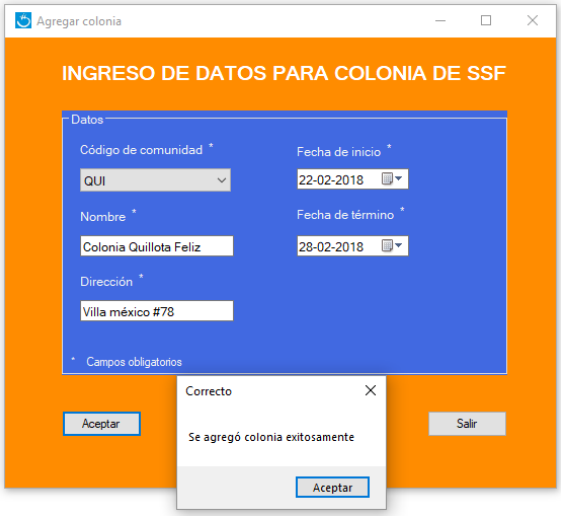


Figura 3.10: Diseño de pantalla correspondiente a la creación de una nueva colonia (caso exitoso).

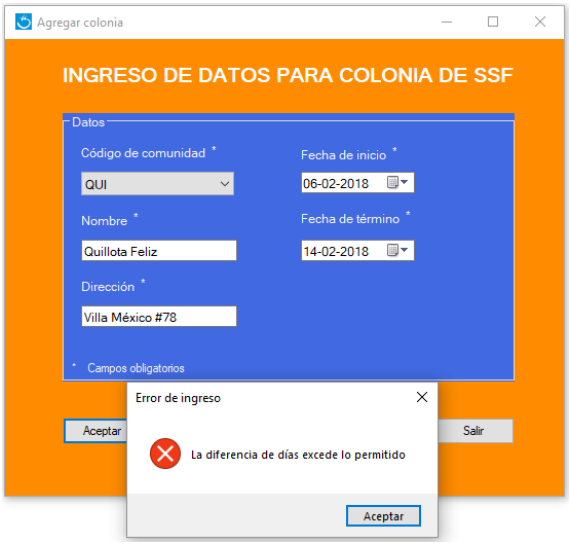
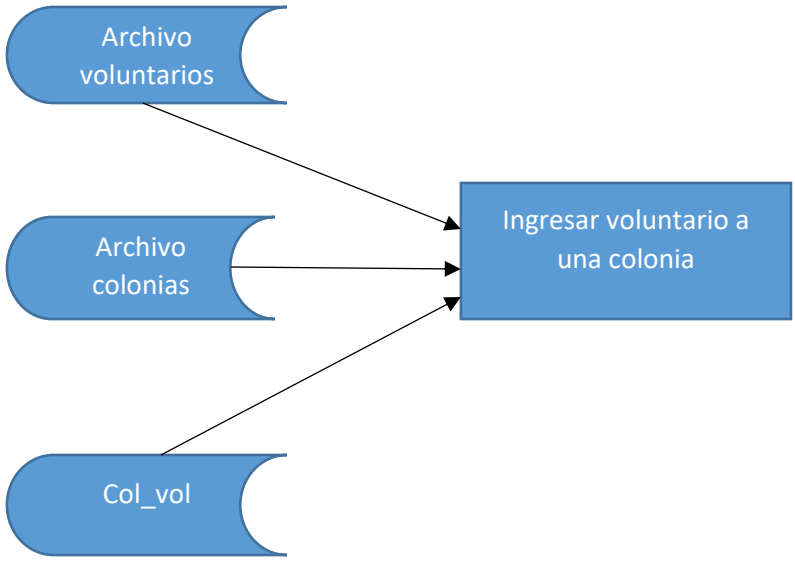


Figura 3.10: Diseño de pantalla correspondiente a la creación de una nueva colonia (caso sin éxito).

3.4.6 PROGRAMA: AGREGAR UN VOLUNTARIO A UNA COLONIA

Agregar un voluntario (previamente registrado) a una colonia		
	Nombre del programa	AgregarVoluntarioAColonia.vb
	Objetivo	Permite visualizar todos los voluntarios inscritos y registrarlos en una determinada colonia.
	Reglas del proceso	Inicialmente, el programa interactuará con la tabla Colonia, Voluntario y Rol para recuperar los códigos de las colonias previamente registradas, los voluntarios inscritos con todos sus datos adjuntos y los códigos de rol almacenados en la BD. Un registro satisfactorio es consecuencia de la selección de un voluntario, la selección de un ítem por cada opción y que la persona no se repita en la colonia asignada. Si esto es válido, la entidad quedará registrada en la tabla Col_Vol con el código de colonia, rut y rol.
	Documentación del código	Páginas 75 - 77

	Diagrama de bloque	

3.4.6.1 DISEÑO DE PANTALLAS: AGREGAR VOLUNTARIO A COLONIA

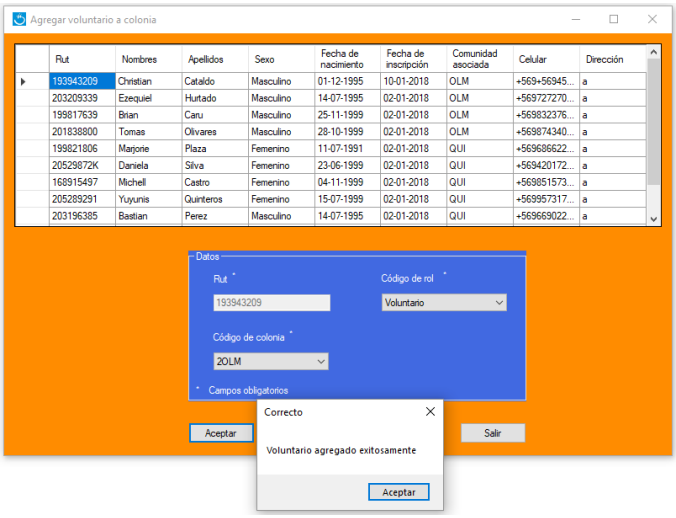


Figura 3.11: Diseño de pantalla correspondiente a la inserción de un voluntario a una colonia (caso exitoso).

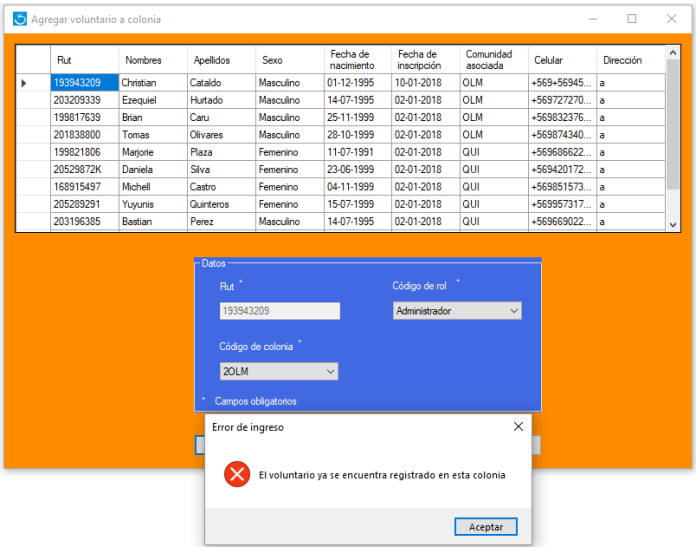
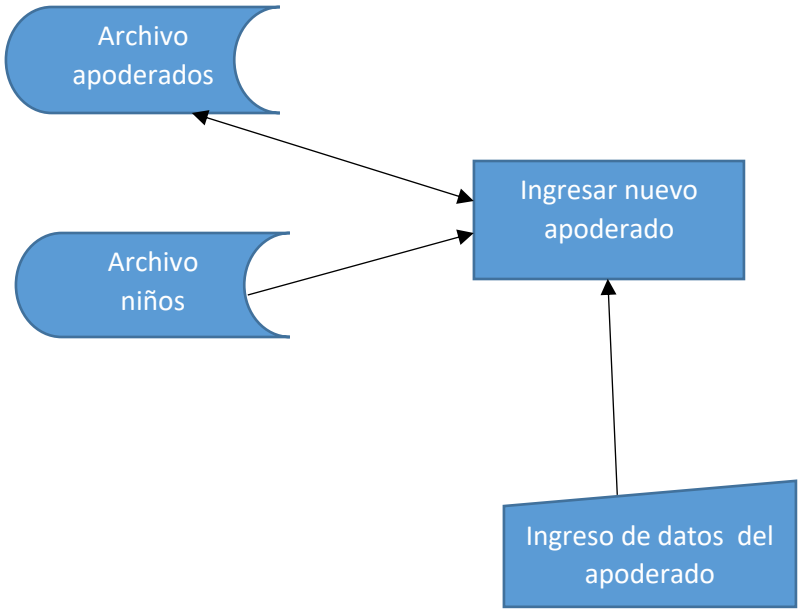


Figura 3.12: Diseño de pantalla correspondiente a la inserción de un voluntario a una colonia (caso sin éxito).

3.4.7 PROGRAMA: NUEVO APODERADO

Registrar un nuevo apoderado	
Nombre del programa	AgregarApoderado.vb
Objetivo	Otorga un formulario para ingresar los datos de un nuevo apoderado o tutor del infante.
Reglas del proceso	Inicialmente, el formulario interactuará con la tabla Apoderado sirviendo de apoyo para todo tipo de validaciones. Un registro exitoso se lleva a cabo solo si se logran llenar todos los campos marcados como obligatorios, el rut posee un formato real, la fecha de nacimiento no excede de la actual, el apoderado es mayor de edad, la cantidad de dígitos del celular no es inferior a ocho y no existan duplicaciones de apoderado (rut anteriormente almacenado en la tabla respectiva). En caso contrario, el usuario del programa será notificado a través de un cuadro de mensaje indicándole el error correspondiente.
Documentación del código	Páginas 78 - 80

	Diagrama de bloque	

3.4.7.1 DISEÑO DE PANTALLAS: NUEVO APODERADO

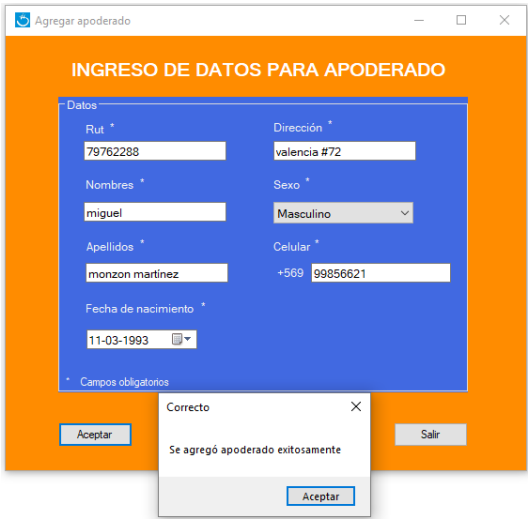


Figura 3.13: Diseño de pantalla correspondiente al ingreso de un nuevo apoderado (caso exitoso).

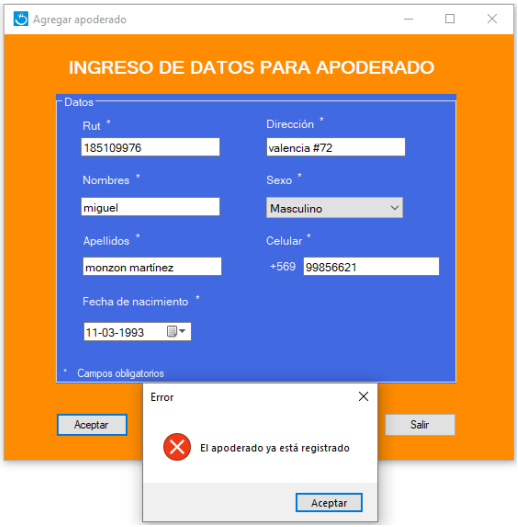
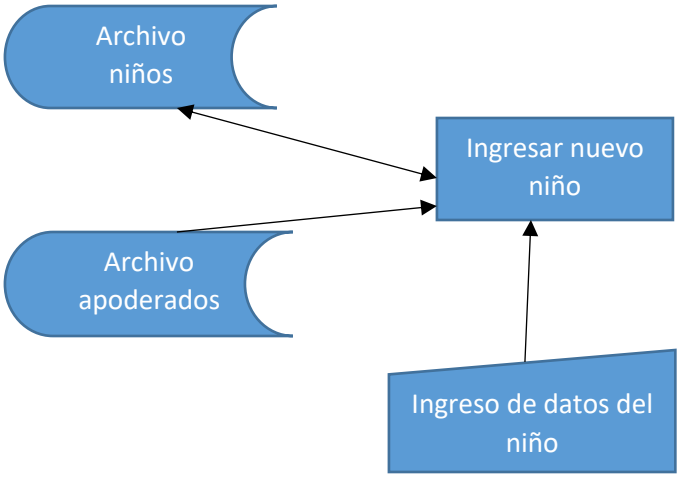


Figura 3.14: Diseño de pantalla correspondiente al ingreso de un nuevo apoderado (caso sin éxito).

3.4.8 PROGRAMA: NUEVO NIÑO

Registrar un nuevo niño	
Nombre del programa	AgregarNiño.vb
Objetivo	Otorga un formulario de registro para ingresar los datos de un nuevo niño.
Reglas del proceso	Inicialmente, el formulario interactuará con la tabla Apoderado y Niño. Ambas servirán de apoyo para las validaciones a especificar. Un registro exitoso es almacenado en la tabla Niño y se lleva a cabo solo si todos los campos marcados como obligatorios son completados, el rut posee un formato real, la fecha de nacimiento no excede de la actual, la edad del niño se encuentre entre el rango de cuatro y catorce años, su apoderado se encuentre registrado previamente y el formato del rut sea congruente, su fecha de inscripción corresponda a la actual o a un tiempo pasado, la cantidad de dígitos del celular no sea inferior a ocho y no se hallen duplicaciones del mismo niño (rut anteriormente almacenado en la tabla respectiva). En caso contrario, el usuario del programa será notificado a través de un cuadro de mensaje indicándole el error pertinente.
Documentación del código	Páginas 81 - 86

	Diagrama de bloque	

3.4.8.1 DISEÑO DE PANTALLAS: NUEVO NIÑO

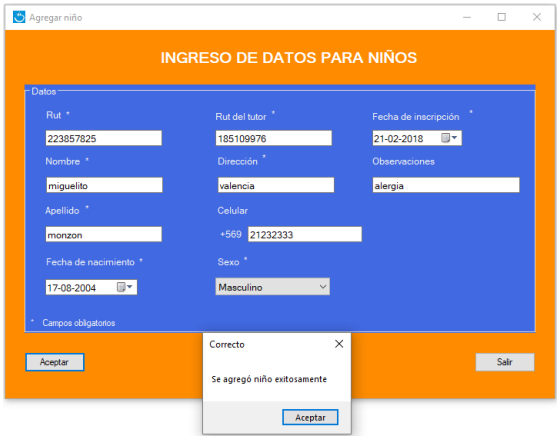


Figura 3.15: Diseño de pantalla correspondiente al ingreso de un nuevo niño (caso exitoso).

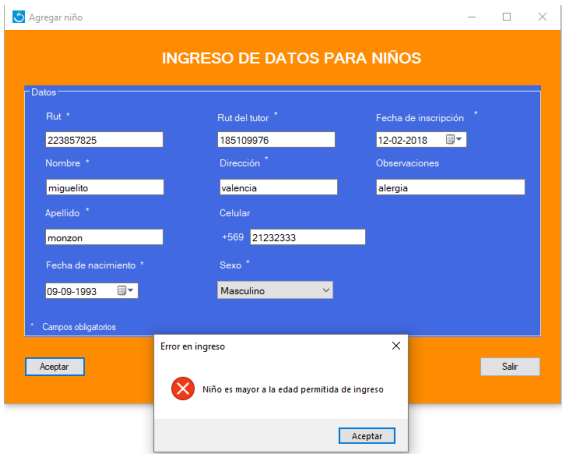
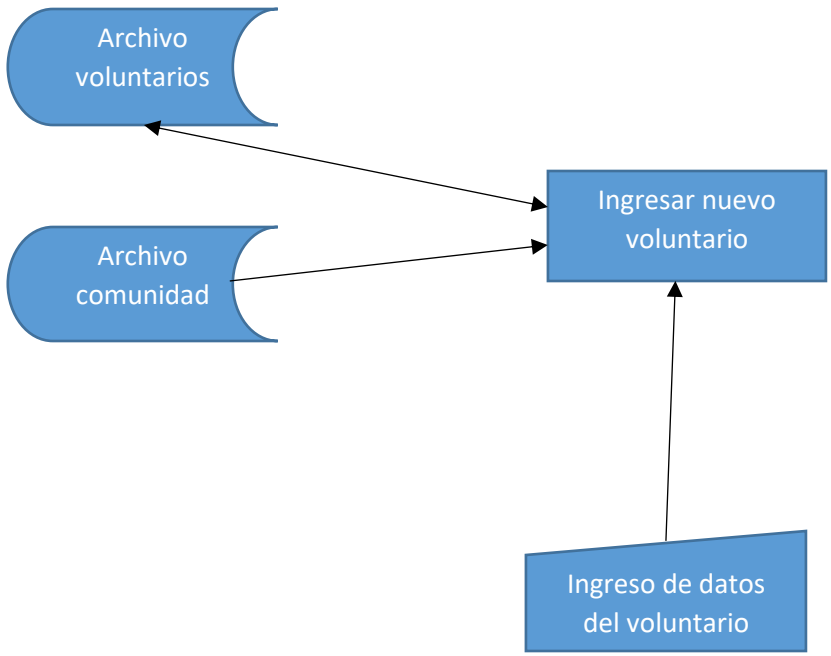


Figura 3.16: Diseño de pantalla correspondiente al ingreso de un nuevo niño (caso sin éxito).

3.4.9 PROGRAMA: NUEVO VOLUNTARIO

Registrar un nuevo voluntario	
Nombre del programa	AgregarVoluntario.vb
Objetivo	Otorga un formulario de registro para ingresar los datos de un nuevo voluntario.
Reglas del proceso	Inicialmente, el programa interactuará con la tabla Voluntario y Comunidad desempeñando la labor de verificación de posibles duplicaciones y cargando los códigos de las comunidades registradas en el sistema, respectivamente. Un registro exitoso es almacenado en la tabla Voluntario solo si todos los campos marcados como obligatorios son completados, el rut posee un formato real, la fecha de nacimiento no excede de la actual, la edad del voluntario se encuentra entre el rango de dieciséis y veintiséis años, su fecha de inscripción corresponde a la actual o a un tiempo anterior, la cantidad de dígitos del celular no es inferior a ocho y no se encuentren duplicaciones de la misma entidad (rut anteriormente almacenado en la tabla respectiva). En caso contrario, el usuario del programa será notificado a través de un cuadro de mensaje indicándole el error pertinente.
Documentación del código	Páginas 87 - 92

	Diagrama de bloque	

3.4.9.1 DISEÑO DE PANTALLAS: NUEVO VOLUNTARIO

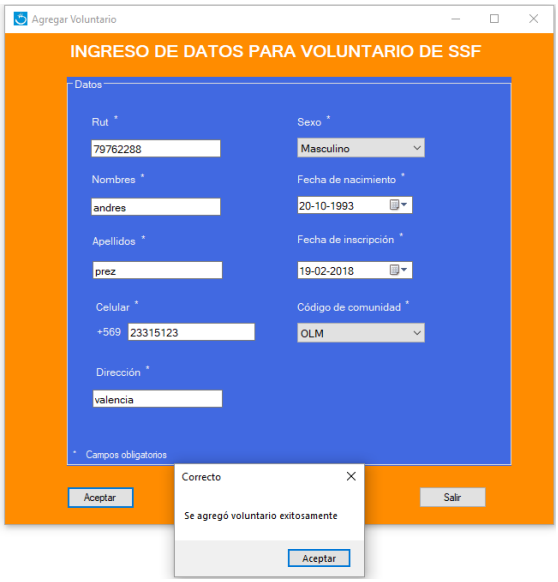
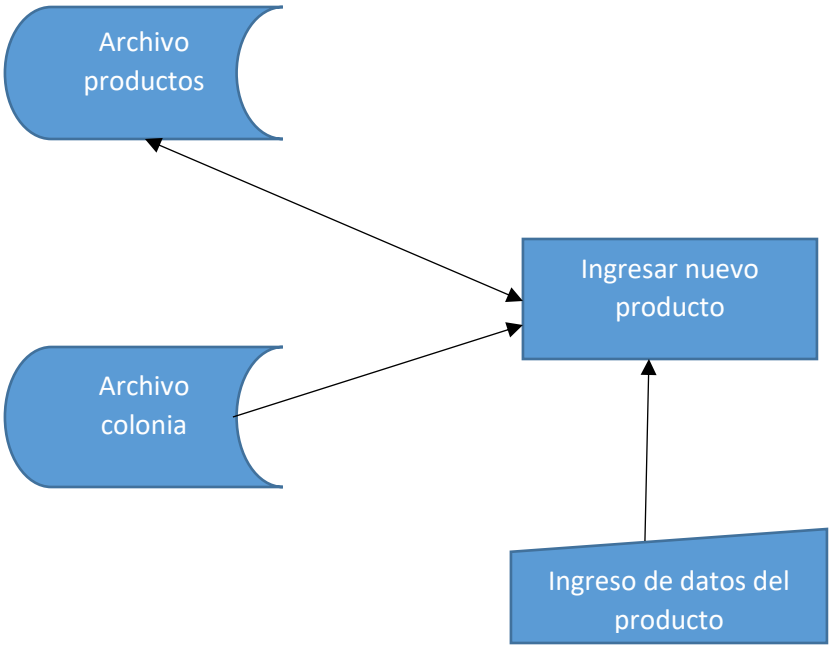


Figura 3.17: Diseño de pantalla correspondiente al ingreso de un nuevo voluntario (caso exitoso).

Figura 3.18: Diseño de pantalla correspondiente al ingreso de un nuevo voluntario (caso sin éxito).

3.4.10 PROGRAMA: NUEVO PRODUCTO

Registrar un nuevo producto	
Nombre del programa	AgregarProducto.vb
Objetivo	Otorga un formulario para ingresar los datos de un nuevo producto.
Reglas del proceso	Inicialmente, el programa interactuará con la tabla Colonia para realizar la consulta de los códigos de las entidades previamente registradas. Así también invocará la tabla Producto para validar que no se encuentren coincidencias después. Para que el registro sea exitoso y grabado en la tabla Producto se debe cumplir que todos los campos sean llenados y no repetirse el nombre del producto en una determinada colonia. En caso contrario, el usuario será notificado a través de un cuadro de mensaje con el error correspondiente.
Documentación del código	Páginas 93 - 94

	Diagrama de bloque	

3.4.10.1 DISEÑO DE PANTALLAS: NUEVO PRODUCTO

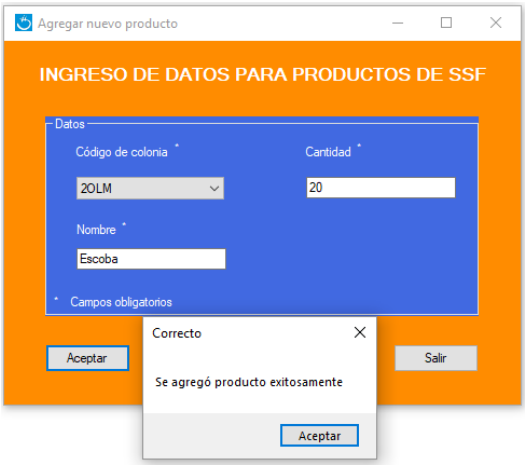


Figura 3.19: Diseño de pantalla correspondiente al registro de un nuevo producto (caso exitoso).

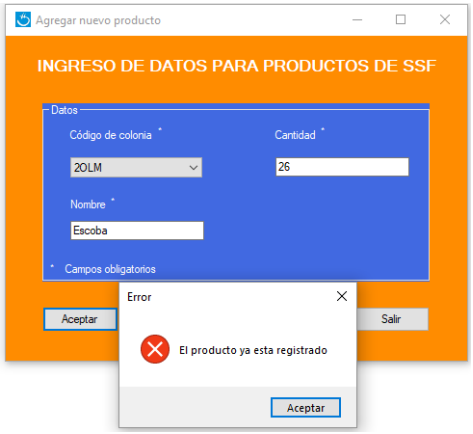
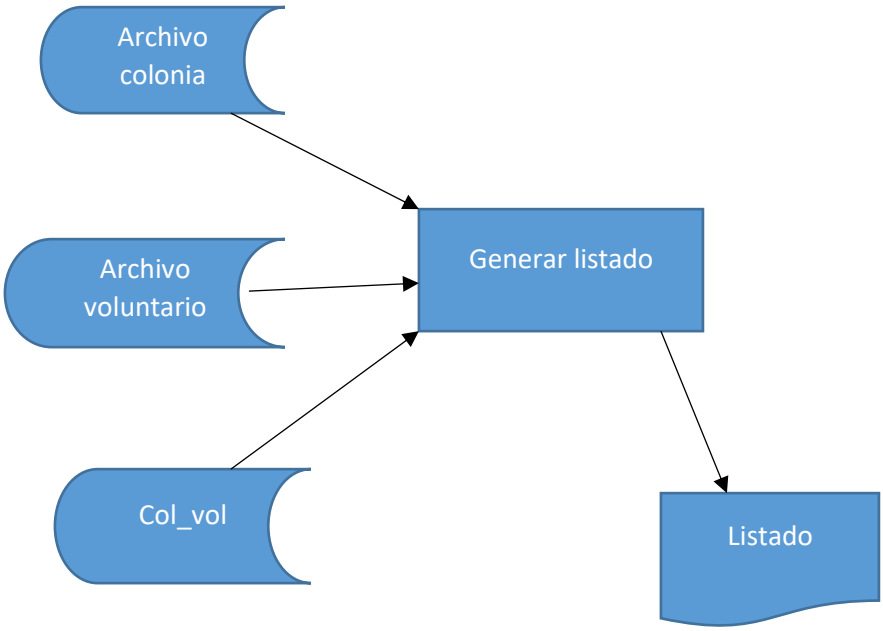


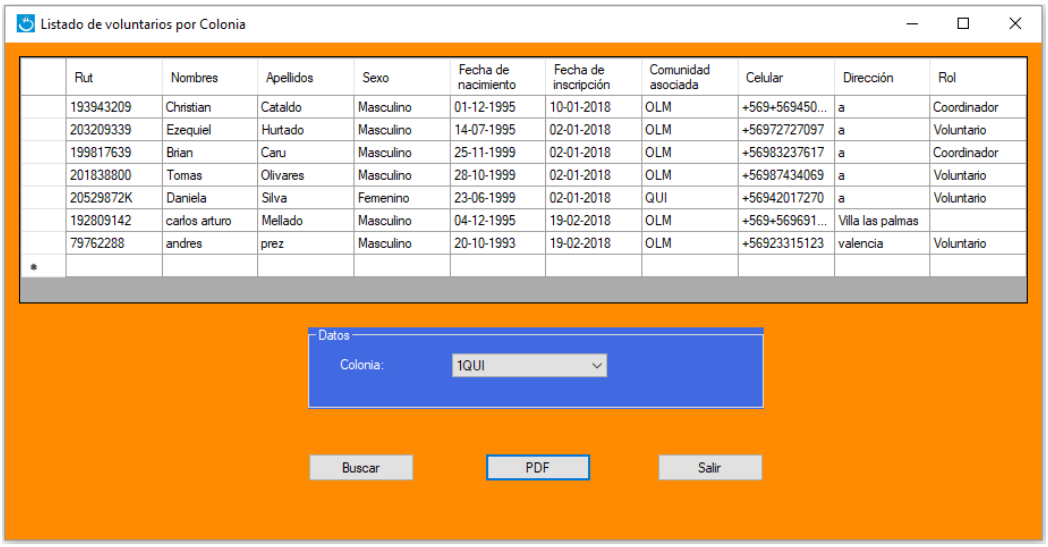
Figura 3.20: Diseño de pantalla correspondiente al registro de un nuevo producto (caso sin éxito).

3.4.11 PROGRAMA: LISTADO DE VOLUNTARIOS POR COLONIA

Visualizar los voluntarios de una colonia en específica	
Nombre del programa	ListadoVoluntariosColonia.vb
Objetivo	Visualizar los voluntarios inscritos en una determinada colonia.
Reglas del proceso	Inicialmente, el programa interactuará con la tabla Colonia para cargar los códigos de las entidades previamente registradas. Una vez realizada la consulta se accederá a la tabla Col_Vol para recuperar los rut de los voluntarios asociados al código de la colonia que fue seleccionado. Para finalizar, se invocará a la tabla Voluntario para visualizar los datos de todos ellos en caso de que el programa llegase a encontrar coincidencias y opcionalmente generar un documento imprimible. En el escenario contrario, el usuario será notificado a través de un cuadro de mensaje con el aviso correspondiente.
Documentación del código	Páginas 95 - 98

	Diagrama de bloque	

3.4.11.1 DISEÑO DE PANTALLAS: LISTAR VOLUNTARIOS POR COLONIA



Rut	Nombres	Apellidos	Sexo	Fecha de nacimiento	Fecha de inscripción	Comunidad asociada	Celular	Dirección	Rol
193943209	Christian	Cataldo	Masculino	01-12-1995	10-01-2018	OLM	+569+569450...	a	Coordinador
203209339	Ezequiel	Hurtado	Masculino	14-07-1995	02-01-2018	OLM	+56972727097	a	Voluntario
199817639	Brian	Caru	Masculino	25-11-1999	02-01-2018	OLM	+56983237617	a	Coordinador
201838800	Tomas	Olivares	Masculino	28-10-1999	02-01-2018	OLM	+56987434069	a	Voluntario
20529872K	Daniela	Silva	Femenino	23-06-1999	02-01-2018	QUI	+56942017270	a	Voluntario
192809142	carlos arturo	Mellado	Masculino	04-12-1995	19-02-2018	OLM	+569+569691...	Villa las palmas	
79762288	andres	prez	Masculino	20-10-1993	19-02-2018	OLM	+56923315123	valencia	Voluntario

Figura 3.21: Diseño de pantalla correspondiente al listado de los voluntarios para una determinada colonia (caso exitoso).

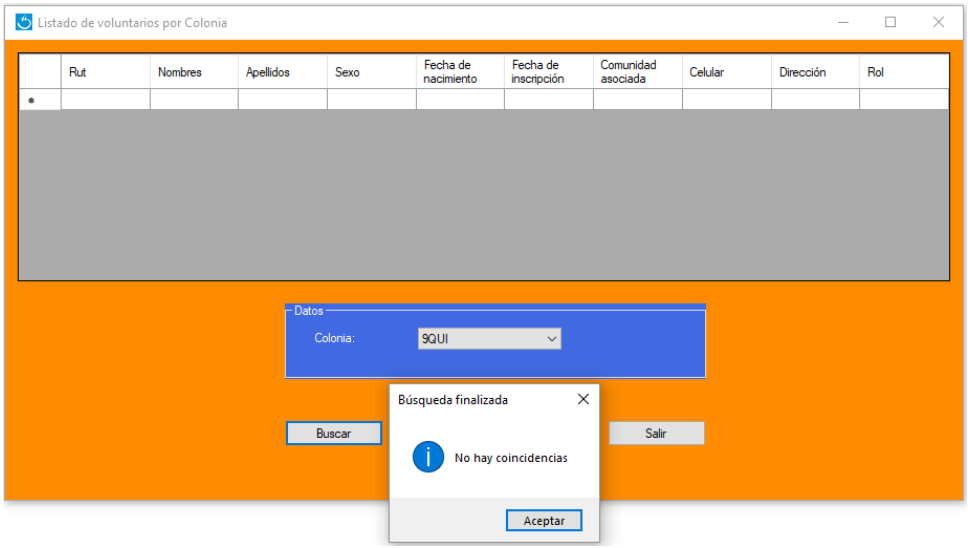


Figura 3.22: Diseño de pantalla correspondiente al listado de los voluntarios para una determinada colonia (caso sin éxito).

CONCLUSIÓN

Día a día la cantidad de información existente en el área computacional crece de manera exponencial. Uno de los principales problemas que conlleva esta tarea es el tratamiento de grandes cantidades de datos de manera segura, eficaz y oportuna. En la organización de carácter socioeducativo, Sonrisas Sin Fronteras, se evidencian innegablemente algunas falencias relacionadas a la automatización de las cosas: demora en el procesamiento de la información, emisión de informes con errores e ineficiencia en la toma de decisiones y difusión de la información. Es por ello que se decidió utilizar Visual Basic.NET y Microsoft Access los cuales permiten el cómputo y organización de datos de manera distribuida, sirviendo como una de las herramientas para resolver dichos problemas. La elaboración e implementación de un sistema informático mejorará la toma de decisiones y cada uno de los procesos que se desarrollan en cuanto a la creación, almacenamiento, actualización y búsqueda de la información relacionada con los voluntarios, niños, entidades pasivas como los apoderados de los niños, productos de cada Colonia y Comunidades; proporcionando eficientes métodos de registro, búsqueda y generación de reportes o documentos en el momento que sean solicitados.

Principalmente, lo que ha permitido llevar a buen término este proyecto de título, ha sido una combinación de varios factores, como los siguientes:

- La elección de metodologías adecuadas para estructurar el proceso de análisis, diseño e implementación que permitiese cumplir con los objetivos establecidos.
- Conocimiento de los requerimientos y del proceso urgente a automatizar, permitieron una mayor claridad a la hora de realizar la tarea de análisis.
- La elección de las herramientas adecuadas y poderosas para desarrollar el sistema control de inventario.

El sistema de inventario actualmente está en una etapa de marcha blanca, lo que se mantendrá, ya que han surgido nuevas comunidades interesadas en implementar el sistema, hasta lograr registrar la información que se necesite oportunamente. Cabe señalar que el

sistema está abierto a incorporar nuevos módulos o programas, debido a que los requerimientos planteados en su momento deben adaptarse a la dinámica que envuelve al proceso y a la organización, es por ello que se ha desarrollado un programa flexible de tal forma de dejarlo capacitado para aceptar estos planteamientos.

Sin duda ha sido una experiencia gratificante el hecho de implementar métodos de diseño de sistemas, ya que se amplía el campo de conocimientos y se aprende a estructurar modelos de datos que facilitarán en gran medida a optimizar los nuevos desafíos en materia de gestiones humanas y administrativas.

BIBLIOGRAFÍA Y LINKOGRAFÍA

[1] http://sonrisassinfronteras.cl

[2] https://sites.google.com/site/evolucionvisualbasic/classroom-news

[3] http://www.mvp-access.es/softjaen/manuales/sql/sjtsqlj001.htm

[4] https://msdn.microsoft.com/es-es/library/ms162140(v=sql.120).aspx

ANEXO: DOCUMENTACIÓN DE CÓDIGO

ANEXO 1: CÓDIGO FUENTE DEL PROGRAMA MENÚ PRINCIPAL

```
Imports WindowsApp3.FrmLogin
Imports WindowsApp3.FrmCrearUsuario
Imports WindowsApp3.FrmApoderados
Imports WindowsApp3.FrmColonia
Imports WindowsApp3.FrmNiños
Imports WindowsApp3.FrmComunidad
Imports WindowsApp3.MenuProducto
Public Class Menu
   Private Sub VoluntarioToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
VoluntarioToolStripMenuItem.Click
      Me.Hide()
      FrmVoluntario.ShowDialog()
   End Sub

   Private Sub CerrarSesionToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
CerrarSesionToolStripMenuItem.Click
      Dim Login As New FrmLogin
      Me.Hide()
      Login.Show()
   End Sub

   Private Sub Menu_Load(sender As Object, e As EventArgs) Handles MyBase.Load
      Me.CenterToScreen()
   End Sub

   Private Sub CrearNuevoUsuarioToolStripMenuItem_Click(sender As Object, e As EventArgs)
Handles CrearNuevoUsuarioToolStripMenuItem.Click
      Dim usuario As New FrmCrearUsuario
      Me.Hide()
      usuario.ShowDialog()
   End Sub

   Private Sub ApoderadosToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
ApoderadosToolStripMenuItem.Click
      Dim apoderado As New FrmApoderados
      Me.Hide()
      apoderado.ShowDialog()
   End Sub
```

```
    Private Sub ColoniasToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
ColoniasToolStripMenuItem.Click
        Dim colonia As New FrmColonia
        Me.Hide()
        colonia.ShowDialog()
    End Sub

    Private Sub NiñosToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
NiñosToolStripMenuItem.Click
        Dim niños As New FrmNiños
        Me.Hide()
        niños.ShowDialog()
    End Sub
    Private Sub ComunidadToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
ComunidadToolStripMenuItem.Click
        Dim comunidad As New FrmComunidad
        Me.Hide()
        comunidad.ShowDialog()
    End Sub

    Private Sub ProductosToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
ProductosToolStripMenuItem.Click
        Dim productos As New MenuProducto
        Me.Hide()
        productos.ShowDialog()
    End Sub

    Private Sub ListarNiñosToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
ListarNiñosToolStripMenuItem.Click
        Dim listadoNiños As New ListadoNiños
        Me.Hide()
        listadoNiños.ShowDialog()
    End Sub

    Private Sub ListarVoluntariosToolStripMenuItem_Click(sender As Object, e As EventArgs)
Handles ListarVoluntariosToolStripMenuItem.Click
        Dim listadoVoluntarios As New ListadoVoluntarios
        Me.Hide()
        listadoVoluntarios.ShowDialog()
    End Sub
```

```
    Private Sub ModificarUsuarioToolStripMenuItem_Click(sender As Object, e As EventArgs)
Handles ModificarUsuarioToolStripMenuItem.Click
        Dim modificarUsuario As New FrmModificarUsuario
        Me.Hide()
        modificarUsuario.ShowDialog()
    End Sub

    Private Sub ListarUsuariosToolStripMenuItem_Click(sender As Object, e As EventArgs)
Handles ListarUsuariosToolStripMenuItem.Click
        Dim listarUsuarios As New FrmListarUsuarios
        Me.Hide()
        listarUsuarios.ShowDialog()
    End Sub

End Class
```

ANEXO 2: CÓDIGO FUENTE DEL PROGRAMA LOGIN

```
Imports WindowsApp3.Conexion
Imports System.Data.OleDb
Imports WindowsApp3.Menu
Public Class FrmLogin
   Dim cnx As New Conexion
   Dim Encrip As New Encriptacion
   Public Sub New()
      InitializeComponent()
   End Sub
   Private Sub FrmLogin_Load(sender As Object, e As EventArgs) Handles MyBase.Load
      Me.CenterToScreen()
   End Sub

   Private Sub btn_ingresar_Click_1(sender As Object, e As EventArgs) Handles
btn_ingresar.Click
      Dim user As String
      Dim pass As String
      Dim Menu As New Menu
      user = txt_usuario.Text
      pass = txt_contraseña.Text
      txt_usuario.Enabled = False
      txt_contraseña.Enabled = False
      If (cnx.Autenticar(user, pass) = True) Then
         Me.Hide()
         ModUsuario.usuario = txt_usuario.Text
         ModUsuario.id = cnx.SeleccionarIDUsuario(txt_usuario.Text)
         ModUsuario.correoantiguo = cnx.SeleccionarCorreo(user)
         user = txt_usuario.Text.ToUpper
         MsgBox("BIENVENIDO/A " & user & "", MsgBoxStyle.Information, "Sesión Iniciada")
         ModUsuario.sw = 1
         Menu.ShowDialog()
      Else
         MsgBox("Usuario o contraseña incorrectos", MsgBoxStyle.Critical, "Credenciales
incorrectas")
         txt_usuario.Enabled = True
         txt_contraseña.Enabled = True
      End If
   End Sub
```

```
    Private Sub lbl_pass_LinkClicked(sender As Object, e As LinkLabelLinkClickedEventArgs)
Handles lbl_pass.LinkClicked
        Dim frmmodificar As New FrmRecuperarUsuario
        Me.Hide()
        frmmodificar.Show()
    End Sub

    Private Sub txt_usuario_GotFocus(sender As Object, e As EventArgs) Handles
txt_usuario.GotFocus
        If txt_usuario.Text = "Usuario" Then
            txt_usuario.Text = ""
            txt_usuario.ForeColor = Color.Black
        End If
    End Sub

    Private Sub txt_contraseña_GotFocus(sender As Object, e As EventArgs) Handles
txt_contraseña.GotFocus
        If txt_contraseña.Text = "Contraseña" Then
            txt_contraseña.Text = ""
            txt_contraseña.PasswordChar = "*"
            txt_contraseña.ForeColor = Color.Black
        End If
    End Sub

    Private Sub txt_usuario_LostFocus(sender As Object, e As EventArgs) Handles
txt_usuario.LostFocus
        If txt_usuario.Text = "" Then
            txt_usuario.Text = "Usuario"
            txt_usuario.ForeColor = SystemColors.ControlDark
        End If
    End Sub

    Private Sub txt_contraseña_LostFocus(sender As Object, e As EventArgs) Handles
txt_contraseña.LostFocus
        If txt_contraseña.Text = "" Then
            txt_contraseña.PasswordChar = ""
            txt_contraseña.Text = "Contraseña"
            txt_contraseña.ForeColor = SystemColors.ControlDark
        End If
    End Sub

    Private Sub FrmLogin_FormClosing(sender As Object, e As FormClosingEventArgs) Handles
Me.FormClosing
```

```
        Dim frmLogin As New FrmLogin
        Me.Hide()
        Me.Close()
    End Sub

    Private Sub lbl_user_LinkClicked(sender As Object, e As LinkLabelLinkClickedEventArgs)
Handles lbl_user.LinkClicked
        ModUsuario.sw = 0
        Dim frmCrear As New FrmCrearUsuario
        Me.Hide()
        frmCrear.Show()
    End Sub


End Class
```

ANEXO 3: CÓDIGO FUENTE DEL PROGRAMA NUEVO USUARIO

```
Imports WindowsApp3.Menu
Imports System.Net.Mail

Public Class FrmCrearUsuario
   Dim cnx As New Conexion
   Dim Encrip As New Encriptacion
   Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
      If ModUsuario.sw = 1 Then
         Dim menu As New Menu
         Me.Dispose()
         menu.Show()
      Else
         Dim frmLogin As New FrmLogin
         Me.Dispose()
         frmLogin.Show()
      End If
   End Sub

   Private Sub Usuario_Load(sender As Object, e As EventArgs) Handles MyBase.Load
      Me.CenterToScreen()
   End Sub

   Private Sub btn_crear_Click(sender As Object, e As EventArgs) Handles btn_crear.Click
      Dim user As String
      Dim passdos As String
      Dim correodos As String
      Dim fechaingreso As Date

      fechaingreso = Today
      user = txt_user.Text
      correodos = txt_correodos.Text

      If txt_user.Text = "" Or txt_passuno.Text = "" Or txt_passdos.Text = "" Or
txt_correouno.Text = "" Or txt_correodos.Text = "" Then
         MsgBox("Debe llenar campos vacíos", MsgBoxStyle.Critical, "Error al crear usuario")
      ElseIf cnx.ValidarUsuario(user) = True Then
         MsgBox("Usuario existente", MsgBoxStyle.Critical, "Error al crear usuario")
      ElseIf (txt_passuno.Text <> txt_passdos.Text) Then
         MsgBox("Contraseñas no coinciden", MsgBoxStyle.Critical, "Error al crear usuario")
      ElseIf (Encrip.IsValidEmail(correodos) = False) Then
```

```
        MsgBox("El formato de correo electrónico es incorrecto", MsgBoxStyle.Critical, "Error al
crear usuario")
    ElseIf (txt_correouno.Text <> txt_correodos.Text) Then
        MsgBox("Correos electrónicos no coinciden", MsgBoxStyle.Critical, "Error al crear
usuario")
    ElseIf cnx.ValidarCorreo(correodos) = True Then
        MsgBox("Correo existente", MsgBoxStyle.Critical, "Error al crear usuario")
    Else
        passdos = Encrip.Encriptar(txt_passdos.Text)

        cnx.IngresarUsuario(user, passdos, correodos, fechaingreso)

        Dim _Message As New System.Net.Mail.MailMessage()
        Dim _SMTP As New System.Net.Mail.SmtpClient

        'CONFIGURACIÓN DEL STMP
        _SMTP.Credentials = New System.Net.NetworkCredential("hangar1929@gmail.com",
"miguel8k1919!")
        _SMTP.Host = "smtp.gmail.com"
        _SMTP.Port = 587
        _SMTP.EnableSsl = True

        ' CONFIGURACION DEL MENSAJE
        _Message.[To].Add(txt_correodos.Text) 'Cuenta de Correo al que se le quiere enviar el e-
mail
        _Message.From = New System.Net.Mail.MailAddress("hangar1929@gmail.com",
"Sonrisas Sin Fronteras", System.Text.Encoding.UTF8) 'Quien lo envía
        _Message.Subject = "Contraseña para autenticación: " 'Sujeto del e-mail
        _Message.SubjectEncoding = System.Text.Encoding.UTF8 'Codificacion
        _Message.Body = "La contraseña de registro es: " + txt_passuno.Text.ToString 'contenido
del mail
        _Message.BodyEncoding = System.Text.Encoding.UTF8
        _Message.Priority = System.Net.Mail.MailPriority.Normal
        _Message.IsBodyHtml = False

        'ENVIO
        Try
            _SMTP.Send(_Message)
            'MessageBox.Show("Mensaje enviado correctamene", "Exito!",
MessageBoxButtons.OK)
        Catch ex As System.Net.Mail.SmtpException
            MessageBox.Show(ex.ToString, "Error!", MessageBoxButtons.OK)
        End Try
```

```
        txt_user.Text = ""
        txt_passuno.Text = ""
        txt_passdos.Text = ""
        txt_correouno.Text = ""
        txt_correodos.Text = ""
    End If
  End Sub

  Private Sub btn_cancelar_Click(sender As Object, e As EventArgs) Handles btn_cancelar.Click
    txt_user.Text = ""
    txt_passuno.Text = ""
    txt_passdos.Text = ""
    txt_correouno.Text = ""
    txt_correodos.Text = ""
  End Sub

End Class
```

ANEXO 4: CÓDIGO FUENTE DEL PROGRAMA NUEVA COMUNIDAD

```
Imports WindowsApp3.FrmComunidad
Public Class AgregarComunidad
   Dim comunidad As New Comunidad
   Dim bnsComunidad As New BnsComunidad
   Dim cnx As New Conexion

   Private Sub AgregarComunidad_Load(sender As Object, e As EventArgs) Handles MyBase.Load
      Me.CenterToScreen()
      Dim tabla As DataTable
      tabla = bnsComunidad.GetComunidad()
      cmb_comunidad.DropDownStyle = ComboBoxStyle.DropDownList
   End Sub

   Private Sub btn_aceptar_Click(sender As Object, e As EventArgs) Handles btn_aceptar.Click
      Dim id As String
      Dim tipotabla As String
      id = cmb_comunidad.SelectedItem
      tipotabla = "Comunidad"
      comunidad.Nombre = txt_nombre.Text
      comunidad.Codigo = cmb_comunidad.SelectedItem
      comunidad.Direccion = txt_direccion.Text

      If (txt_nombre.Text = "") Or (cmb_comunidad.Text = "Seleccione:") Or (txt_direccion.Text =
"") Then
         MsgBox("Debe llenar campos vacíos", MsgBoxStyle.Critical, "Error")
      ElseIf (cnx.validar(id, tipotabla) = True) Then
         MsgBox("Comunidad ya existe", MsgBoxStyle.Critical, "Error")
         txt_nombre.Text = ""
         txt_direccion.Text = ""
         cmb_comunidad.Text = "Seleccione:"
      Else
         bnsComunidad.SetAgregar(comunidad)
         txt_nombre.Text = ""
         txt_direccion.Text = ""
         cmb_comunidad.Text = "Seleccione:"
         Me.AgregarComunidad_Load(Me, Nothing)

      End If
   End Sub
   Private Sub btn_cancelar_Click(sender As Object, e As EventArgs) Handles btn_cancelar.Click
```

```
        txt_nombre.Text = ""
        txt_direccion.Text = ""
        cmb_comunidad.Text = "Seleccione:"
    End Sub
    Private Sub btn_salir_Click(sender As Object, e As EventArgs) Handles btn_salir.Click
        Dim frmcomunidad As New FrmComunidad
        Me.Dispose()
        frmcomunidad.Show()
    End Sub

End Class
```

ANEXO 5: CÓDIGO FUENTE DEL PROGRAMA NUEVA COLONIA

```
Imports WindowsApp3.FrmColonia
Imports WindowsApp3.Colonia
Imports WindowsApp3.BnsColonia
Imports System.Data.OleDb
Imports WindowsApp3.Conexion
Public Class AgregarColonia
   Dim colonia As New Colonia
   Dim bnsColonia As New BnsColonia
   Dim bnsComunidad As New BnsComunidad
   Dim cnx As New Conexion

   Private Sub AgregarColonia_Load(sender As Object, e As EventArgs) Handles MyBase.Load
      Me.CenterToScreen()
      Dim tope_com As Integer
      Dim cadena_com As String
      cmb_comunidad.DropDownStyle = ComboBoxStyle.DropDownList
      tope_com = BnsComunidad.GetCodComunidad().Rows.Count
      For i As Integer = 0 To tope_com - 1
         cadena_com =
bnsComunidad.GetCodComunidad().Rows(i)("Codigo_comunidad").ToString
         cmb_comunidad.Items.Add(cadena_com)
      Next
   End Sub

   Private Sub btn_aceptar_Click(sender As Object, e As EventArgs) Handles btn_aceptar.Click
      Dim ldtFecha1 As Date
      Dim ldtFecha2 As Date
      Dim ldtMesAnterior As Date
      Dim liDias As Integer
      Dim liMeses As Integer
      Dim liAños As Integer

      ldtFecha1 = Me.dtp_fechter.Value
      ldtMesAnterior = DateAdd("M", -1, ldtFecha1)
      ldtFecha2 = Me.dtp_fechini.Value

      liAños = DatePart("yyyy", ldtFecha1) - DatePart("yyyy", ldtFecha2)
      liMeses = DatePart("m", ldtFecha1) - DatePart("m", ldtFecha2)
      liDias = DatePart("d", ldtFecha1) - DatePart("d", ldtFecha2)
      Do While liDias < 0
```

```
        If liMeses = 0 Then
            liDias = liDias + DateAndTime.Day(DateSerial(Year(ldtFecha1), Month(ldtFecha1) + 1,
0))
        Else
            liDias = liDias + DateAndTime.Day(DateSerial(Year(ldtMesAnterior),
Month(ldtMesAnterior) + 1, 0))
        End If
        liMeses = liMeses - 1
        ldtMesAnterior = DateAdd("M", -1, ldtMesAnterior)
    Loop
    If liMeses < 0 Then
        liMeses = liMeses + 12
        liAños = liAños - 1
    End If

    Dim difaños = liAños
    Dim difmeses = liMeses
    Dim difdias = liDias

    If (txt_nombre.Text = "") Or (cmb_comunidad.Text = "Seleccione:") Or (txt_direccion.Text =
"") Or (dtp_fechini.Text = Now) Or (dtp_fechter.Text = Now) Then
        MsgBox("Debe llenar campos vacíos", MsgBoxStyle.Critical, "Error de ingreso")
    ElseIf difaños <> 0 Then
        MsgBox("La diferencia de años excede lo permitido", MsgBoxStyle.Critical, "Error de
ingreso")
    ElseIf difmeses <> 0 And difmeses <> 1 Then
        MsgBox("La diferencia de meses excede lo permitido", MsgBoxStyle.Critical, "Error de
ingreso")
    ElseIf difdias > 7 Then
        MsgBox("La diferencia de días excede lo permitido", MsgBoxStyle.Critical, "Error de
ingreso")
    Else
        colonia.Nombre = txt_nombre.Text
        colonia.CodComunidad = cmb_comunidad.Text
        colonia.Fecha_Inicio = dtp_fechini.Text
        colonia.Fecha_Termino = dtp_fechter.Text
        colonia.Direccion = txt_direccion.Text
        colonia.CodColonia = bnsColonia.GetCodigo().ToString + cmb_comunidad.SelectedItem
        bnsColonia.SetAgregar(colonia)
        txt_nombre.Text = ""
        dtp_fechini.Text = Now
        dtp_fechter.Text = Now
        txt_direccion.Text = ""
```

```
        cmb_comunidad.Text = "Seleccione:"
        Me.AgregarColonia_Load(Me, Nothing)

    End If
  End Sub


  Private Sub btn_cancelar_Click(sender As Object, e As EventArgs) Handles btn_cancelar.Click
    txt_nombre.Text = ""
    dtp_fechini.Text = Now
    dtp_fechter.Text = Now
    txt_direccion.Text = ""
    cmb_comunidad.Text = "Seleccione:"
  End Sub
  Private Sub btn_salir_Click(sender As Object, e As EventArgs) Handles btn_salir.Click
    Dim frmcolonia As New FrmColonia
    Me.Dispose()
    frmcolonia.Show()
  End Sub

End Class
```

ANEXO 6: PROGRAMA FUENTE DEL PROGRAMA AGREGAR VOLUNTARIO A UNA COLONIA

```
Imports WindowsApp3.Voluntario
Imports WindowsApp3.BnsVoluntario
Imports WindowsApp3.BnsColVol
Imports System.Data.OleDb
Imports WindowsApp3.Conexion
Public Class AgregarVoluntarioAColonia
  Dim voluntario As New Voluntario
  Dim bnsColonia As New BnsColonia
  Dim bnsVoluntario As New BnsVoluntario
  Dim bnsColvol As New BnsColVol
  Dim colvol As New Col_Vol
  Dim cnx As New Conexion

  Private Sub AgregaVoluntario_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    Dim tope As Integer
    Dim cadena As String
    Me.CenterToScreen()
    txt_rut.Enabled = False
    dgv_voluntario.DataSource = bnsVoluntario.GetVoluntario()
    tope = bnsColonia.GetCodColonia().Rows.Count
    dgv_voluntario.Columns(0).Visible = False
    cmb_codigo.DropDownStyle = ComboBoxStyle.DropDownList

    cmb_rol.DropDownStyle = ComboBoxStyle.DropDownList
    For i As Integer = 0 To tope - 1
      cadena = bnsColonia.GetCodColonia().Rows(i)("Codigo_Colonia").ToString
      cmb_codigo.Items.Add(cadena)
    Next

    dgv_voluntario.Columns(1).HeaderText = "Rut"
    dgv_voluntario.Columns(2).HeaderText = "Nombres"
    dgv_voluntario.Columns(3).HeaderText = "Apellidos"
    dgv_voluntario.Columns(4).HeaderText = "Sexo"
    dgv_voluntario.Columns(5).HeaderText = "Fecha de nacimiento"
    dgv_voluntario.Columns(6).HeaderText = "Fecha de inscripción"
    dgv_voluntario.Columns(7).HeaderText = "Comunidad asociada"
    dgv_voluntario.Columns(8).HeaderText = "Celular"
    dgv_voluntario.Columns(9).HeaderText = "Dirección"

  End Sub
```

```
    Private Sub dgv_voluntario_CellContentClick(sender As Object, e As
DataGridViewCellEventArgs) Handles dgv_voluntario.CellContentClick
        Dim index As Integer
        index = dgv_voluntario.CurrentRow.Index
        txt_rut.Text = dgv_voluntario.Item(1, index).Value()
    End Sub

    Private Sub btn_aceptar_Click(sender As Object, e As EventArgs) Handles btn_aceptar.Click
        Dim rut As String
        Dim codigo As String
        Dim tipotabla As String
        tipotabla = "Col_Vol"
        rut = txt_rut.Text
        codigo = cmb_codigo.SelectedItem
        If (cnx.validarCol(rut, codigo, tipotabla) = True) Then
            MsgBox("El voluntario ya se encuentra registrado en esta colonia", MsgBoxStyle.Critical,
"Error de ingreso")
        Else
            If (rut = "") Or (cmb_codigo.Text = "") Or (cmb_rol.Text = "") Then
                MsgBox("Faltan campos por completar", MsgBoxStyle.Critical, "Error de ingreso")
            Else
                colvol.Codigo = cmb_codigo.SelectedItem
                colvol.Rut = txt_rut.Text
                colvol.Rol = cmb_rol.SelectedItem
                bnsColvol.SetAgregar(colvol)
                cmb_codigo.Text = ""
                txt_rut.Text = ""
                cmb_rol.Text = ""
            End If
        End If
    End Sub

    Private Sub dgv_voluntario_EditingControlShowing(
      ByVal sender As Object,
      ByVal e As DataGridViewEditingControlShowingEventArgs) _
      Handles dgv_voluntario.EditingControlShowing
        ' Referenciamos el control TextBox subyacente en la celda actual.
        '
        Dim cellTextBox As TextBox =
            DirectCast(e.Control, DataGridViewTextBoxEditingControl)
        cellTextBox.ReadOnly = True
    End Sub
```

```
    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        Dim Colonia As New FrmColonia
        Me.Dispose()
        Colonia.Show()
    End Sub

End Class
```

ANEXO 7: CÓDIGO FUENTE DEL PROGRAMA NUEVO APODERADO

```
Imports WindowsApp3.FrmApoderados
Imports WindowsApp3.Apoderado
Imports WindowsApp3.BnsApoderado
Imports System.Data.OleDb
Imports WindowsApp3.Conexion
Public Class AgregarApoderado
   Dim apoderado As New Apoderado
   Dim bnsApoderado As New BnsApoderado
   Dim id As String
   Dim cnx As New Conexion

   Private Sub AgregarApoderado_Load(sender As Object, e As EventArgs) Handles MyBase.Load
      Me.CenterToScreen()
      Dim tabla As DataTable
      tabla = bnsApoderado.GetApoderado()
      cmb_sexo.DropDownStyle = ComboBoxStyle.DropDownList

   End Sub

   Private Sub btn_aceptar_Click(sender As Object, e As EventArgs) Handles btn_aceptar.Click
      Dim anoactual As Int32 = Year(Now)
      Dim anonac As Int32 = dtp_fechnac.Value.Year
      Dim anosdiferencia As Int32 = anoactual - anonac
      Dim tipotabla As String
      Dim dia As String
      tipotabla = "Apoderado"
      id = txt_rut.Text


      If (txt_rut.Text = "") Or (txt_nombre.Text = "") Or (txt_apellido.Text = "") Or (cmb_sexo.Text
= "Seleccione:") Or (txt_direccion.Text = "") Or (txt_celular.Text = "") Or (dtp_fechnac.Text =
Now) Then
         MsgBox("Debe llenar campos vacíos", MsgBoxStyle.Critical, "Error de ingreso")
      ElseIf (cnx.Modulo11(id) = False) Then
         MsgBox("Ingrese un rut válido", MsgBoxStyle.Critical, "Error en ingreso")
      ElseIf (txt_celular.Text.Length < 8) Then
         MsgBox("Ingrese un celular válido", MsgBoxStyle.Critical, "Error en ingreso")
      ElseIf (anosdiferencia > 100) Then
         MsgBox("Apoderado es de tercera edad", MsgBoxStyle.Critical, "Error en ingreso")
      ElseIf (anosdiferencia < 18) Then
```

```
            MsgBox("Apoderado es menor de edad", MsgBoxStyle.Critical, "Error en ingreso")
        ElseIf cnx.validar(id, tipotabla) = False Then
            Dim rut As String
            rut = txt_rut.Text
            rut = rut.ToUpper()
            rut = rut.Replace(".", "")
            rut = rut.Replace("-", "")
            txt_rut.Text = rut

            apoderado.Rut = txt_rut.Text
            apoderado.Nombre = txt_nombre.Text
            apoderado.Apellido = txt_apellido.Text
            apoderado.Sexo = cmb_sexo.SelectedItem
            apoderado.FechNac = dtp_fechnac.Text
            apoderado.Celular = "+569" + txt_celular.Text
            apoderado.Direccion = txt_direccion.Text
            bnsApoderado.SetAgregar(apoderado)
            txt_rut.Text = ""
            txt_nombre.Text = ""
            txt_apellido.Text = ""
            cmb_sexo.Text = "Seleccione:"
            dtp_fechnac.Text = Now
            txt_celular.Text = ""
            txt_direccion.Text = ""
            Me.AgregarApoderado_Load(Me, Nothing)
        Else
            MsgBox("El apoderado ya está registrado", MsgBoxStyle.Critical, "Error")
        End If
    End Sub

    Private Sub btn_cancelar_Click(sender As Object, e As EventArgs) Handles btn_cancelar.Click
        txt_rut.Text = ""
        txt_nombre.Text = ""
        txt_apellido.Text = ""
        cmb_sexo.Text = "Seleccione:"
        dtp_fechnac.Text = Now
        txt_celular.Text = ""
        txt_direccion.Text = ""
    End Sub
    Private Sub txt_nombre_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles txt_nombre.KeyPress
        If IsNumeric(e.KeyChar) AndAlso e.KeyChar <> ControlChars.Back Then
            e.Handled = True
```

```
        End If
    End Sub

    Private Sub txt_apellido_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles txt_apellido.KeyPress
        If IsNumeric(e.KeyChar) AndAlso e.KeyChar <> ControlChars.Back Then
            e.Handled = True
        End If
    End Sub

    Private Sub txt_celular_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles txt_celular.KeyPress
        If Not IsNumeric(e.KeyChar) AndAlso e.KeyChar <> ControlChars.Back Then
            e.Handled = True
        End If
    End Sub

    Private Sub btn_exit_Click(sender As Object, e As EventArgs) Handles btn_exit.Click
        Dim Frmapoderado As New FrmApoderados
        Me.Dispose()
        Frmapoderado.Show()
    End Sub

End Class
```

ANEXO 8: CÓDIGO FUENTE DEL PROGRAMA NUEVO NIÑO

```
Imports WindowsApp3.FrmNiños

Public Class AgregarNiño
   Dim niño As New Niño
   Dim bnsNiño As New BnsNiño
   Dim id As String
   Dim cnx As New Conexion
   Private Sub AgregarNiño_Load(sender As Object, e As EventArgs) Handles MyBase.Load
      Me.CenterToScreen()
      cmb_sexo.DropDownStyle = ComboBoxStyle.DropDownList
   End Sub

   Private Sub btn_aceptar_Click(sender As Object, e As EventArgs) Handles btn_aceptar.Click


      'Dim anoactual As Integer = Year(Now)
      Dim fechaactual As String = Today
      'Dim diadiferencianac = Now.Day - dtp_fechnac.Value.Day
      'Dim anonac As Integer = dtp_fechnac.Value.Year
      'Dim anosdiferencianac As Integer = anoactual - anonac

      Dim ldtFecha1 As Date
      Dim ldtFecha2 As Date
      Dim ldtMesAnterior As Date
      Dim liDias As Integer
      Dim liMeses As Integer
      Dim liAños As Integer

      ldtFecha1 = Date.Now
      ldtMesAnterior = DateAdd("M", -1, ldtFecha1)
      ldtFecha2 = Me.dtp_fechnac.Value

      liAños = DatePart("yyyy", ldtFecha1) - DatePart("yyyy", ldtFecha2)
      liMeses = DatePart("m", ldtFecha1) - DatePart("m", ldtFecha2)
      liDias = DatePart("d", ldtFecha1) - DatePart("d", ldtFecha2)
      Do While liDias < 0
         If liMeses = 0 Then
            liDias = liDias + DateAndTime.Day(DateSerial(Year(ldtFecha1), Month(ldtFecha1) + 1,
0))
         Else
```

```
            liDias = liDias + DateAndTime.Day(DateSerial(Year(ldtMesAnterior),
Month(ldtMesAnterior) + 1, 0))
        End If
        liMeses = liMeses - 1
        ldtMesAnterior = DateAdd("M", -1, ldtMesAnterior)
    Loop
    If liMeses < 0 Then
        liMeses = liMeses + 12
        liAños = liAños - 1
    End If

    Dim difaños = liAños
    Dim difmeses = liMeses
    Dim difdias = liDias

    '-----------------------------------------------------------------------'

    Dim ldtFecha3 As Date
    Dim ldtFecha4 As Date
    Dim ldtMesAnterior2 As Date
    Dim liDias2 As Integer
    Dim liMeses2 As Integer
    Dim liAños2 As Integer

    ldtFecha3 = Me.dtp_fechnac.Value
    ldtMesAnterior2 = DateAdd("M", -1, ldtFecha3)
    ldtFecha4 = Date.Now

    liAños2 = DatePart("yyyy", ldtFecha3) - DatePart("yyyy", ldtFecha4)
    liMeses2 = DatePart("m", ldtFecha3) - DatePart("m", ldtFecha4)
    liDias2 = DatePart("d", ldtFecha3) - DatePart("d", ldtFecha4)
    Do While liDias2 < 0
        If liMeses2 = 0 Then
            liDias2 = liDias2 + DateAndTime.Day(DateSerial(Year(ldtFecha3), Month(ldtFecha3) + 1,
0))
        Else
            liDias2 = liDias2 + DateAndTime.Day(DateSerial(Year(ldtMesAnterior2),
Month(ldtMesAnterior2) + 1, 0))
        End If
        liMeses2 = liMeses2 - 1
        ldtMesAnterior2 = DateAdd("M", -1, ldtMesAnterior2)
    Loop
    If liMeses2 < 0 Then
```

```
        liMeses2 = liMeses2 + 12
        liAños2 = (liAños2 - 1) * -1
    End If

    liAños2 = liAños2 * -1
    Dim difaños2 = liAños2
    Dim difmeses2 = liMeses2
    Dim difdias2 = liDias2


    Dim tipotabla As String
    Dim tipotabla2 As String
    Dim id As String
    Dim id2 As String
    tipotabla = "Niño"
    tipotabla2 = "Apoderado"

    id = txt_rut.Text
    id = id.ToUpper()
    id = id.Replace(".", "")
    id = id.Replace("-", "")
    txt_rut.Text = id.ToLower

    id2 = txt_ruttutor.Text
    id2 = id2.ToUpper()
    id2 = id2.Replace(".", "")
    id2 = id2.Replace("-", "")
    txt_ruttutor.Text = id2.ToLower


    If (txt_rut.Text = "") Or (txt_nombre.Text = "") Or (txt_apellido.Text = "") Or (cmb_sexo.Text
= "Seleccione:") Or (txt_direccion.Text = "") Or (txt_ruttutor.Text = "") Then
        MsgBox("Debe llenar campos vacíos", MsgBoxStyle.Critical, "Error en ingreso")
    ElseIf (cnx.Modulo11(id) = False) Then
        MsgBox("El rut del niño no es válido", MsgBoxStyle.Critical, "Error en ingreso")
    ElseIf (txt_celular.Text.length > 0 And txt_celular.Text.Length < 8) Then
        MsgBox("Ingrese un celular válido", MsgBoxStyle.Critical, "Error en ingreso")
    ElseIf (difaños < 4) Then
        If (difmeses >= 0) Then
            If (difdias >= 0) Then
                MsgBox("Niño es menor a la edad permitida de ingreso", MsgBoxStyle.Critical, "Error
en ingreso")
            End If
```

```
        End If
    ElseIf (difaños2 = 13) Then
        If (difmeses >= 0) Then
            If (difdias >= 0) Then
                MsgBox("Niño es mayor a la edad permitida de ingreso", MsgBoxStyle.Critical, "Error
en ingreso")
            End If
        End If
    ElseIf (difaños > 13) Then
        MsgBox("Niño es mayor a la edad permitida de ingreso", MsgBoxStyle.Critical, "Error en
ingreso")
    ElseIf (dtp_ingreso.Value.Date > fechaactual) Then
        MsgBox("Fecha de ingreso no es válida", MsgBoxStyle.Critical, "Error en ingreso")
    ElseIf cnx.Modulo11(id2) = False Then
        MsgBox("El rut del apoderado no es válido", MsgBoxStyle.Critical, "Error en ingreso")
    ElseIf cnx.validar(id2, tipotabla2) = False Then
        MsgBox("Apoderado no se encuentra registrado", MsgBoxStyle.Critical, "Error en
ingreso")
    ElseIf cnx.validar(id, tipotabla) = False Then
        Dim rut As String
        rut = txt_rut.Text
            rut = rut.ToUpper()
            rut = rut.Replace(".", "")
            rut = rut.Replace("-", "")
            txt_rut.Text = rut
            niño.Rut = txt_rut.Text
            niño.RutTutor = txt_ruttutor.Text
            niño.Nombre = txt_nombre.Text
            niño.Apellido = txt_apellido.Text
            niño.Sexo = cmb_sexo.SelectedItem
            niño.FechNac = dtp_fechnac.Text
        niño.FechInc = dtp_ingreso.Text
        If txt_celular.Text.Length > 0 Then
            niño.Celular = "+569" + txt_celular.Text
        Else
            niño.Celular = txt_celular.Text
        End If
        niño.Direccion = txt_direccion.Text

        niño.Observacion = txt_observacion.Text
        bnsNiño.SetAgregar(niño)
        txt_rut.Text = ""
        txt_nombre.Text = ""
```

```
        txt_apellido.Text = ""
        cmb_sexo.Text = "Seleccione:"
        dtp_fechnac.Text = Now
        txt_celular.Text = ""
        txt_direccion.Text = ""
        dtp_ingreso.Text = Now
        txt_ruttutor.Text = ""
        Me.AgregarNiño_Load(Me, Nothing)
        'Dim frmNiños As New FrmNiños
        'Me.Dispose()
        'frmNiños.Show()
    Else
        MsgBox("El niño ya está registrado", MsgBoxStyle.Critical, "Error en ingreso")
    End If
End Sub

Private Sub btn_cancelar_Click(sender As Object, e As EventArgs) Handles btn_cancelar.Click
    txt_rut.Text = ""
    txt_nombre.Text = ""
    txt_apellido.Text = ""
    cmb_sexo.Text = "Seleccione:"
    dtp_fechnac.Text = Now
    txt_celular.Text = ""
    txt_direccion.Text = ""
    dtp_ingreso.Text = Now
    txt_ruttutor.Text = ""
End Sub

Private Sub txt_nombre_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles txt_nombre.KeyPress
    If IsNumeric(e.KeyChar) AndAlso e.KeyChar <> ControlChars.Back Then
        e.Handled = True
    End If
End Sub

Private Sub txt_apellido_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles txt_apellido.KeyPress
    If IsNumeric(e.KeyChar) AndAlso e.KeyChar <> ControlChars.Back Then
        e.Handled = True
    End If
End Sub
```

```
    Private Sub txt_celular_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles txt_celular.KeyPress
        If Not IsNumeric(e.KeyChar) AndAlso e.KeyChar <> ControlChars.Back Then
            e.Handled = True
        End If
    End Sub

    Private Sub btn_salir_Click(sender As Object, e As EventArgs) Handles btn_salir.Click
        Dim frmNiños As New FrmNiños
        Me.Dispose()
        frmNiños.Show()
    End Sub

End Class
```

ANEXO 9: CÓDIGO FUENTE DEL PROGRAMA NUEVO VOLUNTARIO

```
Imports WindowsApp3.FrmVoluntario
Imports WindowsApp3.Voluntario
Imports WindowsApp3.BnsVoluntario
Imports System.Data.OleDb
Imports WindowsApp3.Conexion

Public Class AgregarVoluntario
   Dim voluntario As New Voluntario
   Dim bnsVoluntario As New BnsVoluntario
   Dim bnsComunidad As New BnsComunidad
   Dim bnsRol As New BnsRol
   Dim id As String
   Dim cnx As New Conexion

   Private Sub AgregarVoluntario_Load(sender As Object, e As EventArgs) Handles MyBase.Load
      Me.CenterToScreen()
      Dim tope_com As Integer
      Dim cadena_com As String

      tope_com = bnsComunidad.GetCodComunidad().Rows.Count
      For i As Integer = 0 To tope_com - 1
         cadena_com =
bnsComunidad.GetCodComunidad().Rows(i)("Codigo_comunidad").ToString
         cmb_comunidad.Items.Add(cadena_com)
      Next

      cmb_sexo.DropDownStyle = ComboBoxStyle.DropDownList
      cmb_comunidad.DropDownStyle = ComboBoxStyle.DropDownList

   End Sub

   Private Sub btn_aceptar_Click(sender As Object, e As EventArgs) Handles btn_aceptar.Click
      Dim anoactual As Int32 = Year(Now)
      Dim anonac As Int32 = dtp_fechnac.Value.Year
      Dim anosdiferencia As Int32 = anoactual - anonac
      Dim tipotabla As String
      Dim ldtFecha1 As Date
      Dim ldtFecha2 As Date
      Dim ldtMesAnterior As Date
      Dim liDias As Integer
```

```
    Dim liMeses As Integer
    Dim liAños As Integer
    Dim fechaactual As String = Today
    ldtFecha1 = Date.Now
    ldtMesAnterior = DateAdd("M", -1, ldtFecha1)
    ldtFecha2 = Me.dtp_fechnac.Value

    liAños = DatePart("yyyy", ldtFecha1) - DatePart("yyyy", ldtFecha2)
    liMeses = DatePart("m", ldtFecha1) - DatePart("m", ldtFecha2)
    liDias = DatePart("d", ldtFecha1) - DatePart("d", ldtFecha2)
    Do While liDias < 0
      If liMeses = 0 Then
        liDias = liDias + DateAndTime.Day(DateSerial(Year(ldtFecha1), Month(ldtFecha1) + 1,
0))

Else
        liDias = liDias + DateAndTime.Day(DateSerial(Year(ldtMesAnterior),
Month(ldtMesAnterior) + 1, 0))
      End If
      liMeses = liMeses - 1
      ldtMesAnterior = DateAdd("M", -1, ldtMesAnterior)
    Loop
    If liMeses < 0 Then
      liMeses = liMeses + 12
      liAños = liAños - 1
    End If

    Dim difaños = liAños
    Dim difmeses = liMeses
    Dim difdias = liDias

    '-----------------------------------------------------------------------'

    Dim ldtFecha3 As Date
    Dim ldtFecha4 As Date
    Dim ldtMesAnterior2 As Date
    Dim liDias2 As Integer
    Dim liMeses2 As Integer
    Dim liAños2 As Integer

    ldtFecha3 = Me.dtp_fechnac.Value
    ldtMesAnterior2 = DateAdd("M", -1, ldtFecha3)
    ldtFecha4 = Date.Now
```

```
    liAños2 = DatePart("yyyy", ldtFecha3) - DatePart("yyyy", ldtFecha4)
    liMeses2 = DatePart("m", ldtFecha3) - DatePart("m", ldtFecha4)
    liDias2 = DatePart("d", ldtFecha3) - DatePart("d", ldtFecha4)
    Do While liDias2 < 0
      If liMeses2 = 0 Then
        liDias2 = liDias2 + DateAndTime.Day(DateSerial(Year(ldtFecha3), Month(ldtFecha3) + 1,
0))
      Else
        liDias2 = liDias2 + DateAndTime.Day(DateSerial(Year(ldtMesAnterior2),
Month(ldtMesAnterior2) + 1, 0))
      End If
      liMeses2 = liMeses2 - 1
      ldtMesAnterior2 = DateAdd("M", -1, ldtMesAnterior2)
    Loop
    If liMeses2 < 0 Then
      liMeses2 = liMeses2 + 12
      liAños2 = (liAños2 - 1) * -1
    End If

    liAños2 = liAños2 * -1
    Dim difaños2 = liAños2
    Dim difmeses2 = liMeses2
    Dim difdias2 = liDias2
    tipotabla = "Voluntario"
    id = txt_rut.Text

    If (txt_rut.Text = "") Or (txt_nombre.Text = "") Or (txt_apellido.Text = "") Or (cmb_sexo.Text
= "Seleccione:") Or (txt_celular.Text = "") Or (txt_direccion.Text = "") Or
(cmb_comunidad.SelectedItem = "") Then
      MsgBox("Debe llenar campos vacíos", MsgBoxStyle.Critical, "Error en ingreso")
    ElseIf (cnx.Modulo11(id) = False) Then
      MsgBox("Rut del voluntario no es válido", MsgBoxStyle.Critical, "Error en ingreso")
    ElseIf (txt_celular.Text.Length > 0 And txt_celular.Text.Length < 8) Then
      MsgBox("Ingrese un celular válido", MsgBoxStyle.Critical, "Error en ingreso")
    ElseIf (anosdiferencia < 15) Then
      MsgBox("Voluntario es menor a la edad permitida de ingreso", MsgBoxStyle.Critical,
"Error en ingreso")
    ElseIf (difaños < 16) Then
      If (difmeses >= 0) Then
        If (difdias >= 0) Then
          MsgBox("Voluntario es menor a la edad permitida de ingreso", MsgBoxStyle.Critical,
"Error en ingreso")
```

```
            End If
        End If
    ElseIf (difaños2 = 26) Then
        If (difmeses >= 0) Then
            If (difdias >= 0) Then
                MsgBox("Voluntario es mayor a la edad permitida de ingreso", MsgBoxStyle.Critical,
"Error en ingreso")
            End If
        End If
    ElseIf (dtp_ingreso.Value.Date > fechaactual) Then
        MsgBox("Fecha de ingreso no es válida", MsgBoxStyle.Critical, "Error en ingreso")
    ElseIf cnx.validar(id, tipotabla) = False Then
        Dim rut As String
        rut = txt_rut.Text
        rut = rut.ToUpper()
        rut = rut.Replace(".", "")
        rut = rut.Replace("-", "")
        txt_rut.Text = rut
        voluntario.Rut = txt_rut.Text
        voluntario.Nombre = txt_nombre.Text
        voluntario.Apellido = txt_apellido.Text
        voluntario.Sexo = cmb_sexo.SelectedItem
        voluntario.FechaNac = dtp_fechnac.Text
        voluntario.Codigo_Comunidad = cmb_comunidad.SelectedItem
        voluntario.FechaIni = dtp_ingreso.Text
        voluntario.Telefono = "+569" + txt_celular.Text
        voluntario.Direccion = txt_direccion.Text
        bnsVoluntario.SetAgregar(voluntario)
        txt_rut.Text = ""
        txt_nombre.Text = ""
        txt_apellido.Text = ""
        cmb_sexo.Text = "Seleccione:"
        dtp_fechnac.Text = Now
        txt_celular.Text = ""
        txt_direccion.Text = ""
        dtp_ingreso.Text = Now
        'Me.AgregarVoluntario_Load(Me, Nothing)
    Else
        MsgBox("El voluntario ya está registrado", MsgBoxStyle.Critical, "Error en ingreso")
    End If
End Sub
```

```
Private Sub btn_cancelar_Click(sender As Object, e As EventArgs) Handles btn_cancelar.Click
        txt_rut.Text = ""
        txt_nombre.Text = ""
        txt_apellido.Text = ""
        cmb_sexo.Text = "Seleccione:"
        dtp_fechnac.Text = Now
        txt_celular.Text = ""
        txt_direccion.Text = ""
        cmb_comunidad.Text = "Seleccione:"
        dtp_ingreso.Text = Now
    End Sub

    Private Sub btn_salir_Click(sender As Object, e As EventArgs) Handles btn_salir.Click
        Dim frmmantenedor As New FrmVoluntario
        Me.Dispose()
        frmmantenedor.Show()
    End Sub
    Private Sub txt_nombre_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles txt_nombre.KeyPress
        If IsNumeric(e.KeyChar) AndAlso e.KeyChar <> ControlChars.Back Then
            e.Handled = True
        End If
    End Sub

    Private Sub txt_apellido_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles txt_apellido.KeyPress
        If IsNumeric(e.KeyChar) AndAlso e.KeyChar <> ControlChars.Back Then
            e.Handled = True
        End If
    End Sub

    Private Sub txt_celular_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles txt_celular.KeyPress
        If Not IsNumeric(e.KeyChar) AndAlso e.KeyChar <> ControlChars.Back Then
            e.Handled = True
        End If
    End Sub

    Private Sub dgv_voluntario_EditingControlShowing(
        ByVal sender As Object,
        ByVal e As DataGridViewEditingControlShowingEventArgs) _
```

```
        ' Referenciamos el control TextBox subyacente en la celda actual.
        '
        Dim cellTextBox As TextBox =
            DirectCast(e.Control, DataGridViewTextBoxEditingControl)

        cellTextBox.ReadOnly = True

    End Sub

End Class
```

ANEXO 10: CÓDIGO FUENTE DEL PROGRAMA NUEVO PRODUCTO

```
Public Class AgregarProducto
  Dim bnsProducto As New BnsProducto
  Dim bnsColonia As New BnsColonia
  Dim producto As New Producto
  Dim cnx As New Conexion
  Private Sub AgregarProducto_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    Me.CenterToScreen()
    Dim tope As Integer
    Dim cadena As String
    tope = bnsColonia.GetCodColonia().Rows.Count
    For i As Integer = 0 To tope - 1
      cadena = bnsColonia.GetCodColonia().Rows(i)("Codigo_Colonia").ToString
      cmb_codigo.Items.Add(cadena)
    Next
    cmb_codigo.DropDownStyle = ComboBoxStyle.DropDownList
  End Sub

  Private Sub btn_aceptar_Click(sender As Object, e As EventArgs) Handles btn_aceptar.Click
    Dim codigo As String
    Dim nombre As String
    Dim tipotabla As String


    codigo = cmb_codigo.SelectedItem
    nombre = txt_nombre.Text
    tipotabla = "Producto"

    If (txt_cantidad.Text = "") Or (txt_nombre.Text = "") Then
      MsgBox("Debe llenar campos vacíos", MsgBoxStyle.Critical, "Error")
    ElseIf cnx.validarProducto(codigo, nombre, tipotabla) = False Then
      producto.CodigoColonia = cmb_codigo.SelectedItem
      producto.Nombre = txt_nombre.Text
      producto.Cantidad = txt_cantidad.Text
      producto.CodProd = bnsProducto.GetCodProd().ToString + cmb_codigo.SelectedItem
      bnsProducto.SetAgregar(producto)

      cmb_codigo.Text = ""
      txt_cantidad.Text = ""
      txt_nombre.Text = ""
    Else
```

```
        MsgBox("El producto ya esta registrado", MsgBoxStyle.Critical, "Error")
        cmb_codigo.Text = ""
        txt_cantidad.Text = ""
        txt_nombre.Text = ""
    End If

  End Sub

  Private Sub btn_cancelar_Click_1(sender As Object, e As EventArgs) Handles
btn_cancelar.Click
    cmb_codigo.Text = ""
    txt_cantidad.Text = ""
    txt_nombre.Text = ""
  End Sub

  Private Sub btn_salir_Click(sender As Object, e As EventArgs) Handles btn_salir.Click
    Dim salir As New MenuProducto
    Me.Dispose()
    salir.Show()
  End Sub

End Class
```

ANEXO 11: CÓDIGO FUENTE DEL PROGRAMA LISTADO DE VOLUNTARIOS POR COLONIA

```
Imports iTextSharp.text.pdf
Imports iTextSharp.text
Imports System.IO
Public Class ListadoVoluntariosColonia
   Dim bnsColonia As New BnsColonia
   Dim bnsVoluntario As New BnsVoluntario
   Private Sub ListadoVoluntariosColonia_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
      Me.CenterToScreen()
      Dim tope As Integer
      Dim cadena As String
      tope = bnsColonia.GetCodColonia().Rows.Count
      For i As Integer = 0 To tope - 1
         cadena = bnsColonia.GetCodColonia().Rows(i)("Codigo_Colonia").ToString
         cmb_codigo.Items.Add(cadena)
      Next
      cmb_codigo.DropDownStyle = ComboBoxStyle.DropDownList
      btn_pdf.Visible = False
   End Sub
   Private Sub btn_buscar_Click(sender As Object, e As EventArgs) Handles btn_buscar.Click
      Dim codigo As String
      Dim cantvoluntarios As Integer

      codigo = cmb_codigo.SelectedItem
      cantvoluntarios = bnsVoluntario.GetCantVoluntarioColonia(codigo)

      If cantvoluntarios = 0 Then
         MsgBox("No hay coincidencias", MsgBoxStyle.Information, "Búsqueda finalizada")
         dgv_voluntario.DataSource = bnsVoluntario.GetVoluntarioColonia(codigo)
         dgv_voluntario.Columns(0).Visible = False
         dgv_voluntario.Columns(10).Visible = False
         dgv_voluntario.Columns(11).Visible = False
         dgv_voluntario.Columns(12).Visible = False
         dgv_voluntario.Columns(1).HeaderText = "Rut"
         dgv_voluntario.Columns(2).HeaderText = "Nombres"
         dgv_voluntario.Columns(3).HeaderText = "Apellidos"
         dgv_voluntario.Columns(4).HeaderText = "Sexo"
         dgv_voluntario.Columns(5).HeaderText = "Fecha de nacimiento"
         dgv_voluntario.Columns(6).HeaderText = "Fecha de inscripción"
         dgv_voluntario.Columns(7).HeaderText = "Comunidad asociada"
```

```
        dgv_voluntario.Columns(8).HeaderText = "Celular"
        dgv_voluntario.Columns(9).HeaderText = "Dirección"
        dgv_voluntario.Columns(13).HeaderText = "Rol"
        btn_pdf.Visible = False
    Else
        dgv_voluntario.DataSource = bnsVoluntario.GetVoluntarioColonia(codigo)
        dgv_voluntario.Columns(0).Visible = False
        dgv_voluntario.Columns(10).Visible = False
        dgv_voluntario.Columns(11).Visible = False
        dgv_voluntario.Columns(12).Visible = False
        dgv_voluntario.Columns(1).HeaderText = "Rut"
        dgv_voluntario.Columns(2).HeaderText = "Nombres"
        dgv_voluntario.Columns(3).HeaderText = "Apellidos"
        dgv_voluntario.Columns(4).HeaderText = "Sexo"
        dgv_voluntario.Columns(5).HeaderText = "Fecha de nacimiento"
        dgv_voluntario.Columns(6).HeaderText = "Fecha de inscripción"
        dgv_voluntario.Columns(7).HeaderText = "Comunidad asociada"
        dgv_voluntario.Columns(8).HeaderText = "Celular"
        dgv_voluntario.Columns(9).HeaderText = "Dirección"
        dgv_voluntario.Columns(13).HeaderText = "Rol"
        btn_pdf.Visible = True
    End If
End Sub
Private Sub dgv_voluntario_EditingControlShowing(
    ByVal sender As Object,
    ByVal e As DataGridViewEditingControlShowingEventArgs) _
    Handles dgv_voluntario.EditingControlShowing
    Dim cellTextBox As TextBox =
        DirectCast(e.Control, DataGridViewTextBoxEditingControl)
    cellTextBox.ReadOnly = True
End Sub
Public Function GetColumnasSize(ByVal dg As DataGridView) As Single()
    'funcion para obtener el tamaño de la columnas del datagridview
    Dim values As Single() = New Single(dg.ColumnCount - 1) {}
    For i As Integer = 0 To dg.ColumnCount - 1
        values(i) = CSng(dg.Columns(i).Width)
    Next
    Return values
End Function
Public Sub ExportarDatosPDF(ByVal document As Document)
    'Se crea un objeto PDFTable con el numero de columnas del DataGridView.
    Dim datatable As New PdfPTable(dgv_voluntario.ColumnCount)
    'Se asignan algunas propiedades para el diseño del PDF.
```

```
        datatable.DefaultCell.Padding = 3
        Dim headerwidths As Single() = GetColumnasSize(dgv_voluntario)
        datatable.SetWidths(headerwidths)
        datatable.WidthPercentage = 100
        datatable.DefaultCell.BorderWidth = 2
        datatable.DefaultCell.HorizontalAlignment = Element.ALIGN_CENTER
        'Se crea el encabezado en el PDF.
        Dim encabezado As New Paragraph("Reporte de voluntarios registrados en la colonia", New
Font(Font.Name = "Tahoma", 20, Font.Bold))

        'Se crea el texto abajo del encabezado.
        Dim texto As New Phrase("Reporte de voluntarios en la colonia realizados en la fecha :" +
Now.Date(), New Font(Font.Name = "Tahoma", 14, Font.Bold))
        'Se capturan los nombres de las columnas del DataGridView.
        For i As Integer = 0 To dgv_voluntario.ColumnCount - 1

            datatable.AddCell(dgv_voluntario.Columns(i).HeaderText)
        Next

        datatable.HeaderRows = 1
        datatable.DefaultCell.BorderWidth = 1
        'Se generan las columnas del DataGridView.

        For i As Integer = 0 To dgv_voluntario.Rows.Count - 2
            For j As Integer = 0 To dgv_voluntario.Columns.Count - 1
                datatable.AddCell(dgv_voluntario(j, i).Value.ToString())
            Next
            datatable.CompleteRow()
        Next
        'Se agrega el PDFTable al documento.
        document.Add(encabezado)
        document.Add(texto)
        document.Add(datatable)
    End Sub

    Private Sub btn_pdf_Click(sender As Object, e As EventArgs) Handles btn_pdf.Click

        Try
            'Intentar generar el documento.
            Dim doc As New Document(PageSize.A4.Rotate(), 10, 10, 10, 10)
            'Path que guarda el reporte en el escritorio de windows (Desktop).
```

```
        Dim filename As String =
Environment.GetFolderPath(Environment.SpecialFolder.DesktopDirectory) + "\Lista de
voluntarios por colonia.pdf"
        Dim file As New FileStream(filename, FileMode.Create, FileAccess.Write,
FileShare.ReadWrite)
        PdfWriter.GetInstance(doc, file)
        doc.Open()
        ExportarDatosPDF(doc)
        doc.Close()
        Process.Start(filename)
      Catch ex As Exception
        'Si el intento es fallido, mostrar MsgBox.
        MessageBox.Show("No se puede generar el documento PDF.", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error)
      End Try
    End Sub

    Private Sub btn_salir_Click(sender As Object, e As EventArgs) Handles btn_salir.Click
      Dim salir As New ListadoVoluntarios
      Me.Dispose()
      salir.Show()
    End Sub

End Class
```