

**Instituto de Educación Superior Tecnológico Público**  
**“De Las Fuerzas Armadas”**



**TRABAJO DE APLICACIÓN PROFESIONAL**

**IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL DE  
ASISTENCIA DE PERSONAL Y VIGILANCIA CANINA PARA LA  
EMPRESA SECURITY DOG ORMEÑO**

**PARA OPTAR EL TÍTULO PROFESIONAL TÉCNICO EN  
COMPUTACIÓN E INFORMÁTICA**

**PRESENTADO POR:**

**DÍAZ ESPINOZA, Orlando.**

**QUISPE ILLACANCHI, Ruddy.**

**LIMA, PERÚ**

**2021**



A nuestros padres, porque habernos apoyado  
Constantemente con valores en esta etapa de  
nuestras vidas.

## **Agradecimientos**

A Dios porque sin él nada de esto hubiera sido posible, guiándonos siempre por el camino del éxito.

A nuestros padres quienes a lo largo de toda nuestra vida han apoyado y motivado nuestra formación académica, su tenacidad y lucha interminable han hecho de ellos un gran ejemplo a seguir por nosotros.

A nuestros profesores, a quienes les debemos gran parte de nuestros conocimientos, gracias por prepararnos para un futuro competitivo no solo como mejores profesionales sino también como mejores personas.

Al Instituto de Educación Superior Tecnológico Público “De Las Fuerzas Armadas”, por ser el centro donde se nos proporcionó conocimiento.

A todas las personas que nos brindaron el apoyo para la culminación del presente trabajo de aplicación profesional.

## Índice

	<b>Página</b>
Carátula	i
Agradecimientos	iv
Índice	v
Índice de figuras	viii
Resumen	x
Introducción	xi
<b>CAPÍTULO I: DETERMINACIÓN DEL PROBLEMA</b>	<b>12</b>
1.1 Formulación del problema	14
1.1.1 Problema general	14
1.1.2 Problemas específicos	14
1.2 Objetivos	14
1.2.1 Objetivo general	14
1.2.2 Objetivos específicos	14
1.3 Justificación del trabajo de aplicación profesional	15
<b>CAPÍTULO II: MARCO TEÓRICO</b>	<b>15</b>
2.1. Estado del arte	17
2.2. Bases teóricas	19
2.2.1. Sistema	19
2.2.2. Aplicaciones de escritorio	20
2.2.3. Control de personal	20
2.2.4. Vigilancia canina	20
2.2.5. Usabilidad	21
2.2.6. Plataforma .NET	21
2.2.7. Visual Studio .Net	22
2.2.8. Visual Basic .Net	23
2.2.9. Base de datos	23

2.2.10.	Sistema de gestión de base de datos	24
2.2.11.	Microsoft SQL Server	25
2.2.12.	Transact - SQL	25
2.2.13.	UML	26
2.2.14.	RUP	26
<b>CAPÍTULO III: DESARROLLO DEL TRABAJO</b>		<b>29</b>
3.1	Finalidad	29
3.2	Propósito	29
3.3	Componentes	29
3.3.1	Levantamiento de la información	30
3.3.2	Análisis de la información	32
3.3.2.1	Análisis del proceso actual	32
3.3.2.2	Análisis del proceso propuesto	33
3.3.3	Diseño del sistema de escritorio	35
3.3.3.1	Estructura del sistema	35
3.3.3.2	Descripción general de la estructura del sistema	36
3.3.3.3	Diagrama de la Base de Datos en SQL Server	45
3.3.4	Descripción de las tablas de la base de datos	46
3.3.5	Codificación del sistema	50
3.3.6	Pruebas del sistema	81
3.3.7	Implementación del sistema	82
3.4	Actividades	82
3.5	Limitaciones	82
<b>CAPÍTULO IV: RESULTADOS</b>		<b>85</b>
Resultados		85
<b>CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES</b>		<b>87</b>
Conclusiones		87

Recomendaciones	88
Referencias Bibliográficas	89
Apéndice A: Cronograma de Actividades	93
Apéndice B: Cronograma de Presupuesto	94
Apéndice C: Manual de Usuario	95

## Índice de figuras

	<b>Página</b>
Figura 1: Sistema de información	19
Figura 2: Seguridad con caninos	21
Figura 3: Framework .Net, representa el corazón de la plataforma .NET	22
Figura 4: Lenguajes de programación principales de Visual Studio .NET	23
Figura 5: Las bases de datos almacenan la información organizada y relacionada	24
Figura 6: Esquema del funcionamiento de un SGBD	24
Figura 7: Arquitectura de bases de datos SQL Server	25
Figura 8: Principales símbolos del UML	26
Figura 9: Fases de la metodología RUP	27
Figura 10: Etapas del desarrollo del Trabajo de aplicación	30
Figura 11: Preguntas formuladas al Gerente de la empresa Security Dog Ormeño	<b>¡Error!</b>
<b>Marcador no definido.</b>	
Figura 12: Preguntas realizadas al encargado de personal y servicio de vigilancia canina	<b>¡Error! Marcador no definido.</b>
Figura 13: Análisis del proceso actual registro de asistencia de personal	<b>¡Error! Marcador no definido.</b>
Figura 14: Análisis del proceso actual servicio de vigilancia canina	<b>¡Error! Marcador no definido.</b>
Figura 15: Análisis del proceso propuesto registro de asistencia de personal	<b>¡Error! Marcador no definido.</b>
Figura 16: Análisis del proceso propuesto servicio de vigilancia canina	<b>¡Error! Marcador no definido.</b>
Figura 17: Estructura general del sistema	36
Figura 18: Formulario de ingreso al sistema – asistencia personal	37
Figura 19: Formulario asistencia de personal	<b>¡Error! Marcador no definido.</b>
Figura 20: Formulario de ingreso al sistema – administrador	38
Figura 21: Formulario Iniciar sesión	38
Figura 22: Formulario principal del sistema	<b>¡Error! Marcador no definido.</b>
Figura 23: Menú principal del sistema	39
Figura 24: Formulario mantenimiento de personal	<b>¡Error! Marcador no definido.</b>



Figura 25: Formulario mantenimiento de canes	<b>¡Error! Marcador no definido.</b>
Figura 26: Formulario mantenimiento de puestos de servicio	<b>¡Error! Marcador no definido.</b>
Figura 27: Formulario mantenimiento de asistencia de personal	<b>¡Error! Marcador no definido.</b>
Figura 28: Formulario mantenimiento turnos del personal	<b>¡Error! Marcador no definido.</b>
Figura 29: Formulario mantenimiento de distribución del personal y canes	42
Figura 30: Formulario mantenimiento de incidencias	<b>¡Error! Marcador no definido.</b>
Figura 31: Formulario reporte de asistencias de personal y canes	<b>¡Error! Marcador no definido.</b>
Figura 32: Formulario información de los desarrolladores	<b>¡Error! Marcador no definido.</b>
Figura 33: Barra con acceso directo a diferentes opciones	44
Figura 34: Diagrama de la base de datos	45
Figura 35: Estructura de la tabla usuario	46
Figura 36: Estructura de la tabla Personal	46
Figura 37: Estructura de la tabla Puesto	46
Figura 38: Estructura de la tabla Can	47
Figura 39: Estructura de la tabla detalle_can	47
Figura 40: Estructura de la tabla tipo_can	47
Figura 41: Estructura de la tabla estado_cant	48
Figura 42: Estructura de la tabla turno	48
Figura 43: Estructura de la tabla asistencia	48
Figura 44: Estructura de la tabla incidencias	49
Figura 45: Estructura de la tabla detalle_incidencia	49
Figura 46: Estructura de la tabla distribución	49

## Resumen

El presente trabajo de aplicación profesional se desarrolló con el propósito de controlar la asistencia de personal y el servicio de vigilancia canina de la empresa Security Dog Ormeño E.I.R.L., ubicada en calle Santa Florencia Nro. 559 Dpto. 401 Urb. Pando Et. tres, del distrito de Lima.

Hoy en día la empresa realiza de forma manual, el control de asistencia de su personal y del servicio de vigilancia canina que brinda, por lo que este trabajo de aplicación profesional pretende implementar un sistema informático de escritorio que permita controlar la asistencia de personal y vigilancia canina de la Empresa Security Dog Ormeño. Así mismo, este sistema será amigable brindando facilidades de manejo para los usuarios.

Para el desarrollo de este trabajo de aplicación se utilizó la metodología RUP (Proceso Unificado de Rational), la misma que utiliza el UML (Lenguaje de Modelado Unificado) y como herramienta CASE al software Rational Rose, así como para la etapa de planificación se utilizó el software Microsoft Project, para elaborar el diagrama de Gantt en el cual se detallan las diferentes etapas, tareas, duración del trabajo además del personal asignado a cada tarea.

**Palabras claves:** Asistencia de personal, vigilancia canina, sistema de escritorio, RUP

## **Introducción**

Este documento describe todo lo realizado en el trabajo de aplicación profesional. Consiste en el desarrollo de un sistema informático de escritorio para la empresa Security Dog Ormeño E.I.R.L. en el distrito de Lima – Lima y tiene como gerente al Sr. José Ormeño Arone. El objetivo del sistema de escritorio es controlar la asistencia de personal de la empresa, así como el servicio de vigilancia canina que brinda.

El sistema de escritorio desarrollado realiza el control de asistencias del personal mediante una ventana que registra nombre, turno, hora de entrada, etc., así como también realiza la distribución del personal en los diferentes puestos donde prestarán el servicio con sus respectivos canes, además de la generación de reportes.

Para el desarrollo del sistema se utilizó el Lenguaje de Programación Visual Basic .Net, así como para el almacenamiento de la información se utilizó el sistema de gestión de base de datos SQL Server.

El desarrollo de este trabajo de aplicación profesional fue estructurado en los siguientes capítulos:

Capítulo I: En este capítulo se realizó el planteamiento y formulación del problema, se describe también la descripción del objetivo general, objetivos específicos y justificación del trabajo de aplicación profesional.

Capítulo II: En este capítulo se mencionan diferentes trabajos como tesis, tesinas, proyectos finales, que han sido elaborados y que guardan relación con el presente trabajo de aplicación profesional; también se especifica la parte teórica que es importante para fundamentar las tecnologías que forman parte del sistema de escritorio.

Capítulo III: En este capítulo se explica el desarrollo del trabajo de aplicación profesional donde se describe la finalidad, el propósito, los componentes, las actividades y las limitaciones que se presentaron durante la ejecución del trabajo de aplicación profesional.

Capítulo IV: En este capítulo se describen los resultados obtenidos del trabajo de aplicación profesional del sistema de escritorio, de tal manera que permitan confirmar los objetivos propuestos.

Capítulo V: En este capítulo se establecen las conclusiones a las que se llegaron luego de la recopilación de información, análisis, diseño del sistema, codificación y pruebas del sistema de escritorio y las recomendaciones que se deben tener en cuenta para el buen funcionamiento del sistema de escritorio.

## **CAPÍTULO I**

### **DETERMINACIÓN DEL PROBLEMA**

## **1.1 Formulación del problema**

Actualmente la empresa Security Dog Ormeño, carece de un sistema informático de escritorio que permita controlar las asistencias de su personal, así como también de los servicios de vigilancia que brinda, todo este proceso se realiza en forma manual, realizándose apuntes en papel por ende no se cuenta con información relacionada y en tiempo real.

### **1.1.1 Problema general**

¿Mejorará el control de asistencia de personal y el servicio de puestos de vigilancia canina, con la implementación de un sistema en la empresa Security Dog Ormeño?

### **1.1.2 Problemas específicos**

¿Cómo mejora el proceso de registro del personal con la implementación de un sistema de escritorio en la empresa “Secutiry Dog Ormeño”?

¿Cómo mejora el proceso de control de asistencia del personal con la implementación de un sistema de escritorio en la empresa “Secutiry Dog Ormeño”?

¿Cómo mejora el proceso del servicio de control de Vigilancia canina con la implementación de un sistema de escritorio en la empresa “Secutiry Dog Ormeño”?

## **1.2 Objetivos**

### **1.2.1 Objetivo general**

Implementar un sistema de control de asistencia de personal y del servicio de vigilancia Canina para la empresa SECURITY DOG ORMEÑO.

### **1.2.2 Objetivos específicos**

Implementar un sistema informático de escritorio para que mejore el proceso de registro de personal en la empresa “Security Dog Ormeño.”

Implementar un sistema informático de escritorio para que mejore el proceso de control de asistencia de personal en la empresa “Security Dog Ormeño.”

Implementar un sistema informático de escritorio para que mejore el proceso de control del servicio de vigilancia canina en la empresa “Security Dog Ormeño.”

### **1.3 Justificación del trabajo de aplicación profesional**

Las dificultades actuales de la empresa Security Dog Ormeño, se centralizan en que el control de asistencia de personal y el servicio de vigilancia canina, se realiza en forma manual realizando apuntes en papel y no se cuenta con información relacionada y en tiempo real.

Con la implementación del sistema se logrará mejorar dichos procesos, ahorrando tiempo.

Siendo los siguientes aspectos que justifican el trabajo de aplicación profesional:

Se mejora el control de asistencias del personal de la empresa

Se mejora el control del servicio de vigilancia canina, además de una correcta distribución de personal y canes.

Al desarrollar el presente trabajo de aplicación profesional traerá beneficios a la empresa y los mantendrá a la vanguardia de la tecnología.

**CAPÍTULO II**  
**MARCO TEÓRICO**



## 2.1. Estado del arte

### Antecedentes de estudio

Se han realizado búsqueda de diferentes tipos de trabajos como tesis, monografías, etc.; que guarden relación con el trabajo de aplicación profesional que venimos desarrollando, a continuación, mencionaremos dichos trabajos encontrados:

(Manrique, 2015), en su tesis: “Optimizar la administración de la brigada canina mediante el desarrollo de un sistema de gestión web”, Explicó que, para el desarrollo de la aplicación web, siguió una serie de pasos basados en un metodología especial de RUP ((Rational Unified Process), que recogió toda la información posible proporcionada por la Municipalidad de Lima en una primera fase. Por lo tanto, las partes relevantes llegaron a un acuerdo sobre los objetivos del proyecto y luego elaboraron. Básicamente, el objetivo de la segunda fase es establecer una infraestructura del sistema con el fin de proporcionar una base estable para el diseño e implementación del proyecto. Las siguientes fases, luego compilar y finalmente ingresar a la fase de transición, el enfoque de esta fase es garantizar que el software esté disponible para la Municipalidad Metropolitana de Lima.

(Cantilla, Inostroza, 2016), en su memoria: “Sistema de control de asistencia de personal de la Universidad del Bío-Bío”, explican que el objetivo del sistema es facilitar a los funcionarios administrativos de Universidad de Bío-Bío la tarea de marcar su ingreso y salida a su jornada laboral, permitiéndole realizar dichos marcajes en cualquier parte del campus a través de una aplicación móvil instalada en su Smartphone, o en la comodidad de su oficina a través de su computador personal. Comentan que esta idea surgió luego de ver el problema que sufren actualmente los funcionarios administrativos de la Universidad del Bio-Bio los cuales tiene como obligación marcar su ingreso y salida en los distintos sensores biométricos ubicados en la universidad, para el desarrollo del sistema utilizaron como metodología principal Cascada en adición con la metodología Crystal.

Indican también que construyeron una Aplicación Móvil y una Plataforma Web, la primera de ellas encargada de proporcionar a los usuarios con Smartphone y acceso a internet, la posibilidad de marcar su ingreso y salida al trabajo, ver sus horas trabajadas en un intervalo de fechas,) consultar su ubicación actual y el punto más cercano de marcado y ver sus marcajes realizados en el día.

(Huanca, 2016), en su tesis: “Implementación de un sistema de control biométrico para la institución educativa San Martín de Porras La Victoria – Huarmey; 2017”, sustenta que, el objetivo general del trabajo es la realización de una implementación de un sistema de control biométrico para la Institución Educativa San Martín de Porras, para que permita tener un mejor control en la asistencia de personal, nombrado, contratado, docentes y administrativos de esta institución.

Afirma que el tipo de investigación fue cuantitativa y descriptiva, con diseño no experimental ya que propone como alternativa de solución la implementación de un sistema biométrico; y de corte transversal porque se está tomando como punto referencial el año 2017.

Indica que como resultado se obtuvo que el 82% del personal encuestado manifestaron que no se sienten satisfechos como se viene realizando el control de asistencia con registro en libros o cuadernos y luego contabilizarlos para elaborar el informe a la UGEL Huarmey, Así mismo, el 100% del personal encuestado están de acuerdo con una propuesta de mejora como es la implementación de un sistema de control biométrico, por lo cual su hipótesis general queda aceptada.

(Iparraguirre, Mendoza, 2018), en su tesis: “Diseño e Implementación de un Sistema Informático para el Proceso de Comercialización y Control de Asistencia del Personal mediante Dispositivo Biométrico, en la Botica “LIZFARMA” – José Leonardo Ortiz.”, explican, que este trabajo se ha realizado con la finalidad de resolver el problema del manejo de información que se presentan en las áreas de comercialización y asistencia de personal. Indican que para el desarrollo del software utilizaron la metodología RUP (Rational Unified Process), con la finalidad de analizar los diferentes procesos que se realizan en la organización y determinar los requerimientos necesarios y plantear la solución a través del desarrollo de una aplicación, con la aplicación desarrollada lograron administrar de forma óptima, el proceso de comercialización, controlando los productos que se compran y venden así como los que se mantienen en almacén, por otro lado en el acceso a la organización han implementado un lector biométrico que permite controlar el ingreso y salida del personal. Finalmente, a través del Análisis de Retorno de Inversión (ROI), determinaron que el proyecto es viable para la organización y recuperable en corto plazo.

(Tusa, 2019), en su proyecto de aplicación: “La automatización de procesos y su incidencia en el control de asistencia docente en la unidad educativa Darío Guevara, del cantón ambato provincia de Tungurahua”, sustenta, que el proyecto está enfocado en el desarrollo de una aplicación Automatizada que permite el control de asistencia docente y demás procesos de control docente. La investigación contiene aspectos importantes sobre la automatización de procesos y su incidencia en el control de asistencia docente, esto permite mejorar la toma de decisiones por parte del área administrativa, de esta manera se lleva información actualizada y correcta, la unidad educativa Darío Guevara, se encuentra en la necesidad de utilizar nuevas herramientas tecnológicas, reemplazando el sistema actual por un sistema automatizado que facilite el almacenamiento y acceso a la información, sin necesidad de desperdiciar, tiempo, recursos y esfuerzo y den resultados oportunos y libres de errores de tal manera que no provoquen malestar entre los docentes de la institución, en el momento de registrarse.

## 2.2. Bases teóricas

### 2.2.1. Sistema

Un sistema es un conjunto de componentes que interaccionan entre sí para lograr un objetivo común. (Fernández, 2006, p. 11)

De acuerdo a la definición anterior, se refiere a que, un sistema es una agrupación de partes que forman un engranaje para obtener un fin.



Figura 1: Sistema de información  
Fuente: INNOVADORAS.COM

### **2.2.2. Aplicaciones de escritorio**

Las aplicaciones de escritorio son programas que se instalan directamente en el ordenador y que no necesitan conexión a Internet para poder trabajar con ellas. Estas aplicaciones te pueden ayudar en muchas tareas como en la gestión de pedidos, el control de stock, la gestión del personal o la comunicación empresarial interna o externa. (Clickage, 2018)

Después de lo anterior, se deduce que, las aplicaciones de escritorio son aplicaciones son programas instalables en máquinas de manera local, sin necesidad de que éstas tengan conexión a internet y que nos facilitan el procesamiento de diferentes tareas.

### **2.2.3. Control de personal**

El control de personal o control de colaboradores permite integrar una serie de procedimientos con la finalidad de registrar y controlar al personal que trabaja en una empresa o institución. (Kimaldi, 2019)

Dadas las condiciones que anteceden, se infiere, que el control de personal consiste en la reunión de diferentes procedimientos que tienen como único propósito controlar el ingreso, las actividades que realiza, así como la salida del personal que labora en una empresa u organización.

### **2.2.4. Vigilancia canina**

Es un complemento al servicio de vigilancia física y es una herramienta efectiva para la defensa y protección ante situaciones de riesgo. (Seguridad Atempí, 2019)

Con referencia a lo anterior, se infiere que, la vigilancia canina representa un servicio muy importante complementario a la vigilancia que brindan las personas, de forma efectiva en momentos de peligro.



*Figura 2: Seguridad con caninos*  
*Fuente: Grupo Atlas de seguridad integral*

### **2.2.5. Usabilidad**

El concepto de usabilidad se utiliza con frecuencia en el ámbito de la tecnología y la informática. Proviene del inglés usability y significa la facilidad con la que un usuario puede usar una herramienta fabricada por otras personas con el propósito de conseguir un objetivo. (economía simple.net, 2018)

Después de lo anterior, se deduce que, Usabilidad es la facilidad que tiene un usuario para interactuar con una determinada herramienta tanto hardware como software.

### **Tecnologías Utilizadas**

Para la implementación de este trabajo de aplicación profesional se ha utilizado una combinación de diferentes tecnologías:

### **2.2.6. Plataforma .NET**

La plataforma .NET es un amplio conjunto de bibliotecas de desarrollo que pueden ser utilizadas con el objetivo principal de acelerar el desarrollo de software y obtener de manera automática características avanzadas de seguridad, rendimiento, etc. (CampusMVP, 2017)

De acuerdo a la definición anterior, se afirma que, la plataforma .NET representa una colección de bibliotecas de clases que proporcionan a los programadores un conjunto de herramientas para la elaboración de software de alta performance.

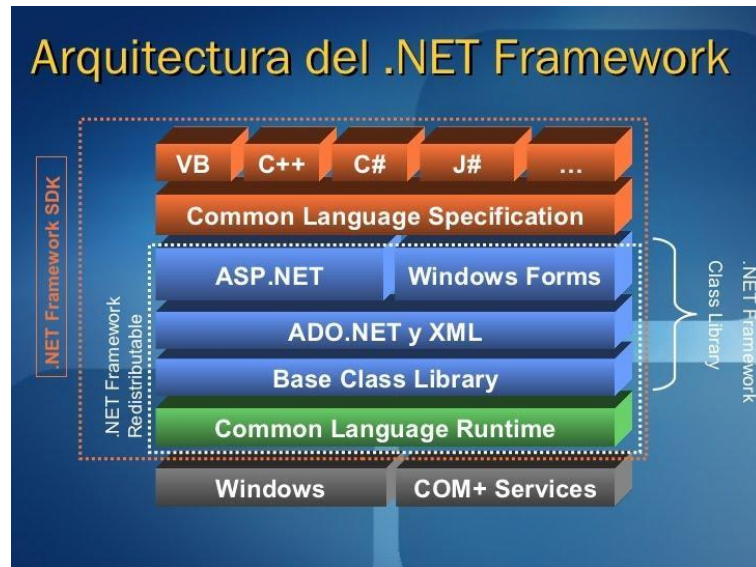
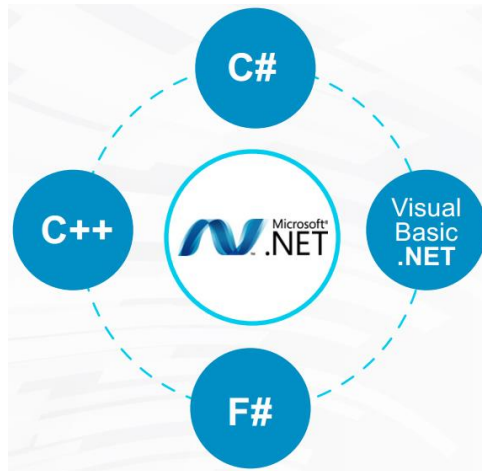


Figura 3: Framework .Net, representa el corazón de la plataforma .NET  
Fuente: Programación .NET C#

### 2.2.7. Visual Studio .Net

Microsoft Visual Studio es un entorno de desarrollo integrado, creado por la compañía Microsoft y disponible para sistemas operativos Windows, Linux y macOS, y la vez es compatible con múltiples lenguajes de programación, tales como C++, C#, Visual Basic .NET, F#, Java, Python, Ruby y PHP, al igual que entornos de desarrollo web, como ASP.NET, fue lanzado en 1997, cuenta con versiones gratis y de venta. (EspacioHonduras, 2020)

Después de lo anterior, se deduce que, Visual Studio .Net es un IDE que nos proporciona un marco de trabajo para el desarrollo de aplicaciones Windows y web empleando diferentes lenguajes de programación como Visual Basic .Net, C# .Net, C++ .Net, etc., además este IDE presenta versiones gratuitas y licenciadas.



*Figura 4:* Lenguajes de programación principales de Visual Studio .NET  
Fuente: .Northware

### 2.2.8. Visual Basic .Net

Es un lenguaje de programación orientado a objetos que se puede considerar una evolución de Visual Basic implementada sobre el framework .NET. (VisualBasic.NET, 2014)

De acuerdo a la definición anterior, se afirma que, Visual Basic .Net es un lenguaje de programación que nos permite aplicar todos los conceptos de programación orientado a objetos y además nos indica que es un lenguaje que ha evolucionado y ahora tiene como corazón al framework .NET.

### 2.2.9. Base de datos

Una base de datos es el sitio donde se almacena un conjunto de datos o informaciones, que pueden ser texto, imágenes, videos, sonidos, etc. Estos datos e informaciones se encontrarán relacionados entre sí y podrán ser consultados y estar accesibles en cualquier momento por los usuarios que tengan accesos a estos. (García, 2015, p. 17)

Con referencia a lo anterior, se infiere que, una base de datos es un conjunto de información relacionada y almacenada y que puede ser consultada, dicha información puede ser texto, imágenes, sonido, vídeos, etc.

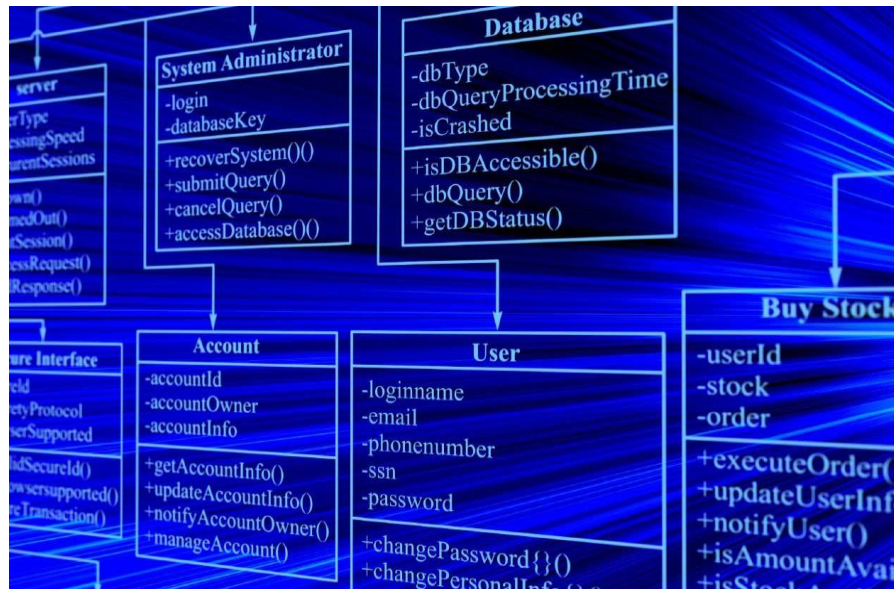


Figura 5: Las bases de datos almacenan la información organizada y relacionada  
 Fuente: Hostname

**2.2.10. Sistema de gestión de base de datos**

El Sistema de gestión de base de datos (SGBD) es una aplicación que permite a los usuarios definir, crear y mantener bases de datos, proporcionando acceso controlado a las mismas. Es una herramienta que sirve de interfaz entre el usuario y las bases de datos. (Hueso, 2015, p. 10)

Después de lo anterior, se deduce que, un sistema de gestión de base de datos es un programa que permite crear y administrar base de datos, representa un puente entre el usuario y las bases de datos.

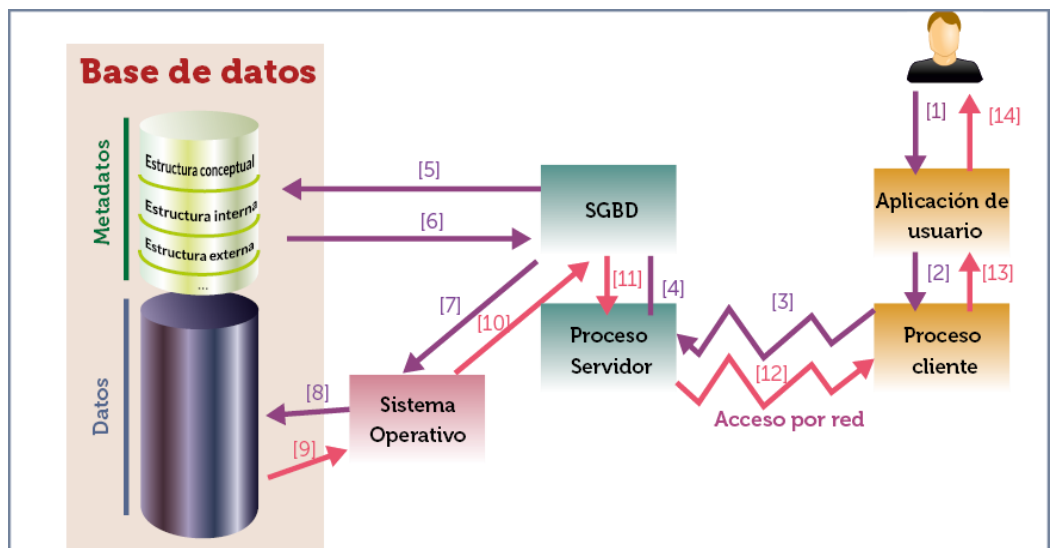


Figura 6: Esquema del funcionamiento de un SGBD  
 Fuente: JorgeSanchez.net



### 2.2.11. Microsoft SQL Server

Microsoft SQL Server es un sistema que sirve para la gestión de base de datos, ha sido producido por Microsoft basado en el modelo relacional y utiliza para sus consultas los lenguajes T-SQL y ANSI SQL. (CISSET, 2020)

De acuerdo a la definición anterior, se afirma que, SQL Server es un software creado por la empresa Microsoft el cual nos permite crear bases de datos de tipo relacional y que permite administrar bases de datos y tiene como lenguaje principal a Transact-SQL.

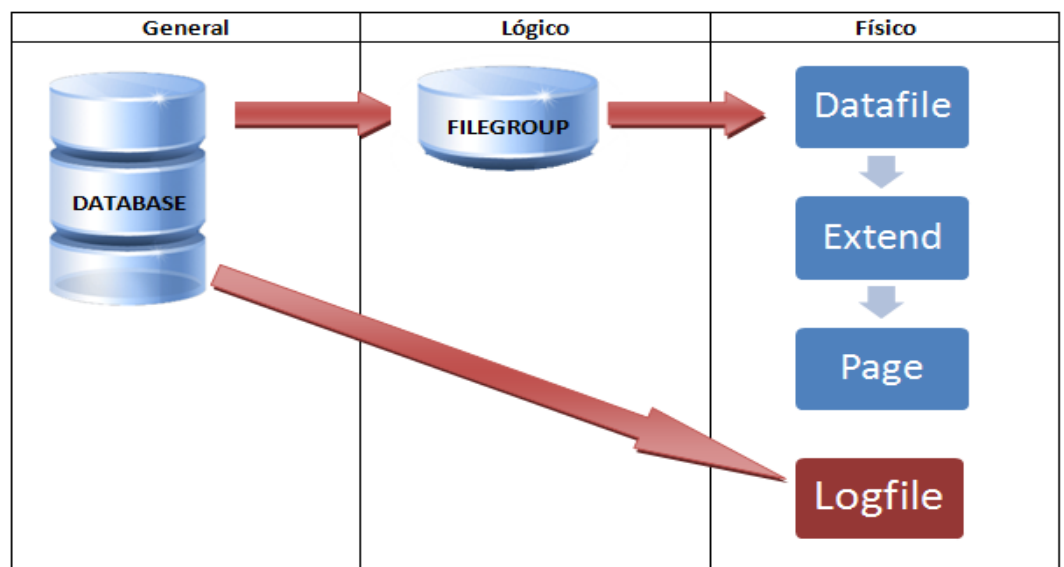


Figura 7: Arquitectura de bases de datos SQL Server  
Fuente: Memorias de un DBA

### 2.2.12. Transact - SQL

El lenguaje Transact SQL o T-SQL es utilizado en la base de datos SQL Server de Microsoft, con características propias de un lenguaje de programación. (Kyocode, 2019)

Con referencia a lo anterior, se infiere que, el lenguaje Transact-SQL es el lenguaje propio de SQL Server, utilizado para crear y gestionar las diferentes estructuras de la base de datos.

### 2.2.13. UML

El lenguaje de modelado unificado (UML) es un estándar para la representación visual de objetos, estados y procesos dentro de un sistema. Por un lado, el lenguaje de modelado puede servir de modelo para un proyecto y garantizar así una arquitectura de información estructurada; por el otro, ayuda a los desarrolladores a presentar la descripción del sistema de una manera que sea comprensible para quienes están fuera del campo. (Digital Guide IONOS, 2018).

Después, de lo anterior, se deduce que, el lenguaje de modelado unificado es utilizado para el análisis de proyectos empleando símbolos como: objetos, líneas de conexión, etc.; que permiten modelar los diferentes procesos que se presentan en un proyecto.

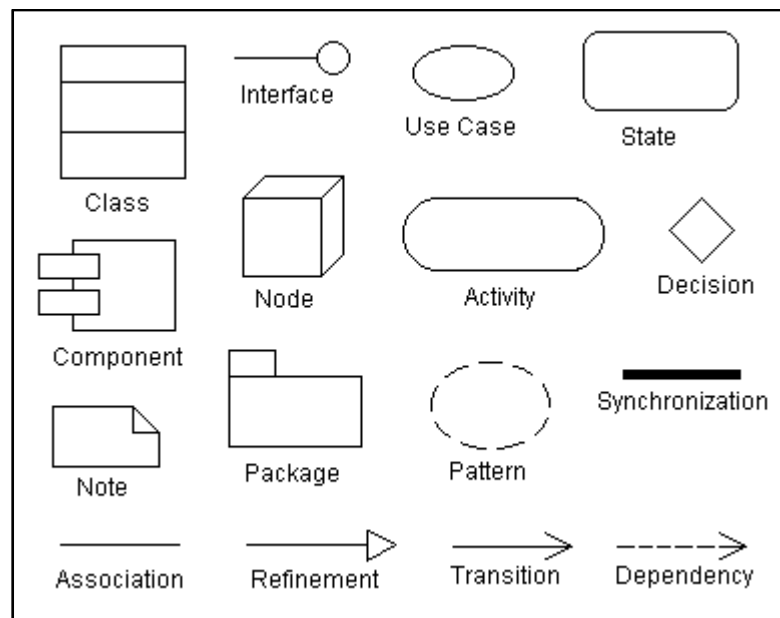


Figura 8: Principales símbolos del UML

Fuente: researchgate.net

### 2.2.14. RUP

El Proceso Unificado Racional (Rational Unified Process en inglés, habitualmente resumido como RUP) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. (Programa en Línea, 2018)

De acuerdo a la definición anterior, se afirma que, el proceso unificado racional es una metodología de desarrollo de software orientada a objetos, que utiliza el lenguaje de modelado unificado como herramienta para modelar los procesos, esta metodología abarca diferentes etapas como análisis, implementación y documentación.

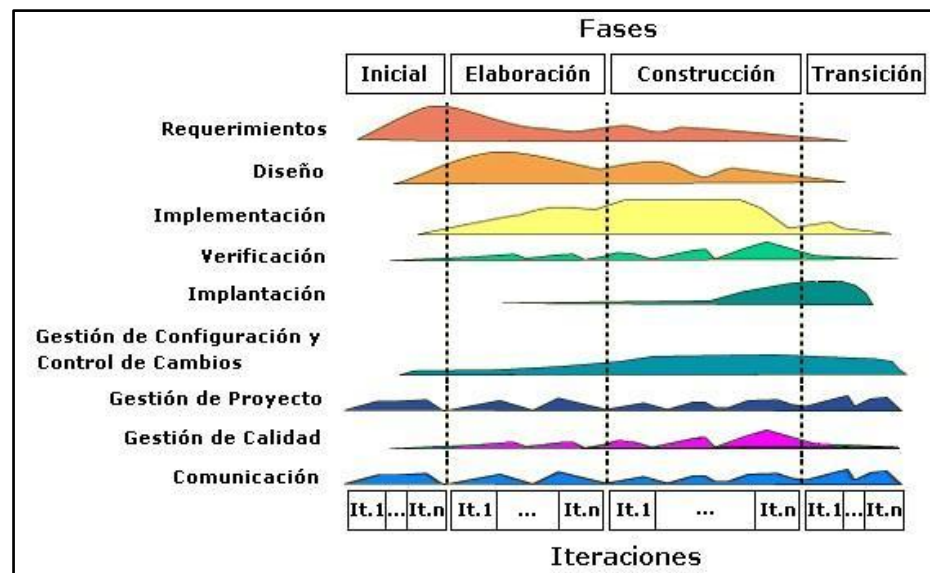


Figura 9: Fases de la metodología RUP

Fuente: fmg

**CAPÍTULO III**  
**DESARROLLO DEL TRABAJO**

### 3.1 Finalidad

La finalidad del presente trabajo de aplicación profesional es mejorar el control de asistencia de personal y el servicio de vigilancia canina, automatizando dichos procesos de tal manera que se cuente con información en tiempo real para una buena toma de decisiones por parte del gerente de la empresa.

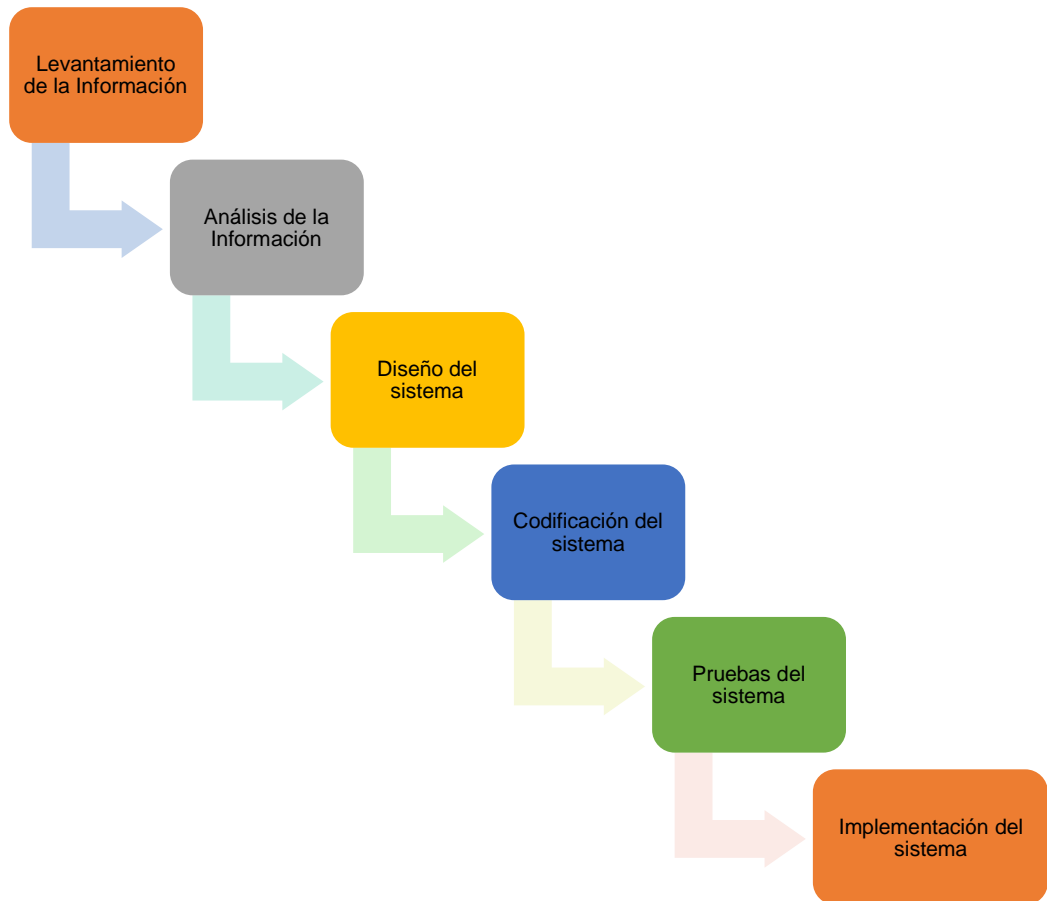
### 3.2 Propósito

Las herramientas utilizadas para el desarrollo del sistema de escritorio se mencionan a continuación:

- **Rational Rose:** Esta herramienta de diseño de software orientada a objetos permite realizar el modelado visual del sistema, principalmente de los casos de uso con la finalidad de tener un sistema de escritorio para la empresa Security Dog Ormeño, utilizando el lenguaje UML.
- **SQL Server:** El sistema de gestión de base de datos se utilizó para crear la base de datos junto con sus tablas y demás estructuras de la empresa Security Dog Ormeño, empleando el lenguaje Transact-SQL.
- **Visual Basic .Net:** El lenguaje de programación se utilizó para codificar los diferentes formularios que forman parte del sistema de escritorio de la empresa Security Dog Ormeño.
- **Microsoft Project:** Esta aplicación de software nos brinda herramientas para la gestión de proyectos, aquí se colocaron las actividades del trabajo de aplicación profesional cada una con su fecha de inicio, fecha fin, duración además del orden en que se ejecutarán dichas actividades para la implementación del sistema de escritorio de la empresa Security Dog Ormeño.

### 3.3 Componentes

Para el desarrollo del sistema de escritorio de la empresa Security Dog Ormeño, se realizaron las siguientes etapas como: Levantamiento de la información, Análisis de la información, Diseño del sistema, codificación del sistema y finalmente las pruebas del sistema.



*Figura 10:* Etapas del desarrollo del Trabajo de aplicación  
*Fuente:* Elaboración propia

A continuación, se describen cada una de las etapas realizadas durante el desarrollo del sistema de escritorio:

### **3.3.1 Levantamiento de la información**

Para el levantamiento de la información se plantearon visitas a la empresa Security Dog Ormeño, con la finalidad de entrevistar al gerente Sr. José Ormeño Arone, y al encargado del control de personal y del servicio de vigilancia canina, al gerente se le realizaron una serie de preguntas con el propósito de tener un panorama de la situación actual, así como los requerimientos del usuario.

**CUESTIONARIO DE PREGUNTAS REALIZADAS AL GERENTE DE  
LA EMPRESA SECURITY DOG ORMEÑO**

¿A qué se dedica la empresa?

¿La empresa actualmente cuenta con un sistema informático?

¿Qué procesos desea automatizar?

¿Cómo se realiza actualmente el proceso de control de personal?

¿Cómo se realiza actualmente el proceso de control del servicio de vigilancia canina?

¿Qué problemas se vienen presentando en el control de personal y en el servicio de vigilancia canina

Posterior a la entrevista, el gerente de la empresa se mostró interesado en implementar un sistema que permita controlar al personal y el servicio de vigilancia canina, ya que es necesario tener la información almacenada la misma que puede ser consultada en cualquier momento a través de un sistema.

Durante el levantamiento de la información se consideró necesario entrevistar también al usuario encargado del control de personal y del servicio de vigilancia canina, para lo cual se le formularon las siguientes interrogantes:

**CUESTIONARIO DE PREGUNTAS REALIZADAS AL USUARIO  
ENCARGADO DEL CONTROL DE PERSONAL Y DEL  
SERVICIO DE VIGILANCIA CANINA**

¿Cuál es su actividad laboral específica en la empresa?

¿Cómo se realiza actualmente el proceso de control de personal?

¿Cómo son los horarios del personal?

¿Qué labor desempeñan?

¿Cómo se realiza actualmente el proceso de control del servicio de vigilancia canina?

¿Qué datos se registran de los canes?

¿Qué problemas se vienen presentando en el control de personal y en el servicio de vigilancia canina

### **3.3.2 Análisis de la información**

El análisis de la información fue realizado teniendo en cuenta los procesos actuales y los procesos propuestos.

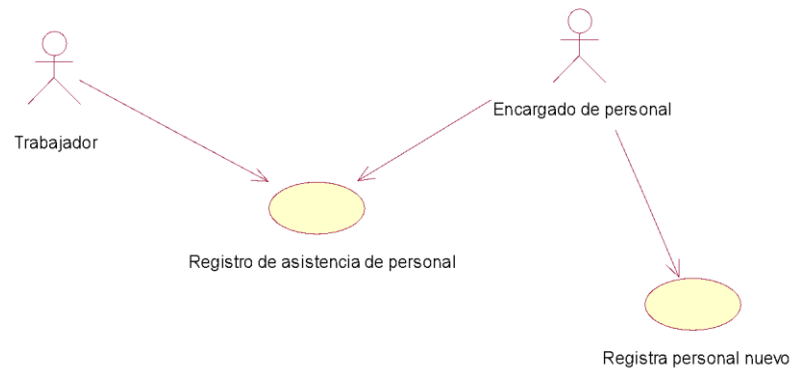
#### **3.3.2.1 Análisis del proceso actual**

En el análisis de los procesos actuales, se identificaron los problemas que se presentaban, los mismos que fueron plasmados a través de un diagrama de casos de uso con ayuda del software Rational Rose. Los cuales se mencionan a continuación:

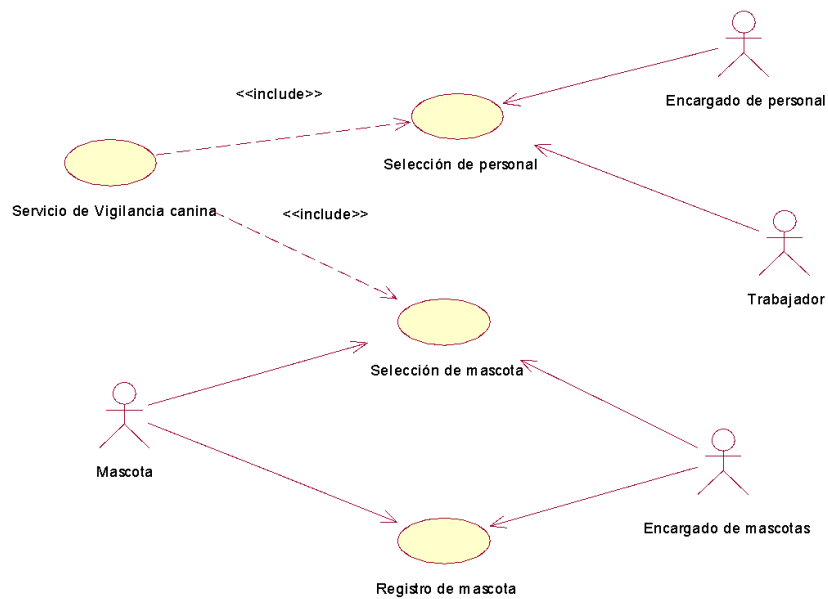
- El proceso de registro de personal se realiza en forma manual.
- El proceso de control del servicio de vigilancia canina se realiza en forma manual.



### Proceso actual: Registro de asistencia de personal



### Proceso actual: Servicio de vigilancia canina



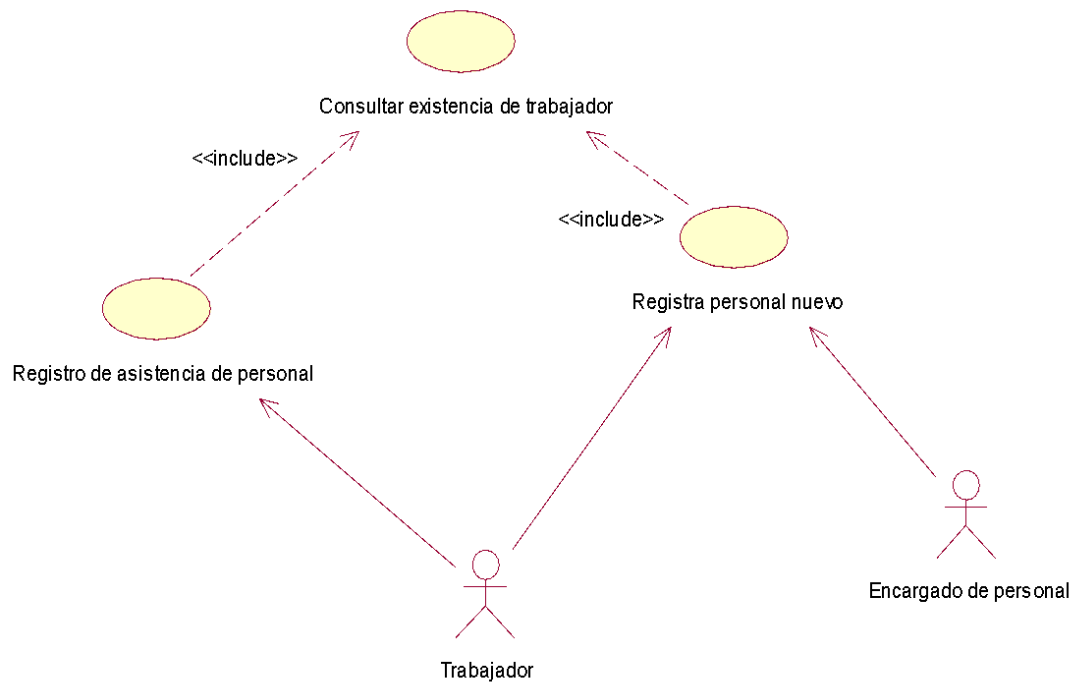
#### 3.3.2.2 Análisis del proceso propuesto

En el análisis de los procesos propuestos se tuvo en cuenta la información recopilada en la etapa del Levantamiento de la información, luego se creó el diagrama de casos de uso de los procesos mejorados que serán implementados en la empresa Security Dog Ormeño, presentándose así los

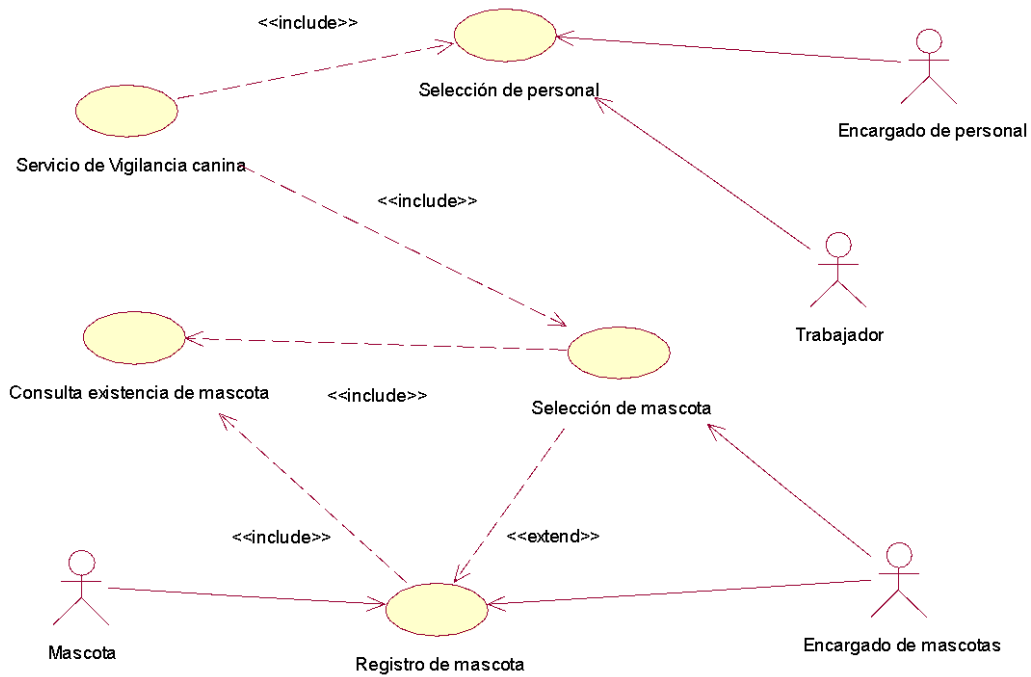
procesos propuestos para este trabajo de aplicación profesional, se mencionan a continuación:

- Elaboración de un sistema para el control de asistencia de personal
- Automatizar el control del servicio de vigilancia canina.

### Proceso propuesto: Registro de asistencia de personal



### Proceso propuesto: Servicio de vigilancia canina

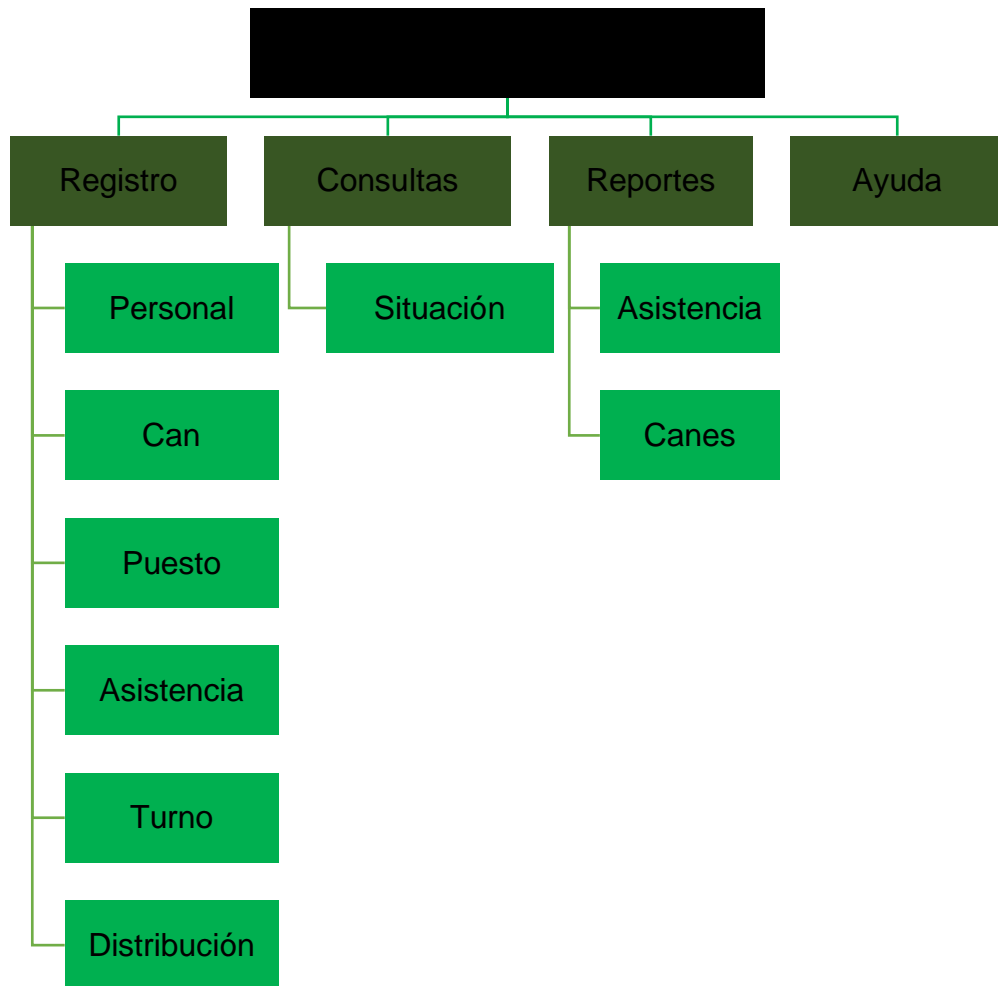


### 3.3.3 Diseño del sistema de escritorio

Para el diseño del sistema de la empresa Security Dog Ormeño se tuvo en cuenta los requerimientos de los usuarios, diseñándose una ventana de ingreso al sistema, el menú principal, el mismo que hace un llamado a los diferentes formularios consultas y reportes, teniendo en cuenta la usabilidad y organización del sistema, para ello se utilizaron herramientas como el IDE de Visual Studio, así como el Crystal Report.

#### 3.3.3.1 Estructura del sistema

El sistema de escritorio se encuentra distribuido de acuerdo al siguiente menú principal:



*Figura 17: Estructura general del sistema  
Fuente: Elaboración propia*

### 3.3.3.2 Descripción general de la estructura del sistema

Se describen cada una de las partes que conforman el sistema.

Empezaremos describiendo el formulario Iniciar sesión, luego se explica cada una de las opciones del menú principal.

Al ejecutar el Sistema de control Ormeño se muestra la siguiente ventana, la misma que nos muestra dos botones uno para marcar la asistencia del personal y el otro que permite realizar las diferentes tareas de administrador.



Figura 18: Formulario de ingreso al sistema – asistencia personal  
Fuente: Elaboración propia

Al hacer clic en el botón Asistencia Personal se muestra la siguiente ventana, la misma que permite que cada personal pueda marcar su asistencia cada vez que ingresa a realizar sus actividades laborales.

	Asistencia	DNI	Nombres	Turno	HoraIngreso	FechaIngreso	Observaciones
*							

Al ingresar si hacemos clic en el botón Administrador



Figura 20: Formulario de ingreso al sistema – administrador  
Fuente: Elaboración propia

Se muestra la ventana de Bienvenido al Sistema:

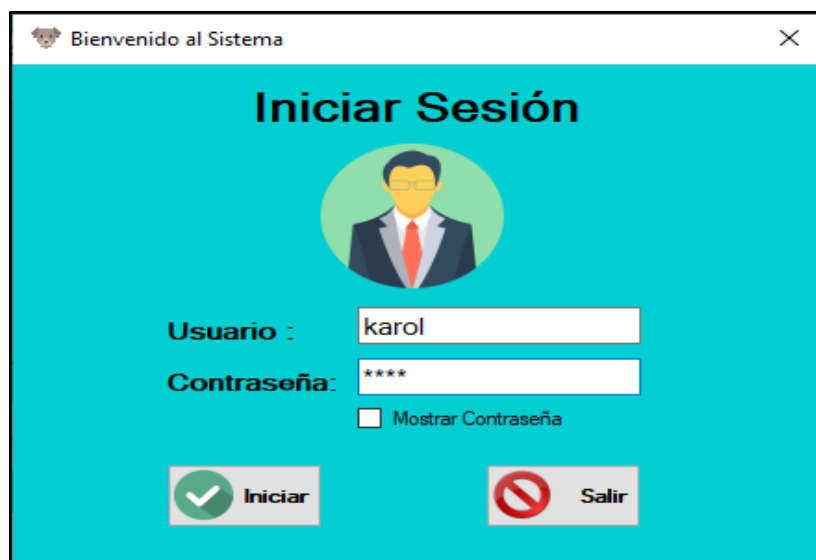
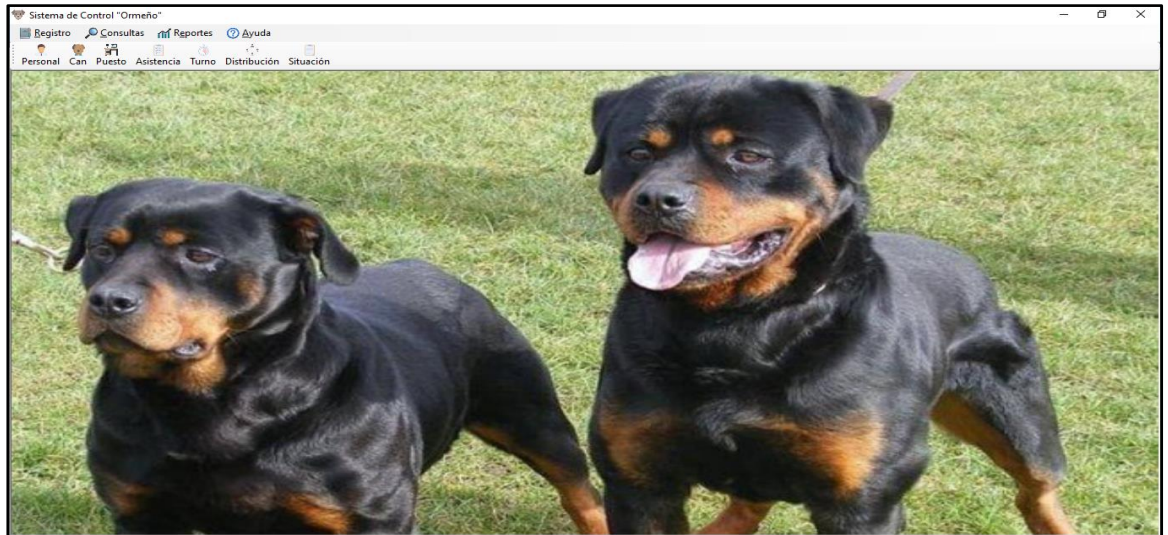


Figura 21: Formulario Iniciar sesión  
Fuente: Elaboración propia

Aquí debemos colocar el usuario y contraseña del administrador y luego darle clic en el botón Iniciar.

Si damos clic en el botón Aceptar, se muestra la ventana principal del sistema:



En la parte superior se muestra el siguiente menú:

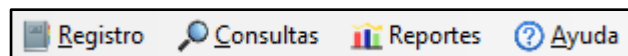


Figura 23: Menú principal del sistema

Fuente: Elaboración propia

### Opción Registro

Al hacer clic aparecen las siguientes sub opciones:

- **Personal:** Aquí registramos al personal que trabaja en la empresa, aquí podemos realizar el mantenimiento a los registros del personal de la empresa

- **Can:** Permite registrar a los datos de los canes que brindan el servicio de vigilancia canina, datos como nombre, color, raza, fecha de nacimiento, así como una imagen del can.



The screenshot shows a window titled "Can" with a light blue background. The main heading is "Registro de Canes". The form includes the following fields and controls:

- Codigo Can :** A text input field.
- Nombre :** A text input field.
- Color :** A text input field.
- Raza :** A text input field.
- Estado :** A dropdown menu.
- Fecha :** A date input field with the value "8/01/2021".
- Imagen :** A large empty rectangular area with a small "..." button to its right.

At the bottom of the window, there are four buttons: "Nuevo" (with a dog icon), "Modificar" (with a paw print icon), "Registrar" (with a dog icon), and "Eliminar" (with a dog icon).

- **Puesto:** Se registran los diferentes puestos donde se presta el servicio de vigilancia canina.



The screenshot shows a window titled "Puesto" with a light blue background. The main heading is "Puestos de Servicio". The form includes the following fields and controls:

- Codigo Puesto:** A text input field.
- Nombre :** A text input field.
- Ubicación :** A text input field.

At the bottom of the window, there are four buttons: "Nuevo" (with a person icon), "Modificar" (with a person icon), "Registrar" (with a person icon), and "Eliminar" (with a person icon and a red 'X' over it).

- **Asistencia:** Permite registrar la asistencia del personal, aquí podemos dar mantenimiento a los registros de asistencia del personal.



Asistencia

Fecha de Ingreso : 30/10/2020

## Asistencia del Personal

Asistencia:

Nombre:

Turno :

Observaciones :

Hora de Ingreso : 10:12:35

	Asistencia	DNI	Nombres	Turno	HoraIngreso	FechaIngreso	Observaciones
*							

Nuevo Modificar Registrar Eliminar

- **Turno:** esta opción nos permite ingresar y dar mantenimiento a los turnos del personal cuando prestan el servicio de vigilancia canina, los turnos pueden ser mañana, tarde y noche.

Turno

## Turno del Personal

Codigo Turno :

Turno :

Nuevo Modificar Registrar Eliminar

- **Distribución:** Esta opción representa una de las opciones más importantes del sistema, ya que es donde se especifica la distribución tanto del personal como de los canes y en que puesto se brindará el servicio de vigilancia canina.

**Distribución del Personal**

**Personal**  
 DNI:   
 Nombre:   
 Apellidos:   
 Buscar personal:

**PUESTO**  
 Fecha de Inicio: 6/11/2020  
 Puesto:   
 Buscar puesto:

**CAN**  
 Hora de Inicio: 08:15:12  
 Nombre Can:   
 Buscar Can:

Seleccionar	DNI	Nombres	Apellidos
<input type="checkbox"/>			

Seleccionar	Cod	Puesto
<input type="checkbox"/>		

Seleccionar	Cod	Nombre
<input type="checkbox"/>		

**Nuevo** **Modificar** **Registrar** **Eliminar**

Cod	Dni	Nombres	Apellidos	Puesto	Ubicacion	Can	FechaInicio	HoraInicio
▶▶								

Figura 29: Formulario mantenimiento de distribución del personal y canes  
 Fuente: Elaboración propia

### Opción Consultas

Al hacer clic aparece la siguiente sub opción:

- **Situación:** Permite registrar algún tipo de incidencias ocurridas durante el servicio de vigilancia canina.

**Incidencias** Fecha: 8/01/2021

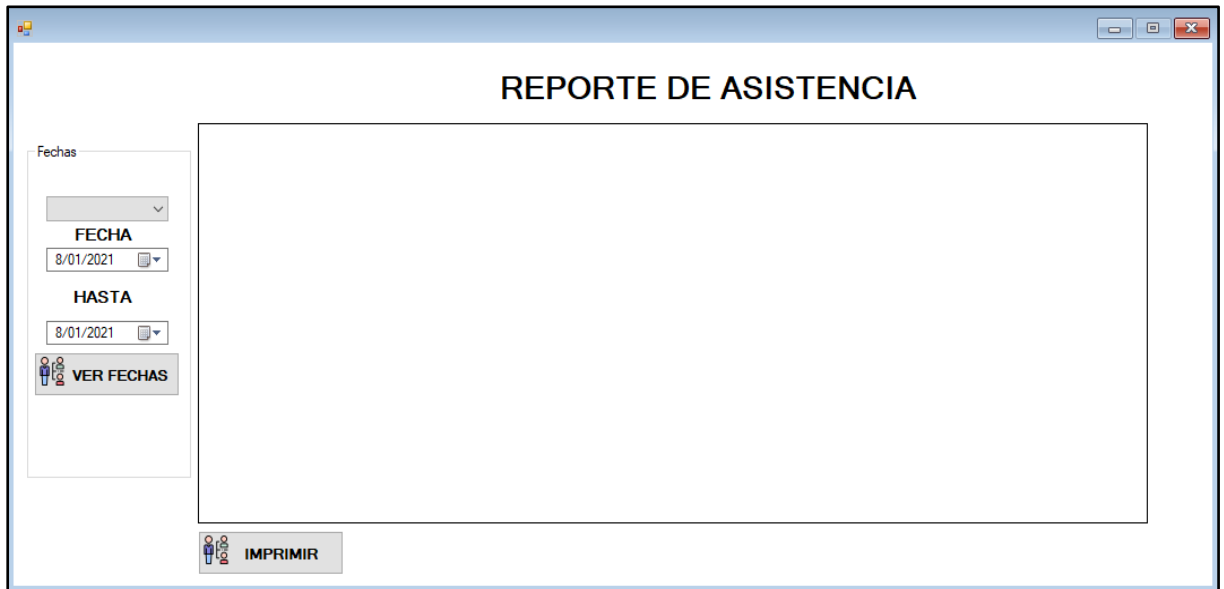
Incidencias:  **Nuevo**

Fecha de asistencia: 8/01/2021  
 DNI:   
 Nombres y apellidos:   
 Estado:    
 Turno:   
 Descripción:

**Modificar** **Registrar** **Eliminar**

### Opción Reportes

- **Asistencias y Canes:** En esta opción se muestra un reporte de asistencias y de canes.



### Opción Ayuda

Al hacer clic aparece la siguiente sub opción:

- **Acerca de:** En esta opción se presenta el grupo de desarrolladores del sistema de control Ormeño.



En la parte superior de la ventana principal se muestra la siguiente barra con diferentes accesos rápidos a las diferentes opciones ya descritas anteriormente.

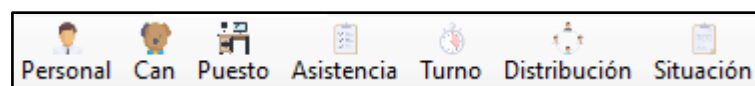


Figura 33: Barra con acceso directo a diferentes opciones

*Fuente:* Elaboración propia

### 3.3.3.3 Diagrama de la Base de Datos en SQL Server

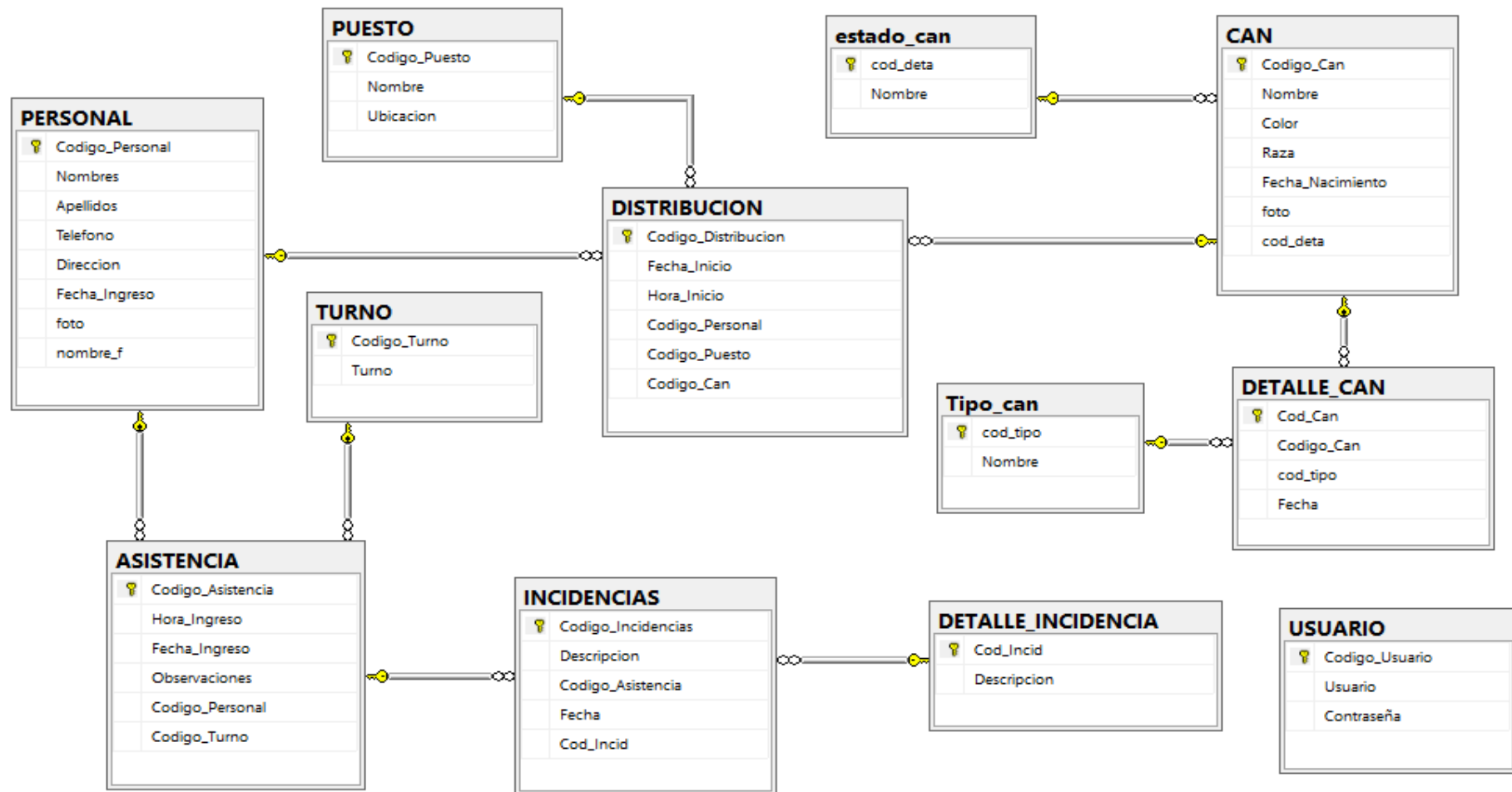


Figura 34: Diagrama de la base de datos

Fuente: Elaboración propia

### 3.3.4 Descripción de las tablas de la base de datos

**Tabla: Usuario**

USUARIO			
	Column Name	Data Type	Allow Nulls
🔑	Codigo_Usuario	int	<input type="checkbox"/>
	Usuario	varchar(50)	<input type="checkbox"/>
	Contraseña	varchar(50)	<input type="checkbox"/>
			<input type="checkbox"/>

*Figura 35: Estructura de la tabla usuario  
Fuente: Elaboración propia*

**Tabla: Personal**

PERSONAL			
	Column Name	Data Type	Allow Nulls
🔑	Codigo_Personal	char(8)	<input type="checkbox"/>
	Nombres	varchar(25)	<input type="checkbox"/>
	Apellidos	varchar(25)	<input type="checkbox"/>
	Telefono	varchar(25)	<input type="checkbox"/>
	Direccion	varchar(25)	<input type="checkbox"/>
	Fecha_Ingreso	varchar(15)	<input type="checkbox"/>
	foto	varchar(MAX)	<input type="checkbox"/>
	nombre_f	varchar(20)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

*Figura 36: Estructura de la tabla Personal  
Fuente: Elaboración propia*

**Tabla: Puesto**

PUESTO			
	Column Name	Data Type	Allow Nulls
🔑	Codigo_Puesto	int	<input type="checkbox"/>
	Nombre	varchar(25)	<input type="checkbox"/>
	Ubicacion	varchar(25)	<input type="checkbox"/>
			<input type="checkbox"/>

*Figura 37: Estructura de la tabla Puesto  
Fuente: Elaboración propia*

**Tabla: Can**

CAN			
	Column Name	Data Type	Allow Nulls
🔑	Codigo_Can	int	<input type="checkbox"/>
	Nombre	varchar(25)	<input type="checkbox"/>
	Color	varchar(25)	<input type="checkbox"/>
	Raza	varchar(25)	<input type="checkbox"/>
	Fecha_Nacimiento	date	<input type="checkbox"/>
	foto	varchar(MAX)	<input type="checkbox"/>
	cod_deta	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Figura 38: Estructura de la tabla Can

Fuente: Elaboración propia

**Tabla Detalle\_Can**

DETALLE_CAN			
	Column Name	Data Type	Allow Nulls
🔑	Cod_Can	int	<input type="checkbox"/>
	Codigo_Can	int	<input checked="" type="checkbox"/>
	cod_tipo	int	<input checked="" type="checkbox"/>
	Fecha	date	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Figura 39: Estructura de la tabla detalle\_can

Fuente: Elaboración propia

**Tabla: Tipo\_Can**

Tipo_can			
	Column Name	Data Type	Allow Nulls
🔑	cod_tipo	int	<input type="checkbox"/>
	Nombre	varchar(55)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Figura 40: Estructura de la tabla tipo\_can

Fuente: Elaboración propia

**Tabla: Estado\_Cant**

estado_can			
	Column Name	Data Type	Allow Nulls
🔑	cod_deta	int	<input type="checkbox"/>
	Nombre	varchar(55)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Figura 41: Estructura de la tabla estado\_cant  
Fuente: Elaboración propia

**Tabla: Turno**

TURNO			
	Column Name	Data Type	Allow Nulls
🔑	Codigo_Turno	int	<input type="checkbox"/>
	Turno	char(6)	<input type="checkbox"/>
			<input type="checkbox"/>

Figura 42: Estructura de la tabla turno  
Fuente: Elaboración propia

**Tabla: Asistencia**

ASISTENCIA			
	Column Name	Data Type	Allow Nulls
🔑	Codigo_Asistencia	int	<input type="checkbox"/>
	Hora_Ingreso	varchar(15)	<input type="checkbox"/>
	Fecha_Ingreso	varchar(15)	<input type="checkbox"/>
	Observaciones	varchar(15)	<input type="checkbox"/>
	Codigo_Personal	char(8)	<input type="checkbox"/>
	Codigo_Turno	int	<input type="checkbox"/>
			<input type="checkbox"/>

Figura 43: Estructura de la tabla asistencia  
Fuente: Elaboración propia



**Tabla: Incidencias**

INCIDENCIAS			
	Column Name	Data Type	Allow Nulls
🔑	Codigo_Incidencias	char(30)	<input type="checkbox"/>
	Descripcion	varchar(50)	<input type="checkbox"/>
	Codigo_Asiencia	int	<input type="checkbox"/>
	Fecha	date	<input checked="" type="checkbox"/>
	Cod_Incid	int	<input type="checkbox"/>
			<input type="checkbox"/>

Figura 44: Estructura de la tabla incidencias  
Fuente: Elaboración propia

**Tabla: Detalle\_Incidencia**

DETALLE_INCIDENCIA			
	Column Name	Data Type	Allow Nulls
🔑	Cod_Incid	int	<input type="checkbox"/>
	Descripcion	varchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Figura 45: Estructura de la tabla detalle\_incidencia  
Fuente: Elaboración propia

**Tabla: Distribución**

DISTRIBUCION			
	Column Name	Data Type	Allow Nulls
🔑	Codigo_Distribucion	int	<input type="checkbox"/>
	Fecha_Inicio	date	<input type="checkbox"/>
	Hora_Inicio	char(15)	<input type="checkbox"/>
	Codigo_Personal	char(8)	<input type="checkbox"/>
	Codigo_Puesto	int	<input type="checkbox"/>
	Codigo_Can	int	<input type="checkbox"/>
			<input type="checkbox"/>

Figura 46: Estructura de la tabla distribución  
Fuente: Elaboración propia

### 3.3.5 Codificación del sistema

#### Creación de la base de datos

```

use master
CREATE DATABASE Security_Dogs_Ormeño
GO
USE Security_Dogs_Ormeño

CREATE TABLE USUARIO
(
Codigo_Usuario int identity primary key,
Usuario varchar(50) not null,
Contraseña varchar(50) not null
)

CREATE TABLE PERSONAL(
Codigo_Personal char(8) NOT NULL PRIMARY KEY,
Nombres VARCHAR(25)NOT NULL,
Apellidos VARCHAR(25)NOT NULL,
Telefono VARCHAR(25)NOT NULL,
Direccion VARCHAR (25)NOT NULL,
Fecha_Ingreso VARCHAR (15) NOT NULL,
foto varchar(max) not null,
nombre_f varchar(20)
)

CREATE TABLE PUESTO
(
Codigo_Puesto int identity NOT NULL PRIMARY KEY,
Nombre VARCHAR (25) NOT NULL,
Ubicacion VARCHAR (25) NOT NULL,
)

CREATE TABLE CAN
(
Codigo_Can int identity NOT NULL PRIMARY KEY,
Nombre VARCHAR (25) NOT NULL,
Color VARCHAR (25) NOT NULL,
Raza VARCHAR (25) NOT NULL,
Fecha_Nacimiento DATE NOT NULL,
foto varchar(max) not null,
cod_deta int FOREIGN KEY REFERENCES estado_can (cod_deta)
)

CREATE TABLE DETALLE_CAN
(
Cod_Can int identity PRIMARY KEY,
Codigo_Can int FOREIGN KEY REFERENCES CAN (Codigo_Can),
cod_tipo int FOREIGN KEY REFERENCES Tipo_can (cod_tipo),

```

```
Fecha date
)
```

```
CREATE TABLE TIPO_CAN
(
cod_tipo int identity PRIMARY KEY,
Nombre VARCHAR (55)
)
insert into Tipo_can values ('Veterinario')
insert into Tipo_can values ('Baño')
```

```
CREATE TABLE ESTADO_CAN
(
cod_deta int identity(1,1) PRIMARY KEY,
Nombre VARCHAR (55)
)
```

```
CREATE TABLE TURNO
(
Codigo_Turno int identity NOT NULL PRIMARY KEY,
Turno CHAR (6)NOT NULL,
)
```

```
CREATE TABLE ASISTENCIA
(
Codigo_Asistencia int identity NOT NULL PRIMARY KEY,
Hora_Ingreso VARCHAR (15) NOT NULL,
Fecha_Ingreso VARCHAR (15) NOT NULL,
Observaciones VARCHAR (15) NOT NULL,
Codigo_Personal char(8) NOT NULL FOREIGN KEY REFERENCES
PERSONAL (Codigo_Personal)on delete cascade,
Codigo_Turno int NOT NULL FOREIGN KEY REFERENCES TURNO
(Codigo_Turno)on delete cascade
)
```

```
CREATE TABLE INCIDENCIAS
(
Codigo_Incidencias CHAR (30) NOT NULL PRIMARY KEY,
Descripcion VARCHAR (50) NOT NULL,
Codigo_Asistencia int NOT NULL FOREIGN KEY REFERENCES
ASISTENCIA (Codigo_Asistencia)on delete cascade,
Fecha date,
Cod_Incid int NOT NULL FOREIGN KEY REFERENCES
DETALLE_INCIDENCIA (Cod_Incid )
)
```

```
CREATE TABLE DETALLE_INCIDENCIA
(
```

```
Cod_Incid int identity(1,1) PRIMARY KEY,
Descripcion VARCHAR (50)
)
```

```
CREATE TABLE DISTRIBUCION
(
Codigo_Distribucion INT IDENTITY PRIMARY KEY,
Fecha_Inicio DATE NOT NULL,
Hora_Inicio CHAR(15) NOT NULL,
Codigo_Personal char(8) NOT NULL FOREIGN KEY REFERENCES
PERSONAL (Codigo_Personal)on delete cascade,
Codigo_Puesto int NOT NULL FOREIGN KEY REFERENCES PUESTO
(Codigo_Puesto)on delete cascade,
Codigo_Can int NOT NULL FOREIGN KEY REFERENCES CAN
(Codigo_Can)on delete cascade
)
```

## Codificación del sistema

### Conexion.vb

```
Imports System.Data.SqlClient
Module conexion
    Dim cn As New SqlConnection("Data Source=DESKTOP-
9HGPO4E\SQLEXPRESS;Database=Security_Dogs_Ormeño;Integrated
Security=SSPI")
End Module
```

### Consultas.vb

```
Imports System.Data.SqlClient
Imports System.Data.SqlDbType
Imports System.Data.SqlClient.SqlParameter
Public Class consultas
    Dim cn As New SqlConnection("Data Source=DESKTOP-
9HGPO4E\SQLEXPRESS;Database=Security_Dogs_Ormeño;Integrated
Security=SSPI")
    Dim cmd As New SqlCommand

    Public Function buscarcan(ByRef dato As String, ByRef can As String, dgv As
DataGridView)
        cn.Open()
        Try
            Dim ds As New DataSet
            Dim dt As New DataTable
            ds.Clear()
            Dim cmd As New SqlCommand(dato, cn)
```

```

cmd.CommandType = CommandType.StoredProcedure
cmd.Parameters.Add(New SqlParameter("@buscar", can))
Dim da As New SqlDataAdapter(cmd)
da.Fill(dt)
dgv.DataSource = dt
cn.Close()
Catch ex As Exception
MsgBox(ex.Message)
End Try
End Function
End Class

```

### **Entrada.vb**

```

Public Class Entrada

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
Asistencia.txtnum.Text = "1"
Asistencia.Show()

End Sub

Private Sub Button2_Click(sender As Object, e As EventArgs) Handles
Button2.Click
FrmInicioSesion.Show()
End Sub

End Class

```

### **Asistencia.vb**

```

Imports System.Data.SqlClient
Public Class Asistencia
Dim dt As New DataTable
Dim ds As New DataSet
Dim da As New SqlDataAdapter
Dim cn As New SqlConnection("Data Source=DESKTOP-
9HGPO4E\SQLEXPRESS;Database=Security_Dogs_Ormeño;Integrated
Security=SSPI")

Public Function asistencia()
cn.Open()
Try

Dim ds As New DataSet
Dim dt As New DataTable
ds.Clear()

```

```

Dim cmd As New SqlCommand("mostrarpersonal", cn)
cmd.CommandType = CommandType.StoredProcedure

cmd.Parameters.Add(New SqlParameter("@fecha", Me.txtdate.Text))
Dim da As New SqlDataAdapter(cmd)
da.Fill(dt)
dgvasistencia.DataSource = dt
cmd.ExecuteNonQuery()

cn.Close()

Catch ex As Exception
    MsgBox(ex.Message)
End Try
End Function

Public Function limpia()
    txtid.Text = ""
    LblHora.Text = ""
    txtdate.Text = Now
    txtobservaciones.Text = ""
    txtidper.Text = ""
    txtidper.Enabled = True
    txtidper.Focus()
End Function

Public Function agregar()
    cn.Open()
    Try

        Dim ds As New DataSet
        Dim dt As New DataTable
        ds.Clear()
        Dim cmd As New SqlCommand("agrega_asistencia", cn)
        cmd.CommandType = CommandType.StoredProcedure

        cmd.Parameters.Add(New SqlParameter("@Hora_Ingreso",
Me.LblHora.Text))
        cmd.Parameters.Add(New SqlParameter("@Fecha_Ingreso",
Me.txtdate.Text))
        cmd.Parameters.Add(New SqlParameter("@Observaciones",
Me.txtobservaciones.Text))
        cmd.Parameters.Add(New SqlParameter("@Codigo_Personal",
Me.txtidper.Text))
        cmd.Parameters.Add(New SqlParameter("@Codigo_Turno",
Me.cbo1.Text))
        cmd.ExecuteNonQuery()
        cn.Close()
    
```

```

Catch ex As Exception
    MsgBox(ex.Message)
End Try
End Function

```

```

Public Function modifica()
    cn.Open()
    Try
        Dim t As Integer
        Dim ds As New DataSet
        Dim dt As New DataTable
        ds.Clear()
        Dim cmd As New SqlCommand("actualiza_asistencia", cn)
        cmd.CommandType = CommandType.StoredProcedure
        cmd.Parameters.Add(New SqlParameter("@Codigo_Assistencia",
Me.txtid.Text))
        cmd.Parameters.Add(New SqlParameter("@Hora_Ingreso",
Me.LblHora.Text))
        cmd.Parameters.Add(New SqlParameter("@Fecha_Ingreso",
Me.txtdate.Text))
        cmd.Parameters.Add(New SqlParameter("@Observaciones",
Me.txtobservaciones.Text))
        cmd.Parameters.Add(New SqlParameter("@Codigo_Personal",
Me.txtidper.Text))
        cmd.Parameters.Add(New SqlParameter("@Codigo_Turno",
Me.cbo1.Text))
        cmd.ExecuteNonQuery()
        cn.Close()
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
End Function

```

```

Public Function elimina()
    cn.Open()
    Try
        Dim t As Integer
        Dim ds As New DataSet
        Dim dt As New DataTable
        ds.Clear()
        Dim cmd As New SqlCommand("elimina_asistencia", cn)
        cmd.CommandType = CommandType.StoredProcedure
        cmd.Parameters.Add(New SqlParameter("@Codigo_Assistencia",
Me.txtid.Text))
        cmd.ExecuteNonQuery()
        cn.Close()
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
End Function

```

```

End Try
End Function

Public Function llenarcombobox()
    cn.Open()
    Try
        Dim da As New SqlDataAdapter("select * from TURNO", cn)
        Dim dt As New DataTable
        da.Fill(dt)
        cbo2.DataSource = dt
        cbo1.DataSource = dt
        cbo2.ValueMember = "Turno"
        cbo1.ValueMember = "Codigo_Turno"
        cn.Close()
    Catch ex As Exception

    Finally
        cn.Close()
    End Try
End Function
Sub ConfigurarDataGrid()
    Try
        dgvasistencia.RowsDefaultCellStyle.BackColor = Color.LightGray
        dgvasistencia.AlternatingRowsDefaultCellStyle.BackColor =
Color.LightSkyBlue
        Me.dgvasistencia.Columns(0).Width = 60
        Me.dgvasistencia.Columns(1).Width = 65
        Me.dgvasistencia.Columns(2).Width = 140
        Me.dgvasistencia.Columns(3).Width = 57
        Me.dgvasistencia.Columns(4).Width = 77
        Me.dgvasistencia.Columns(5).Width = 77
        Me.dgvasistencia.Columns(6).Width = 125
    Catch ex As Exception

    End Try

End Sub

Private Sub Asistencia_Load(sender As Object, e As EventArgs) Handles
 MyBase.Load

    txtnum.Visible = False
    txtid.Visible = False
    Label2.Visible = False

    Btn_Modificar.Enabled = False
    Btn_Eliminar.Enabled = False
    llenarcombobox()
    asistencia()
    ConfigurarDataGrid()

```



```
cbo1.Visible = False
```

```
End Sub
```

```
Private Sub dgvasistencia_CellClick(sender As Object, e As
DataGridViewCellEventArgs) Handles dgvasistencia.CellClick
    Try
        Btn_Modificar.Enabled = True
        txtidper.Enabled = False
        If dgvasistencia.Rows.Count > 0 Then
            Dim dgvFila As DataGridViewRow = Me.dgvasistencia.CurrentRow()
            Me.txtid.Text = dgvFila.Cells(0).Value

            Btn_Modificar.Enabled = True
            Btn_Eliminar.Enabled = True
            Btn_Registrar.Enabled = False
        Else
            MessageBox.Show("No ha seleccionado ningun elemento", "!!Error",
            MessageBoxButtons.OK, MessageBoxIcon.Error)
        End If
        Catch ex As Exception
        End Try
    End Sub
```

```
Private Sub Btn_Personal_Click(sender As Object, e As EventArgs)
    Personal.Show()
End Sub
```

```
Private Sub Btn_Turno_Click(sender As Object, e As EventArgs)
    Turno.Show()
End Sub
```

```
Private Sub Btn_Nuevo_Click(sender As Object, e As EventArgs) Handles
Btn_Nuevo.Click
    Dim result As String
    result = MessageBox.Show("¿Desea limpiar el formulario?", "Nuevo",
    MessageBoxButtons.OKCancel, MessageBoxIcon.Question)
    If result = DialogResult.OK Then
        limpia()
        txtid.Enabled = True
    End If
End Sub
```

```
Private Sub Btn_Registrar_Click(sender As Object, e As EventArgs) Handles
Btn_Registrar.Click
    If txtidper.Text = "" Then
        MessageBox.Show("Existe espacios en blanco verifique nuevamente",
        "Ormeño", MessageBoxButtons.OK, MessageBoxIcon.Error)
    Else
        Dim result As String
```

```

        result = MessageBox.Show("¿Desea agregar el dato?", "Agregando",
        MessageBoxButtons.OKCancel, MessageBoxIcon.Question)
        If result = DialogResult.OK Then
            agregar()
            asistencia()
            limpia()
        End If
    End If
End Sub

Private Sub Btn_Modificar_Click(sender As Object, e As EventArgs)
    If txtid.Text = "" Or txtobservaciones.Text = "" Or txtidper.Text = "" Then
        MessageBox.Show("Existe espacios en blanco verifique nuevamente",
        "Ormeño", MessageBoxButtons.OK, MessageBoxIcon.Error)
    Else

        Dim result As String
        result = MessageBox.Show("¿Desea modificar el dato?", "Modificando",
        MessageBoxButtons.OKCancel, MessageBoxIcon.Question)
        If result = DialogResult.OK Then
            modifica()
            asistencia()
            limpia()
        End If
    End If
End Sub

Private Sub Btn_Eliminar_Click(sender As Object, e As EventArgs)
    If txtid.Text = "" Or txtobservaciones.Text = "" Or txtidper.Text = "" Then
        MessageBox.Show("Existe espacios en blanco verifique nuevamente",
        "Ormeño", MessageBoxButtons.OK, MessageBoxIcon.Error)
    Else

        Dim result As String
        result = MessageBox.Show("¿Desea eliminar el dato?", "Eliminando",
        MessageBoxButtons.OKCancel, MessageBoxIcon.Question)
        If result = DialogResult.OK Then
            elimina()
            asistencia()
            limpia()
        End If
    End If
End Sub

Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles
Timer1.Tick
    LblHora.Text = TimeOfDay

End Sub

```

```

Private Sub txtnum_TextChanged(sender As Object, e As EventArgs) Handles
txtnum.TextChanged
    If txtnum.Text = "1" Then
        Btn_Modificar.Enabled = False
        Btn_Eliminar.Enabled = False
        txtdate.Enabled = False
    Else
        'Btn_Modificar.Enabled = True
        'Btn_Eliminar.Enabled = True
        'txtdate.Enabled = True
    End If
End Sub

```

```

Private Sub Btn_Modificar_Click_1(sender As Object, e As EventArgs)
Handles Btn_Modificar.Click
    If txtid.Text = "" Or txtidper.Text = "" Then
        MessageBox.Show("Existe espacios en blanco verifique nuevamente",
"Ormeño", MessageBoxButtons.OK, MessageBoxIcon.Error)
    Else
        Dim result As String
        result = MessageBox.Show("¿Desea agregar el dato?", "Agregando",
MessageBoxButtons.OKCancel, MessageBoxIcon.Question)
        If result = DialogResult.OK Then
            modifica()
            asistencia()
            limpia()
            Btn_Modificar.Enabled = False
            Btn_Eliminar.Enabled = False
            Btn_Registrar.Enabled = True
            txtidper.Enabled = True
        End If
    End If
End Sub

```

```

Private Sub txtid_TextChanged(sender As Object, e As EventArgs) Handles
txtid.TextChanged
    datosasistencia()
End Sub

```

```

Private Sub Btn_Eliminar_Click_1(sender As Object, e As EventArgs) Handles
Btn_Eliminar.Click
    If txtid.Text = "" Or txtidper.Text = "" Then
        MessageBox.Show("Existe espacios en blanco verifique nuevamente",
"Ormeño", MessageBoxButtons.OK, MessageBoxIcon.Error)
    Else

        Dim result As String
        result = MessageBox.Show("¿Desea eliminar los datos ?", "Eliminando",
MessageBoxButtons.OKCancel, MessageBoxIcon.Question)
    End If
End Sub

```

```

    If result = DialogResult.OK Then
        elimina()
        asistencia()
        limpia()
        Btn_Modificar.Enabled = False
        Btn_Eliminar.Enabled = False
        Btn_Registrar.Enabled = True
    End If
End If
End Sub

```

```

Private Sub txtidper_TextChanged(sender As Object, e As EventArgs) Handles
txtidper.TextChanged
    Try
        Dim cm As New SqlCommand
        cn.Open()
        cm = New SqlCommand("select nombres + ' ' + apellidos as Nombre from
PERSONAL where Codigo_Personal= " + txtidper.Text + "'", cn)
        cm.CommandType = CommandType.Text
        dt = New DataTable
        da = New SqlDataAdapter(cm)
        da.Fill(dt)
        txtnombre.Text = dt.Rows(0)(0).ToString

        cn.Close()
    Catch ex As Exception
        txtnombre.Text = ""
    Finally
        cn.Close()
    End Try
End Sub

```

```

Private Sub txtdate_ValueChanged(sender As Object, e As EventArgs) Handles
txtdate.ValueChanged
    asistencia()
End Sub

```

```

Private Sub txthora_TextChanged(sender As Object, e As EventArgs) Handles
txthora.TextChanged
    Try
        If txthora.Text < 19 And txthora.Text >= 6 Then

            Btn_Eliminar.Enabled = True
            Btn_Modificar.Enabled = True
            Btn_Nuevo.Enabled = True
            Btn_Registrar.Enabled = True
        Else

```

```

        Btn_Eliminar.Enabled = False
        Btn_Modificar.Enabled = False
        Btn_Nuevo.Enabled = False
        Btn_Registrar.Enabled = False
    End If
Catch ex As Exception

End Try

End Sub
End Class

```

### **Login.vb**

```

Imports System.Data.SqlClient
Public Class FrmInicioSesion
Dim cn As New SqlConnection("Data Source=DESKTOP-
9HGPO4E\SQLEXPRESS;Database=Security_Dogs_Ormeño;Integrated
Security=SSPI")
    Dim intentos As Integer

    Dim da As SqlDataAdapter
    Public Function AccesoUsuarios()
        Dim dt As New DataTable
        Dim ds As New DataSet
        ds.Tables.Add(dt)
        cn.Open()
        Dim da As New SqlDataAdapter("select * from USUARIO", cn)
        da.Fill(dt)
        For Each datarow In dt.Rows
            If txtuser.Text = datarow.item(1) And txtpass.Text = datarow(2) Then
                Return True
            End If
        Next
        cn.Close()
        Return False
    End Function
Private Sub Btn_entrar_Click(sender As Object, e As EventArgs) Handles
Btn_entrar.Click
    Try
        If AccesoUsuarios() = True Then
            MessageBox.Show("Bienvenido al Sistema " + txtuser.Text, "Usuario",
MessageBoxButtons.OK, MessageBoxIcon.Information)
            txtuser.Text = ""
            txtpass.Text = ""
            Principal.Show()
        Else
            intentos = intentos + 1

```

```

        MessageBox.Show("Por favor ingrese su usuario y su contraseña",
"Security Dogs ,Ormeño", MessageBoxButtons.OK,
MessageBoxIcon.Information)
        txtuser.Text = ""
        txtpass.Text = ""
        If intentos = 3 Then
            MsgBox("Has agotado el numero de intentos, por favor espera 10
segundos y vuelve a intentarlo", MsgBoxStyle.Critical, "Intentos Agotados")
            intentos = 0
            Btn_entrar.Enabled = False
            Timer1.Start()
            Me.Close()
        End If
    End If
    Catch ex As Exception
    End Try
End Sub

```

```

Private Sub CheckBox1_CheckedChanged(sender As Object, e As EventArgs)
Handles CheckBox1.CheckedChanged
    If CheckBox1.Checked = True Then
        txtpass.PasswordChar = ""
    Else
        txtpass.PasswordChar = "*"
    End If
End Sub

```

```

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Try
        Dim result As String
        result = MessageBox.Show("¿Desea cerrar el Login?", "Ormeño",
MessageBoxButtons.OKCancel, MessageBoxIcon.Question)
        If result = DialogResult.OK Then
            Me.Close()
        End If
    Catch ex As Exception
    End Try
End Sub

```

```

Private Sub FrmInicioSesion_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
    intentos = 0
End Sub

```

```

Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles
Timer1.Tick
    Btn_entrar.Enabled = True
    Timer1.Stop()
End Sub

```

End Class

### **Principal.vb**

Public Class Principal

Private Sub PersonalToolStripMenuItem\_Click(sender As Object, e As EventArgs) Handles PersonalToolStripMenuItem.Click

Personal.MdiParent = Me

Personal.Show()

End Sub

Private Sub CanToolStripMenuItem\_Click(sender As Object, e As EventArgs) Handles CanToolStripMenuItem.Click

Can.MdiParent = Me

Can.Show()

End Sub

Private Sub PuestoToolStripMenuItem\_Click(sender As Object, e As EventArgs) Handles PuestoToolStripMenuItem.Click

Puesto.MdiParent = Me

Puesto.Show()

End Sub

Private Sub AsistenciaToolStripMenuItem\_Click(sender As Object, e As EventArgs) Handles AsistenciaToolStripMenuItem.Click

Asistencia.MdiParent = Me

Asistencia.Show()

End Sub

Private Sub TurnoToolStripMenuItem\_Click(sender As Object, e As EventArgs) Handles TurnoToolStripMenuItem.Click

Turno.MdiParent = Me

Turno.Show()

End Sub

Private Sub DistribuciónToolStripMenuItem\_Click(sender As Object, e As EventArgs) Handles DistribuciónToolStripMenuItem.Click

Distribución.MdiParent = Me

Distribución.Show()

End Sub

Private Sub SituaciónToolStripMenuItem\_Click(sender As Object, e As EventArgs) Handles SituaciónToolStripMenuItem.Click

Incidencias.MdiParent = Me

Incidencias.Show()

End Sub

```
Private Sub AcercaDeToolStripMenuItem_Click(sender As Object, e As
EventArgs) Handles AcercaDeToolStripMenuItem.Click
    Acerca_De.MdiParent = Me
    Acerca_De.Show()
End Sub
```

```
Private Sub ToolStripButton1_Click(sender As Object, e As EventArgs)
Handles ToolStripButton1.Click
    Personal.MdiParent = Me
    Personal.Show()
End Sub
```

```
Private Sub ToolStripButton2_Click(sender As Object, e As EventArgs)
Handles ToolStripButton2.Click
    Can.MdiParent = Me
    Can.Show()
End Sub
```

```
Private Sub ToolStripButton3_Click(sender As Object, e As EventArgs)
Handles ToolStripButton3.Click
    Puesto.MdiParent = Me
    Puesto.Show()
End Sub
```

```
Private Sub ToolStripButton4_Click(sender As Object, e As EventArgs)
Handles ToolStripButton4.Click
    Asistencia.MdiParent = Me
    Asistencia.Show()
End Sub
```

```
Private Sub ToolStripButton5_Click(sender As Object, e As EventArgs)
Handles ToolStripButton5.Click
    Turno.MdiParent = Me
    Turno.Show()
End Sub
```

```
Private Sub ToolStripButton6_Click(sender As Object, e As EventArgs)
Handles ToolStripButton6.Click
    Distribución.MdiParent = Me
    Distribución.Show()
End Sub
```

```
Private Sub ToolStripButton7_Click(sender As Object, e As EventArgs)
Handles ToolStripButton7.Click
    Incidencias.MdiParent = Me
    Incidencias.Show()
End Sub
```

```
End Class
```



**Personal.vb**

```

Imports System.Data.SqlClient
Public Class Personal

    Dim cn As New SqlConnection("Data Source=DESKTOP-
9HGPO4E\SQLEXPRESS;Database=Security_Dogs_Ormeño;Integrated
Security=SSPI")

    'Funcion Nuevo Personal
    Sub personal()
        Try
            Dim dt As New DataTable
            Dim da As New SqlDataAdapter("selectCodigo_Personal as DNI ,
Nombres,Apellidos,Telefono,Direccion,nombre_f AS
NomFoto,foto,Fecha_Ingreso as Ingreso from PERSONAL", cn)
            da.Fill(dt)
            dgvpersonal.DataSource = dt
            cn.Close()
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try
    End Sub

    'Funcionn Limpiar Personal
    Public Function limpiar()
        txtid.Text = ""
        txtnombre.Text = ""
        txtapellido.Text = ""
        txttelefono.Text = ""
        txtdireccion.Text = ""
        txtdate.Text = Now
        PictureBox1.Image = Nothing
        txtnombrefoto.Text = ""
        txtid.Focus()

    End Function

    'Funcion Agregar Personal
    Public Function agregar(ByRef image As String)
        cn.Open()
        Try

            Dim ds As New DataSet
            Dim dt As New DataTable
            ds.Clear()
            Dim cmd As New SqlCommand("personal", cn)
            cmd.CommandType = CommandType.StoredProcedure
            cmd.Parameters.Add(New SqlParameter("@dni", Me.txtid.Text))

```

```

        cmd.Parameters.Add(New SqlParameter("@Nombres",
Me.txtnumero.Text))
        cmd.Parameters.Add(New SqlParameter("@Apellidos",
Me.txtapellido.Text))
        cmd.Parameters.Add(New SqlParameter("@Telefono",
Me.txttelefono.Text))
        cmd.Parameters.Add(New SqlParameter("@Direccion",
Me.txtdireccion.Text))
        cmd.Parameters.Add(New SqlParameter("@Fecha_Ingreso",
Me.txtdate.Text))
        cmd.Parameters.Add(New SqlParameter("@foto", image))
        cmd.Parameters.Add(New SqlParameter("@nombre_f",
Me.txtnombrefoto.Text))
        cmd.ExecuteNonQuery()
        cn.Close()
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
End Function

'Funcion Actualizar Personal
Public Function actualizar()
    cn.Open()
    Try

        Dim ds As New DataSet
        Dim dt As New DataTable
        ds.Clear()
        Dim cmd As New SqlCommand("actualiza_personal", cn)
        cmd.CommandType = CommandType.StoredProcedure
        cmd.Parameters.Add(New SqlParameter("@Codigo_Personal",
Me.txtid.Text))
        cmd.Parameters.Add(New SqlParameter("@Nombres",
Me.txtnumero.Text))
        cmd.Parameters.Add(New SqlParameter("@Apellidos",
Me.txtapellido.Text))
        cmd.Parameters.Add(New SqlParameter("@Telefono",
Me.txttelefono.Text))
        cmd.Parameters.Add(New SqlParameter("@Direccion",
Me.txtdireccion.Text))
        cmd.Parameters.Add(New SqlParameter("@Fecha_Ingreso",
Me.txtdate.Text))
        cmd.Parameters.Add(New SqlParameter("@foto", Me.IMAGEN))
        cmd.Parameters.Add(New SqlParameter("@nomf",
Me.txtnombrefoto.Text))
        cmd.ExecuteNonQuery()
        cn.Close()
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try

```

End Function

'Funcion eliminar Personal

Public Function eliminar()

    cn.Open()

    Try

        Dim ds As New DataSet

        Dim dt As New DataTable

        ds.Clear()

        Dim cmd As New SqlCommand("elimina\_personal", cn)

        cmd.CommandType = CommandType.StoredProcedure

        cmd.Parameters.Add(New SqlParameter("@Codigo\_Personal",

Me.txtid.Text))

        cmd.ExecuteNonQuery()

        cn.Close()

    Catch ex As Exception

        MsgBox(ex.Message)

    End Try

End Function

Sub ConfigurarDataGrid()

    dgvpersonal.RowsDefaultCellStyle.BackColor = Color.LightGray

    dgvpersonal.AlternatingRowsDefaultCellStyle.BackColor =

Color.LightSkyBlue

End Sub

Private Sub Personal\_Load(sender As Object, e As EventArgs) Handles  
MyBase.Load

    personal()

    ConfigurarDataGrid()

    Btn\_Modificar.Enabled = False

    Btn\_Eliminar.Enabled = False

End Sub

Private Sub dgvpersonal\_CellClick(sender As Object, e As  
DataGridViewCellEventArgs) Handles dgvpersonal.CellClick

    Try

        Btn\_Registrar.Enabled = False

        txtid.Enabled = False

        If dgvpersonal.Rows.Count > 0 Then

            Dim dgvFila As DataGridViewRow = Me.dgvpersonal.CurrentRow()

            Me.txtid.Text = dgvFila.Cells(0).Value

            Me.txtnombre.Text = dgvFila.Cells(1).Value

            Me.txtapellido.Text = dgvFila.Cells(2).Value

            Me.txttelefono.Text = dgvFila.Cells(3).Value

            Me.txtdireccion.Text = dgvFila.Cells(4).Value

            Me.txtdate.Text = dgvFila.Cells(7).Value

            Me.IMAGEN = dgvFila.Cells(6).Value

            PictureBox1.Load(IMAGEN)

```

        Me.txtnombrefoto.Text = dgvFila.Cells(5).Value
        Btn_Modificar.Enabled = True
        Btn_Eliminar.Enabled = True

    Else
        MessageBox.Show("No ha seleccionado ningun elemento", "!!Error",
        MessageBoxButtons.OK, MessageBoxIcon.Error)
    End If
    Catch ex As Exception

    End Try

End Sub

Private Sub dgvpersonal_CellDoubleClick(sender As Object, e As
DataGridViewCellEventArgs) Handles dgvpersonal.CellDoubleClick
    Try

        If txtid.Text = "" Or txtnombre.Text = "" Or txtapellido.Text = "" Or
        txttelefono.Text = "" Or txtdireccion.Text = "" Then
            MessageBox.Show("Existe espacios en blanco verifique nuevamente",
            "Ormeño", MessageBoxButtons.OK, MessageBoxIcon.Error)
        Else
            If dgvpersonal.Rows.Count > 0 Then
                Dim dgvFila As DataGridViewRow = Me.dgvpersonal.CurrentRow()
                Asistencia.txtidper.Text = dgvFila.Cells(0).Value

                Me.Close()

            Else
                MessageBox.Show("No ha seleccionado ningun elemento", "!!Error",
                MessageBoxButtons.OK, MessageBoxIcon.Error)
            End If
        End If
        Catch ex As Exception

        End Try
    End Sub

Private Sub Btn_Nuevo_Click(sender As Object, e As EventArgs) Handles
Btn_Nuevo.Click
    If txtid.Text = "" Or txtnombre.Text = "" Or txtapellido.Text = "" Or
    txttelefono.Text = "" Or txtdireccion.Text = "" Then
        MessageBox.Show("Existe espacios en blanco verifique nuevamente",
        "Ormeño", MessageBoxButtons.OK, MessageBoxIcon.Error)
    Else

        Dim result As String

```

```

        result = MessageBox.Show("¿Desea limpiar el formulario ?", "Nuevo",
        MessageBoxButtons.OKCancel, MessageBoxIcon.Question)
        If result = DialogResult.OK Then
            limpiar()
            txtid.Enabled = True
            Btn_Modificar.Enabled = False
            Btn_Eliminar.Enabled = False
        End If
    End If
End Sub

```

```

Private Sub Btn_Registrar_Click(sender As Object, e As EventArgs) Handles
Btn_Registrar.Click
    If txtnombre.Text = "" Or txtapellido.Text = "" Or txtnombrefoto.Text = ""
Then
        MessageBox.Show("Existe espacios en blanco verifique nuevamente",
        "Ormeño", MessageBoxButtons.OK, MessageBoxIcon.Error)
    Else

```

```

        Dim result As String
        result = MessageBox.Show("¿Desea agregar los datos ?", "Agregando",
        MessageBoxButtons.OKCancel, MessageBoxIcon.Question)
        If result = DialogResult.OK Then
            If IMAGEN = "" Then
                agregar("C:\")
                personal()
                limpiar()
            Else
                agregar(IMAGEN)
                personal()
                limpiar()
            End If
        End If
    End If
End Sub

```

```

Private Sub Btn_Modificar_Click(sender As Object, e As EventArgs) Handles
Btn_Modificar.Click
    If txtid.Text = "" Or txtnombre.Text = "" Or txtapellido.Text = "" Or
txttelefono.Text = "" Or txtdireccion.Text = "" Then
        MessageBox.Show("Existe espacios en blanco verifique nuevamente",
        "Ormeño", MessageBoxButtons.OK, MessageBoxIcon.Error)
    Else
        Dim result As String
        result = MessageBox.Show("¿Desea modificar los datos ?",
        "Modificando", MessageBoxButtons.OKCancel, MessageBoxIcon.Question)
        If result = DialogResult.OK Then
            actualizar()
            personal()

```

```

        limpiar()
        Btn_Registrar.Enabled = True
        txtid.Enabled = True
        Btn_Modificar.Enabled = False
        Btn_Eliminar.Enabled = False
    End If
End If
End Sub

Private Sub Btn_Eliminar_Click(sender As Object, e As EventArgs) Handles
Btn_Eliminar.Click
    If txtid.Text = "" Or txtnombre.Text = "" Or txtapellido.Text = "" Or
txttelefono.Text = "" Or txtdireccion.Text = "" Then
        MessageBox.Show("Existe espacios en blanco verifique nuevamente",
"Ormeño", MessageBoxButtons.OK, MessageBoxIcon.Error)
    Else

        Dim result As String
        result = MessageBox.Show("¿Desea eliminar los datos ?", "Eliminando",
MessageBoxButtons.OKCancel, MessageBoxIcon.Question)
        If result = DialogResult.OK Then
            eliminar()
            personal()
            limpiar()
        End If
    End If
End Sub

Private Sub txtid_TextChanged(sender As Object, e As EventArgs) Handles
txtid.TextChanged

End Sub
Dim IMAGEN As String
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Try
        Me.OpenFileDialog1.ShowDialog()
        If Me.OpenFileDialog1.FileName <> "" Then

            IMAGEN = OpenFileDialog1.FileName
            Dim largo As Integer = IMAGEN.Length
            Dim imagen2 As String
            imagen2 = CStr(Microsoft.VisualBasic.Mid(RTrim(IMAGEN), largo -
2, largo))
            If imagen2 <> "gif" And imagen2 <> "bmp" And imagen2 <> "jpg"
And imagen2 <> "jpeg" And imagen2 <> "GIF" And imagen2 <> "BMP" And
imagen2 <> "JPG" And imagen2 <> "JPEG" And imagen2 <> "png" Then
                imagen2 = CStr(Microsoft.VisualBasic.Mid(RTrim(IMAGEN), largo
- 3, largo))
            End If
        End If
    Catch ex As Exception
    End Try
End Sub

```

```

        If imagen2 <> "jpeg" And imagen2 <> "JPEG" And imagen2 <>
"log1" Then
            MsgBox("Formato no valido") : Exit Sub
            If imagen2 <> "log1" Then Exit Sub
            End If
            PictureBox1.Load(IMAGEN)

        End If
    End If
    Catch ex As Exception

    End Try
    PictureBox1.Load(IMAGEN)
End Sub

End Class

```

### **Distribucion.vb**

```

Imports System.Data.SqlClient
Public Class Distribución
    Dim dt As New DataTable
    Dim ds As New DataSet
    Dim da As New SqlDataAdapter
    Dim cn As New SqlConnection("Data Source=DESKTOP-
9HGPO4E\SQLEXPRESS;Database=Security_Dogs_Ormeño;Integrated
Security=SSPI")
    Public Function DISTRIBUCION()
        cn.Open()
        Try

            Dim ds As New DataSet
            Dim dt As New DataTable
            ds.Clear()
            Dim cmd As New SqlCommand("MOSTRAR_distribucion", cn)
            cmd.CommandType = CommandType.StoredProcedure

            cmd.Parameters.Add(New SqlParameter("@fecha", Me.txtinicio.Text))
            Dim da As New SqlDataAdapter(cmd)
            da.Fill(dt)
            dgvdistribucion.DataSource = dt
            cmd.ExecuteNonQuery()

            cn.Close()

        Catch ex As Exception
            MsgBox(ex.Message)
        End Try
    End Function
    Public Function llenarcombobox()

```

```

cn.Open()
Try
  Dim da As New SqlDataAdapter("select * from PUESTO", cn)
  Dim dt As New DataTable
  da.Fill(dt)
  cbo2.DataSource = dt
  txtidpuesto.DataSource = dt
  cbo2.ValueMember = "Nombre"
  txtidpuesto.ValueMember = "Codigo_Puesto"
  cn.Close()
Catch ex As Exception

Finally
  cn.Close()
End Try
End Function
Public Function llenarcombobox2()
  cn.Open()
  Try
    Dim da As New SqlDataAdapter("select Codigo_Can,Nombre from
CAN", cn)
    Dim dt As New DataTable
    da.Fill(dt)
    cbocan1.DataSource = dt
    txtcan.DataSource = dt
    cbocan1.ValueMember = "Nombre"
    txtcan.ValueMember = "Codigo_Can"
    cn.Close()
  Catch ex As Exception

  Finally
    cn.Close()
  End Try
End Function
Sub datosdist()
  Try
    Dim cm As New SqlCommand

    cn.Open()
    cm = New SqlCommand("select nombres,apellidos from PERSONAL
where Codigo_Personal= '" + txtidpersonal.Text + "'", cn)
    cm.CommandType = CommandType.Text
    dt = New DataTable
    da = New SqlDataAdapter(cm)
    da.Fill(dt)
    txtnombre.Text = dt.Rows(0)(0).ToString
    txtapellido.Text = dt.Rows(0)(1).ToString

    cn.Close()
  Catch ex As Exception

```



```

        txtnombre.Text = ""
        txtapellido.Text = ""
    Finally
        cn.Close()
    End Try
End Sub
Sub personal()
    Try
        Dim dt As New DataTable
        Dim da As New SqlDataAdapter("select Codigo_Personal as DNI ,
Nombres , Apellidos from PERSONAL", cn)
        da.Fill(dt)
        DataGridView1.DataSource = dt
        cn.Close()
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
End Sub

Sub puesto()
    Try
        Dim dt As New DataTable
        Dim da As New SqlDataAdapter("select Codigo_Puesto as Cod , Nombre
as Puesto from PUESTO", cn)
        da.Fill(dt)
        DataGridView2.DataSource = dt
        cn.Close()
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
End Sub

Sub can()
    Try
        Dim dt As New DataTable
        Dim da As New SqlDataAdapter("select Codigo_Can as Cod , Nombre
from CAN", cn)
        da.Fill(dt)
        DataGridView3.DataSource = dt
        cn.Close()
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
End Sub

Public Function limpia()
    txtid.Text = ""
    txtinicio.Text = Now
    txtidpersonal.Enabled = True
    txtidpersonal.Text = ""

```

```

txtnombre.Text = ""
txtapellido.Text = ""
txtbuscarcan.Text = ""
txtbuscarsepersonal.Text = ""
txtbuscarsepuesto.Text = ""
txtidpersonal.Focus()
End Function

```

```

Public Function agregar()
    cn.Open()
    Try

        Dim ds As New DataSet
        Dim dt As New DataTable
        ds.Clear()
        Dim cmd As New SqlCommand("agrega_distribucion", cn)
        cmd.CommandType = CommandType.StoredProcedure
        cmd.Parameters.Add(New SqlParameter("@Fecha_Inicio",
Me.txtinicio.Text))
        cmd.Parameters.Add(New SqlParameter("@Hora_Inicio",
Me.LblHoraInicio.Text))
        cmd.Parameters.Add(New SqlParameter("@Codigo_Personal",
Me.txtidpersonal.Text))
        cmd.Parameters.Add(New SqlParameter("@Codigo_Puesto",
Me.txtidpuesto.Text))
        cmd.Parameters.Add(New SqlParameter("@Codigo_Can",
Me.txtcan.Text))
        cmd.ExecuteNonQuery()
        cn.Close()
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
End Function

```

```

Public Function modifica()
    cn.Open()
    Try
        Dim t As Integer
        Dim ds As New DataSet
        Dim dt As New DataTable
        ds.Clear()
        Dim cmd As New SqlCommand("actualiza_distribucion", cn)
        cmd.CommandType = CommandType.StoredProcedure
        cmd.Parameters.Add(New SqlParameter("@Codigo_Distribucion",
Me.txtid.Text))
        cmd.Parameters.Add(New SqlParameter("@Fecha_Inicio",
Me.txtinicio.Text))

```

```

        cmd.Parameters.Add(New SqlParameter("@Codigo_Personal",
Me.txtidpersonal.Text))
        cmd.Parameters.Add(New SqlParameter("@Codigo_Puesto",
Me.txtidpuesto.Text))
        cmd.Parameters.Add(New SqlParameter("@Codigo_Can",
Me.txtcan.Text))
        cmd.ExecuteNonQuery()
        cn.Close()
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
End Function

Sub datosdis()
    Try
        Dim cm As New SqlCommand

        cn.Open()
        cm = New SqlCommand("select * from DISTRIBUCION where
Codigo_Distribucion=" + txtid.Text + "'", cn)
        cm.CommandType = CommandType.Text
        dt = New DataTable
        da = New SqlDataAdapter(cm)
        da.Fill(dt)
        txtidpersonal.Text = dt.Rows(0)(3).ToString
        txtidpuesto.Text = dt.Rows(0)(4).ToString
        txtcan.Text = dt.Rows(0)(5).ToString
        txtinicio.Text = dt.Rows(0)(1).ToString
        cn.Close()
    Catch ex As Exception

    Finally
        cn.Close()
    End Try
End Sub

Public Function elimina()
    cn.Open()
    Try
        Dim t As Integer
        Dim ds As New DataSet
        Dim dt As New DataTable
        ds.Clear()
        Dim cmd As New SqlCommand("elimina_distribucion", cn)
        cmd.CommandType = CommandType.StoredProcedure
        cmd.Parameters.Add(New SqlParameter("@Codigo_Distribucion",
Me.txtid.Text))
        cmd.ExecuteNonQuery()
        cn.Close()
    Catch ex As Exception

```

```

        MsgBox(ex.Message)
    End Try
End Function

Sub ConfigurarDataGrid1()
    DataGridView1.RowsDefaultCellStyle.BackColor = Color.LightGray
    DataGridView1.AlternatingRowsDefaultCellStyle.BackColor =
Color.LightSkyBlue
    Me.DataGridView1.Columns(0).Width = 65
    Me.DataGridView1.Columns(1).Width = 60
    Me.DataGridView1.Columns(2).Width = 80
End Sub

Sub ConfigurarDataGrid2()
    DataGridView2.RowsDefaultCellStyle.BackColor = Color.LightGray
    DataGridView2.AlternatingRowsDefaultCellStyle.BackColor =
Color.LightSkyBlue
    Me.DataGridView2.Columns(0).Width = 65
    Me.DataGridView2.Columns(1).Width = 65
    Me.DataGridView2.Columns(2).Width = 110
End Sub

Sub ConfigurarDataGrid3()
    Try
        DataGridView3.RowsDefaultCellStyle.BackColor = Color.LightGray
        DataGridView3.AlternatingRowsDefaultCellStyle.BackColor =
Color.LightSkyBlue
        Me.DataGridView3.Columns(0).Width = 65
        Me.DataGridView3.Columns(1).Width = 65
        Me.DataGridView3.Columns(2).Width = 110
    Catch ex As Exception

    End Try

End Sub

Sub ConfigurarDataGrid4()
    Try

        dgvdistribucion.RowsDefaultCellStyle.BackColor = Color.LightGray
        dgvdistribucion.AlternatingRowsDefaultCellStyle.BackColor =
Color.LightSkyBlue
        Me.dgvdistribucion.Columns(0).Width = 60
        Me.dgvdistribucion.Columns(1).Width = 80
        Me.dgvdistribucion.Columns(2).Width = 120
        Me.dgvdistribucion.Columns(3).Width = 160
        Me.dgvdistribucion.Columns(4).Width = 120
        Me.dgvdistribucion.Columns(5).Width = 160
        Me.dgvdistribucion.Columns(6).Width = 80
        Me.dgvdistribucion.Columns(7).Width = 75
    
```

```

    Me.dgvdistribucion.Columns(8).Width = 75
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try

```

```
End Sub
```

```
Private Sub Distribución_Load(sender As Object, e As EventArgs) Handles MyBase.Load
```

```

    txtid.Visible = False
    Label2.Visible = False
    txtidpuesto.Visible = False
    txtcan.Visible = False
    DISTRIBUCION()
    ConfigurarDataGrid4()
    personal()
    ConfigurarDataGrid1()

```

```

    puesto()
    ConfigurarDataGrid2()

```

```

    can()
    ConfigurarDataGrid3()
    llenarcombobox()
    llenarcombobox2()

```

```
End Sub
```

```
Private Sub Btn_Nuevo_Click(sender As Object, e As EventArgs) Handles Btn_Nuevo.Click
```

```

    Dim result As String
    result = MessageBox.Show("¿Desea limpiar el formulario?", "Nuevo",
    MessageBoxButtons.OKCancel, MessageBoxIcon.Question)
    If result = DialogResult.OK Then
        limpia()
    End If

```

```
End Sub
```

```
Private Sub Btn_Registrar_Click(sender As Object, e As EventArgs) Handles Btn_Registrar.Click
```

```

    If txtidpersonal.Text = "" Or txtidpuesto.Text = "" Or txtcan.Text = "" Then
        MessageBox.Show("Existe espacios en blanco verifique nuevamente",
    "Ormeño", MessageBoxButtons.OK, MessageBoxIcon.Error)
    Else
        Dim result As String

```

```

        result = MessageBox.Show("¿Desea agregar datos?", "Agregando",
        MessageBoxButtons.OKCancel, MessageBoxIcon.Question)
        If result = DialogResult.OK Then
            agregar()
            DISTRIBUCION()
            limpia()
            Btn_Modificar.Enabled = False
            Btn_Eliminar.Enabled = False
            Btn_Registrar.Enabled = True
        End If
    End If
End Sub

```

```

Private Sub Btn_Modificar_Click(sender As Object, e As EventArgs) Handles
Btn_Modificar.Click
    If txtid.Text = "" Or txtid.Text = "" Or txtidpersonal.Text = "" Or
txtidpuesto.Text = "" Or txtcan.Text = "" Then
        MessageBox.Show("Existe espacios en blanco verifique nuevamente",
"Ormeño", MessageBoxButtons.OK, MessageBoxIcon.Error)
    Else
        Dim result As String
        result = MessageBox.Show("¿Desea modificar datos?", "Modificando",
        MessageBoxButtons.OKCancel, MessageBoxIcon.Question)
        If result = DialogResult.OK Then
            modifica()
            'agregar()
            DISTRIBUCION()
            txtidpersonal.Enabled = True
            Btn_Modificar.Enabled = False
            Btn_Eliminar.Enabled = False
            Btn_Registrar.Enabled = True
            limpia()
        End If
    End If
End Sub

```

```

Private Sub Btn_Eliminar_Click(sender As Object, e As EventArgs) Handles
Btn_Eliminar.Click
    If txtid.Text = "" Or txtid.Text = "" Or txtidpersonal.Text = "" Or
txtidpuesto.Text = "" Or txtcan.Text = "" Then
        MessageBox.Show("Existe espacios en blanco verifique nuevamente",
"Ormeño", MessageBoxButtons.OK, MessageBoxIcon.Error)
    Else
        Dim result As String
        result = MessageBox.Show("¿Desea eliminar el dato?", "Eliminando",
        MessageBoxButtons.OKCancel, MessageBoxIcon.Question)
        If result = DialogResult.OK Then
            elimina()
            DISTRIBUCION()
            limpia()
        End If
    End If
End Sub

```

```

        Btn_Modificar.Enabled = False
        Btn_Eliminar.Enabled = False
        Btn_Registrar.Enabled = True
    End If
End If
End Sub

```

```

Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles
Timer1.Tick
    LblHoraInicio.Text = TimeOfDay
End Sub

```

```

Private Sub txtidpersonal_TextChanged(sender As Object, e As EventArgs)
Handles txtidpersonal.TextChanged
    Try
        datosdist()
    Catch ex As Exception

    End Try

End Sub

```

```

Private Sub dgvdistribucion_CellClick(sender As Object, e As
DataGridViewCellEventArgs) Handles dgvdistribucion.CellClick
    Try
        Btn_Modificar.Enabled = True

        If dgvdistribucion.Rows.Count > 0 Then
            Dim dgvFila As DataGridViewRow =
Me.dgvdistribucion.CurrentRow()
            Me.txtid.Text = dgvFila.Cells(0).Value
            Me.txtidpersonal.Text = dgvFila.Cells(1).Value
            Btn_Modificar.Enabled = True
            Btn_Eliminar.Enabled = True
            Btn_Registrar.Enabled = False
            txtidpersonal.Enabled = False
        Else
            MessageBox.Show("No ha seleccionado ningun elemento", "!!Error",
MessageBoxButtons.OK, MessageBoxIcon.Error)
        End If
    Catch ex As Exception
    End Try
End Sub

```

```

Private Sub txtid_TextChanged(sender As Object, e As EventArgs) Handles
txtid.TextChanged
    datosdis()
End Sub

```

```

Private Sub DataGridView3_CellClick(sender As Object, e As
DataGridViewCellEventArgs) Handles DataGridView3.CellClick
    Try
        If DataGridView3.Rows.Count > 0 Then
            Dim dgvFila As DataGridViewRow =
Me.DataGridView3.CurrentRow()
            Me.txtcan.Text = dgvFila.Cells(1).Value
        Else
            MessageBox.Show("No ha seleccionado ningun elemento", "!!Error",
MessageBoxButtons.OK, MessageBoxIcon.Error)
        End If
    Catch ex As Exception
    End Try
End Sub

```

```

Private Sub TextBox3_TextChanged(sender As Object, e As EventArgs)
Handles txtbuscarcan.TextChanged
    Try
        Dim consulta As New consultas
        consulta.buscarcan("buscarcan", txtbuscarcan.Text, DataGridView3)
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
End Sub

```

```

Private Sub TextBox2_TextChanged(sender As Object, e As EventArgs)
Handles txtbuscarpuesto.TextChanged
    Try
        Dim consulta As New consultas
        consulta.buscarcan("buscarpuesto", txtbuscarpuesto.Text, DataGridView2)
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
End Sub

```

```

Private Sub TextBox1_TextChanged(sender As Object, e As EventArgs)
Handles txtbuscarpersonal.TextChanged
    Try
        Dim consulta As New consultas
        consulta.buscarcan("buscarpersonal", txtbuscarpersonal.Text,
DataGridView1)
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
End Sub

```

```

Private Sub DataGridView1_CellClick(sender As Object, e As
DataGridViewCellEventArgs) Handles DataGridView1.CellClick
    Try

```



```

    If DataGridView1.Rows.Count > 0 Then
        Dim dgvFila As DataGridViewRow =
Me.DataGridView1.CurrentRow()
        Me.txtidpersonal.Text = dgvFila.Cells(1).Value
        Me.txtnombre.Text = dgvFila.Cells(2).Value
        txtapellido.Text = dgvFila.Cells(3).Value
    Else
        MessageBox.Show("No ha seleccionado ningun elemento", "!!Error",
MessageBoxButtons.OK, MessageBoxIcon.Error)
    End If
    Catch ex As Exception
    End Try
End Sub

Private Sub DataGridView2_CellClick(sender As Object, e As
DataGridViewCellEventArgs) Handles DataGridView2.CellClick
    Try
        If DataGridView2.Rows.Count > 0 Then
            Dim dgvFila As DataGridViewRow =
Me.DataGridView2.CurrentRow()
            Me.txtidpuesto.Text = dgvFila.Cells(1).Value
        Else
            MessageBox.Show("No ha seleccionado ningun elemento", "!!Error",
MessageBoxButtons.OK, MessageBoxIcon.Error)
        End If
        Catch ex As Exception
        End Try
    End Sub

Private Sub txtinicio_ValueChanged(sender As Object, e As EventArgs)
Handles txtinicio.ValueChanged
    Try
        DISTRIBUCION()
    Catch ex As Exception
    End Try
End Sub
End Class

```

### 3.3.6 Pruebas del sistema

En esta etapa se realizaron las pruebas al sistema con la finalidad de que el sistema presente buena performance y usabilidad. Las misma que se detallan a continuación:

- **Pruebas de funcionalidad**, estas pruebas se realizaron durante y al final del desarrollo con la finalidad de que se cumplan los requerimientos funcionales solicitados.

- **Pruebas de estrés**, estas pruebas se realizaron para verificar si el sistema se hacía lento con una determinada carga de datos.
- **Pruebas de usabilidad**, estas pruebas se realizaron con los usuarios de la empresa donde ellos interactuaron con el software y se les solicitó su opinión sobre el uso de este.

Al finalizar las respectivas pruebas realizadas al sistema de la empresa Ormeño se procedió a crear el instalador del software.

### 3.3.7 Implementación del sistema

Para la implementación y puesta en marcha del sistema se tuvo que generar en primer lugar el instalador del sistema, también se elaboró el manual del sistema y se capacitó a los usuarios que tendrán a cargo el manejo del sistema.

## 3.4 Actividades

El presente trabajo de aplicación profesional fue desarrollado utilizando las siguientes etapas de desarrollo:

- ❖ **Levantamiento de la información:** En esta etapa se realizaron entrevistas al gerente de la empresa Ormeño y al encargado del control de personal y del servicio de vigilancia canina con la finalidad de tener información de cómo se realizan los procesos actuales y que requerimientos necesitan.
- ❖ **Análisis de la información:** Después de haber realizado el levantamiento de la información, se realizaron diagramas casos de uso empleando la herramienta Rational Rose, donde se muestran cómo se realizan los procesos actuales y los procesos propuestos.
- ❖ **Diseño del sistema:** En esta etapa se diseñaron el formulario principal, el formulario de asistencia, el formulario de distribución de personal y canes, etc.
- ❖ **Codificación del sistema:** En esta etapa se codificó la base de datos, así como los módulos que forman parte del sistema.
- ❖ **Pruebas del sistema:** Después de haber desarrollado el sistema se realizaron pruebas para poder detectar algunas fallas las mismas que fueron levantadas en su momento.

## 3.5 Limitaciones

- Los autores residían en lugares distantes.

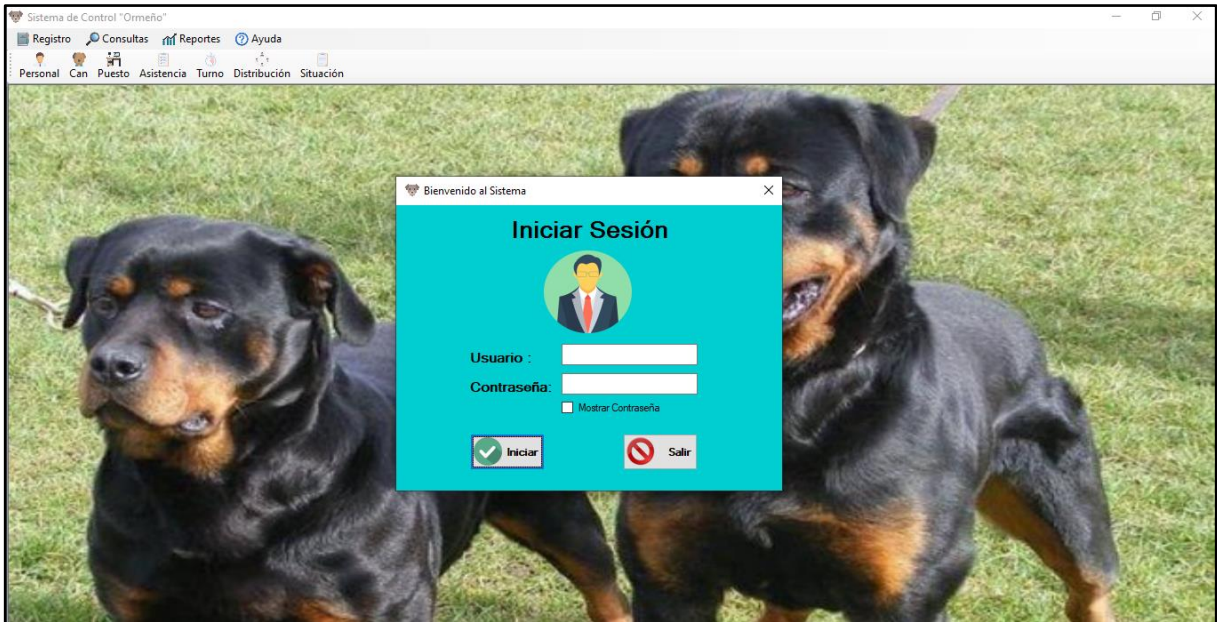
- Los autores tenían inconvenientes en las comunicaciones.
- Los autores trabajaban en horarios diferentes, lo que causaba problemas para reuniones de trabajo.
- Debido a la emergencia sanitaria del COVID-19, nos desatendimos del trabajo de aplicación, ya que teníamos que cuidar a nuestros familiares y a nosotros mismos.

## **CAPÍTULO IV**

### **RESULTADOS**

## Resultados

- ❖ Se desarrolló el sistema de servicio de control de asistencia de personal y vigilancia canina para la empresa Security Dog Ormeño.



- ❖ Se implementó la base de datos que permite administrar mejor la información de la empresa Security Dog Ormeño.
- ❖ Se logró controlar la asistencia del personal.
- ❖ Se logró realizar la distribución del personal y canes para el servicio de vigilancia canina.

## **CAPÍTULO V**

### **CONCLUSIONES Y RECOMENDACIONES**

## Conclusiones

- a) El sistema de escritorio cumple con todos los requerimientos solicitados por el usuario.
- b) Se logró identificar los procesos y registros de asistencia de personal.
- c) Se realizaron pruebas al sistema de escritorio.
- d) El lenguaje de modelado unificado permitió analizar claramente los procesos de registro de asistencia de personal y el servicio de vigilancia canina.
- e) La base de datos se desarrolló utilizando Microsoft SQL Server, la misma que sirvió de fuente de información de consultas para los diferentes procesos que realiza el sistema.
- f) Para el desarrollo del sistema se empleó el paquete informático Microsoft Visual Studio, empleando el lenguaje de programación Visual Basic .Net, con el que se logró codificar los requerimientos solicitados por la empresa Security Dog Ormeño.
- g) El desarrollo del presente trabajo de aplicación permitió enriquecer nuestros conocimientos en programación y base de datos.
- h) Las tecnologías de la información representan un gran apoyo para automatizar los diferentes procesos en las empresas.

### **Recomendaciones**

- a) Se recomienda capacitar al personal que tendrá a cargo el manejo del sistema, todo esto con la finalidad de no tener inconvenientes en el manejo.
- b) Se recomienda tener a la mano el manual del sistema para poder disipar alguna duda en cuanto al funcionamiento.
- c) Se recomienda realizar el mantenimiento y actualización al sistema en períodos semestrales o anuales, todo esto debido a la solicitud de nuevos requerimientos.
- d) Se recomienda realizar copias de seguridad tanto del sistema como de la base de datos.
- e) Se recomienda realizar cambios en las contraseñas de los usuarios por lo menos cada quince días.



## Referencias Bibliográficas

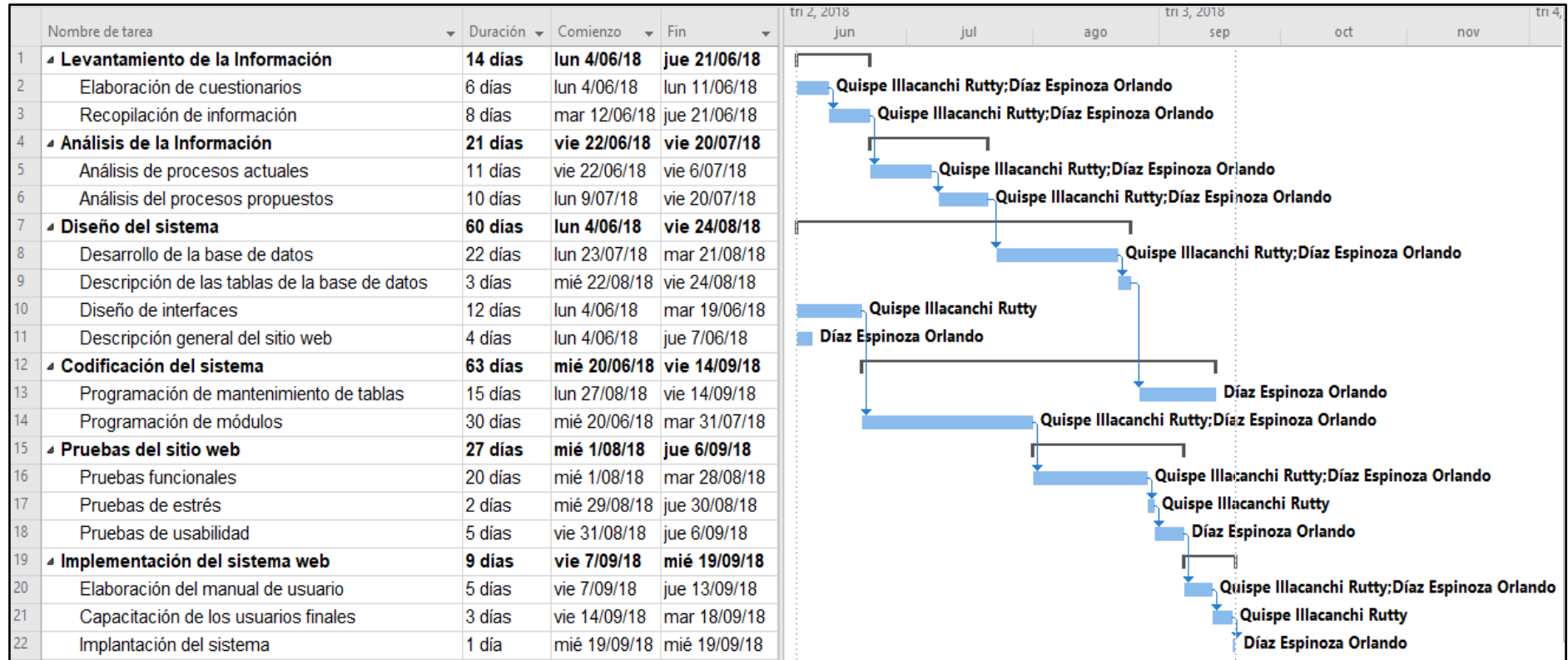
- Álvarez , G. (31 de Mayo de 2019). *Lenguaje Transact SQL o T-SQL*. Obtenido de Kyocode: <https://www.kyocode.com/2019/05/lenguaje-transact-sql/>
- AEMPI. (02 de Julio de 2019). *Vigilancia Canina*. Obtenido de Seguridad Atempí: <https://www.atempi.co/portfolio-item/vigilancia-canina/>
- CampusMVP. (16 de Mayo de 2017). *Qué es la plataforma .NET y cuáles son sus principales partes*. Obtenido de CampusMVP: <https://www.campusmvp.es/recursos/post/que-es-la-plataforma-net-y-cuales-son-sus-principales-partes.aspx>
- Cantillana Flores, F., & Inostroza Urrutia, V. (s.f. de s.f. de 2016). *Sistema de control de asistencia de personal de la Universidad del Bío-Bío*. Obtenido de Universidad del Bío-Bío: <http://repobib.ubiobio.cl/jspui/bitstream/123456789/1592/1/Cantillana%20Flores%20Felipe.pdf>
- CISSET. (2020). *Microsoft SQL Server*. Obtenido de Centro de Innovación y Soluciones Empresariales y Tecnológicas: <https://www.ciset.es/glosario/466-microsoft-sql-server>
- Clickage. (2018). *Aplicaciones de escritorio*. Obtenido de ClickAge Marketing Digital: <https://clickage.es/aplicaciones-de-escritorio/>
- Digital Guide IONOS. (26 de Octubre de 2018). *UML, lenguaje de modelado gráfico*. Obtenido de Digital Guide IONOS: <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/uml-lenguaje-unificado-de-modelado-orientado-a-objetos/>
- EspacioHonduras. (2020). *Concepto de Microsoft Visual Studio, ¿Qué es y para qué sirve Microsoft Visual Studio?* Obtenido de EspacioHonduras: <https://www.espaciohonduras.net/microsoft-visual-studio-concepto-y-que-es-y-para-que-sirve-microsoft-visual-studio>
- Fernández Alarcón, V. (2006). *Desarrollo de Sistemas de Información una Metodología Basada en el Modelado*. Barcelona: Edicions UPC.
- García Jimenez, O. E. (09 de Diciembre de 2014). *Concepto Visual Basic .Net*. Obtenido de VisualBasic.NET: <https://sites.google.com/site/vbbnet/visualbasic-net>

- García Mariscal, A. B. (2015). *Diseño de bases de datos relacionales*. España: Elearning S.L.
- García, I. (09 de Enero de 2018). *Definición de Usabilidad*. Obtenido de economía simple .net: <https://www.economiasimple.net/glosario/usabilidad>
- Huanca Figueroa, Y. Y. (s.f. de s.f. de 2017). *Implementación de un Sistema de Control Biométrico para la Institución Educativa San Martín de Porras La Victoria – Hhuarmey; 2017*. Obtenido de Universidad Católica Los Ángeles de Chimbote: [http://repositorio.uladech.edu.pe/bitstream/handle/123456789/2607/CONTROL\\_%20IMPLEMENTACION\\_HUANCA\\_FIGUEROA\\_YERALDINE\\_YOLANDA.pdf?sequence=1&isAllowed=y](http://repositorio.uladech.edu.pe/bitstream/handle/123456789/2607/CONTROL_%20IMPLEMENTACION_HUANCA_FIGUEROA_YERALDINE_YOLANDA.pdf?sequence=1&isAllowed=y)
- Hueso Ibáñez, L. (2015). *Administración de Sistemas Gestores de Base de Datos*. Madrid: Ra-Ma.
- Iparraguirre Sánchez, J. E., & Mendoza Requejo, H. S. (s.f de s.f. de 2018). *Diseño e Implementación de un Sistema Informático para el Proceso de Comercialización y Control de Asistencia del Personal mediante Dispositivo Biométrico, en la Botica “Lizfarma” – José Leonardo Ortiz*. Obtenido de Universidad Nacional Pedro Ruiz Gallo: <http://repositorio.unprg.edu.pe/bitstream/handle/UNPRG/1795/BC- TES-TMP-647.pdf?sequence=1&isAllowed=y>
- Kimaldi. (29 de Julio de 2019). *¿Qué es el control de personal?* Obtenido de Kimaldi: [https://www.kimaldi.com/blog/control\\_de\\_acceso\\_y\\_presencia/control\\_de\\_personal/](https://www.kimaldi.com/blog/control_de_acceso_y_presencia/control_de_personal/)
- Manrique Rimay, J. S. (s.f. de s.f. de 2015). *Optimizar la Administración de la Brigada Canina mediante el Desarrollo de un Sistema de Gestión Web*. Obtenido de Universidad Ricardo Palma: [http://repositorio.urp.edu.pe/bitstream/handle/URP/2036/manrique\\_js.pdf?sequence=1&isAllowed=y](http://repositorio.urp.edu.pe/bitstream/handle/URP/2036/manrique_js.pdf?sequence=1&isAllowed=y)
- Programa en Línea. (25 de Mayo de 2018). *¿Qué es el Proceso Unificado de Rational (RUP)?* Obtenido de Programa en Línea: <https://www.programaenlinea.net/proceso-unificado-rational-rup/>
- Tusa Pilapanta, I. F. (s.f. de s.f. de 2015). *La Automatización De Procesos Y su Incidencia en el Control de Asistencia Docente en la Unidad Educativa Darío Guevara, del Cantón Ambato Provincia de Tungurahua*. Obtenido de Universidad Técnica de Ambato:

<https://repositorio.uta.edu.ec/jspui/bitstream/123456789/20319/1/Tesis%20Isaias%20Tusa.pdf>

## **APÉNDICES**

**Apéndice A: Cronograma de Actividades**



**Apéndice B: Cronograma de Presupuesto**

<b>Materiales</b>			
<b>DETALLES</b>	<b>CANTIDAD</b>	<b>VALOR(S/)</b>	<b>TOTAL(S/.)</b>
Laptop	1	2800	S/. 2 800.00
Impresora	1	650	S/. 650.00
Papel bond	3	13	S/. 39.00
Memoria USB	2	25	S/. 50.00
Tinta de impresión	1	60	S/. 60.00
Sub total			<b>S/. 3 599.00</b>

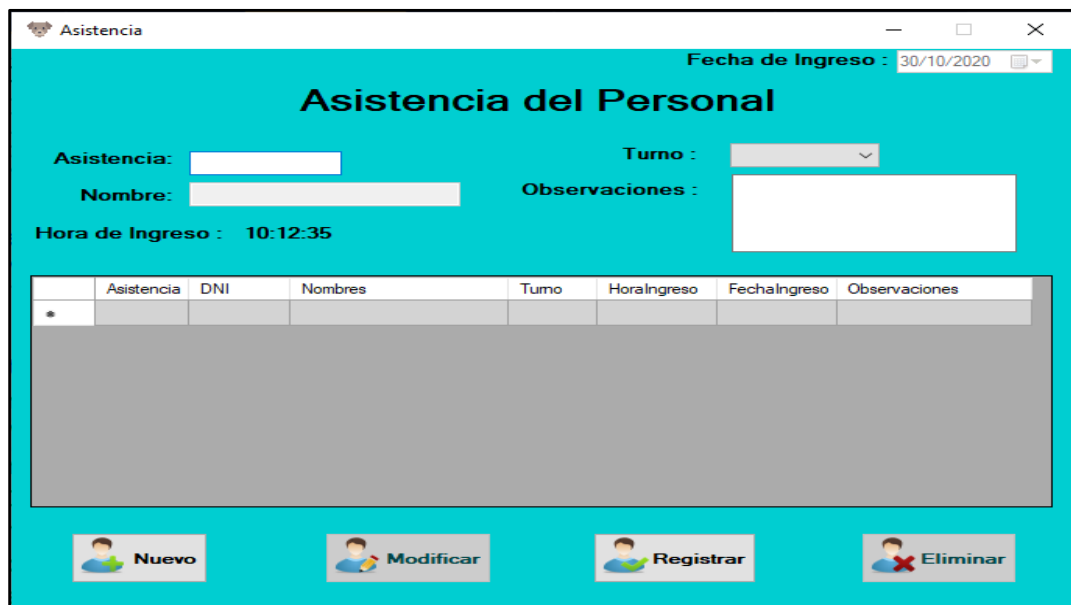
<b>Servicios</b>			
<b>DETALLES</b>	<b>CANTIDAD</b>	<b>VALOR(S/)</b>	<b>TOTAL(S/.)</b>
Internet	5	30	S/.150.00
Comunicaciones	2	100	S/. 200.00
Movilidad	2	100	S/. 200.00
Copias	100	0.1	S/. 10.00
Otros	1	50	S/. 50.00
Sub total			<b>S/. 610.00</b>
<b>Total general</b>			<b>S/. 4 209.00</b>

## Apéndice C: Manual de Usuario

Al ejecutar el Sistema de control Ormeño se muestra la siguiente ventana:



Al hacer clic en el botón Asistencia Personal se muestra la siguiente ventana, la misma que permite que cada personal pueda marcar su asistencia cada vez que ingresa a realizar sus actividades laborales.

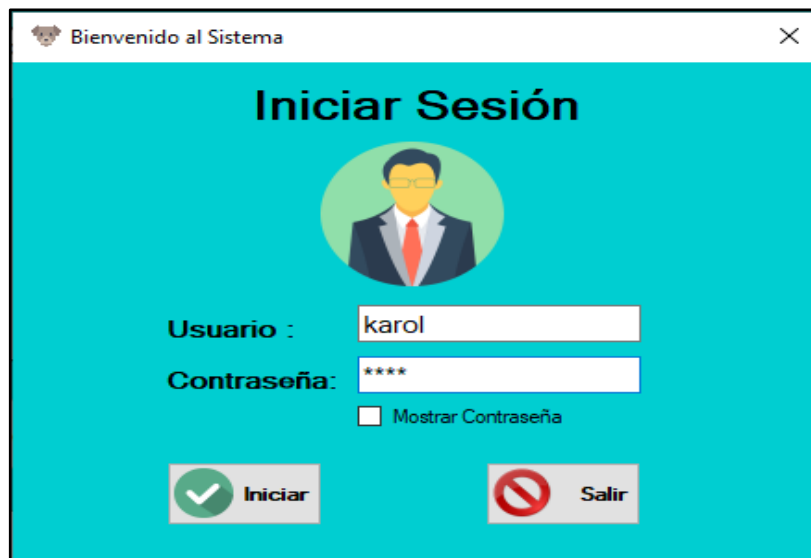


	Asistencia	DNI	Nombres	Turno	HoraIngreso	FechaIngreso	Observaciones
*							

Al ingresar si hacemos clic en el botón Administrador

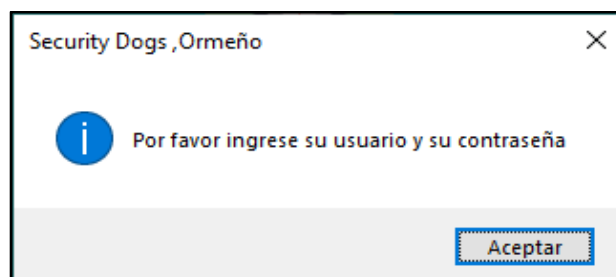


Se muestra la ventana de Bienvenido al Sistema:



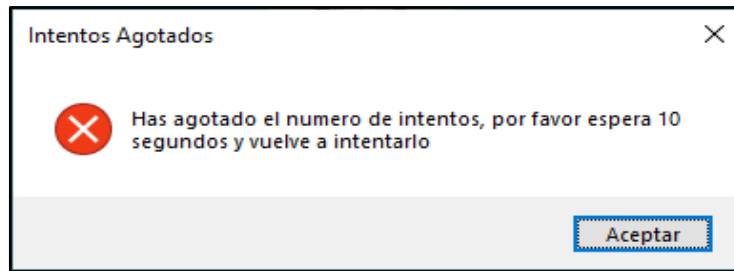
Aquí debemos colocar el usuario y contraseña del administrador y luego darle clic en el botón Iniciar.

Si el usuario y contraseña son incorrectas se muestra el siguiente mensaje:

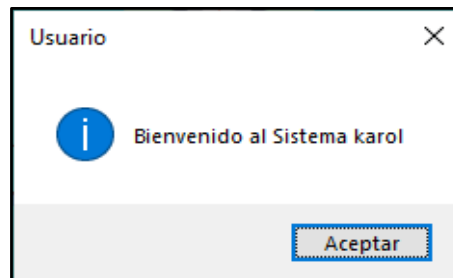


El sistema sólo te permite realizar hasta 3 intentos, si en caso se falla en los 3 intentos se muestra el siguiente mensaje:

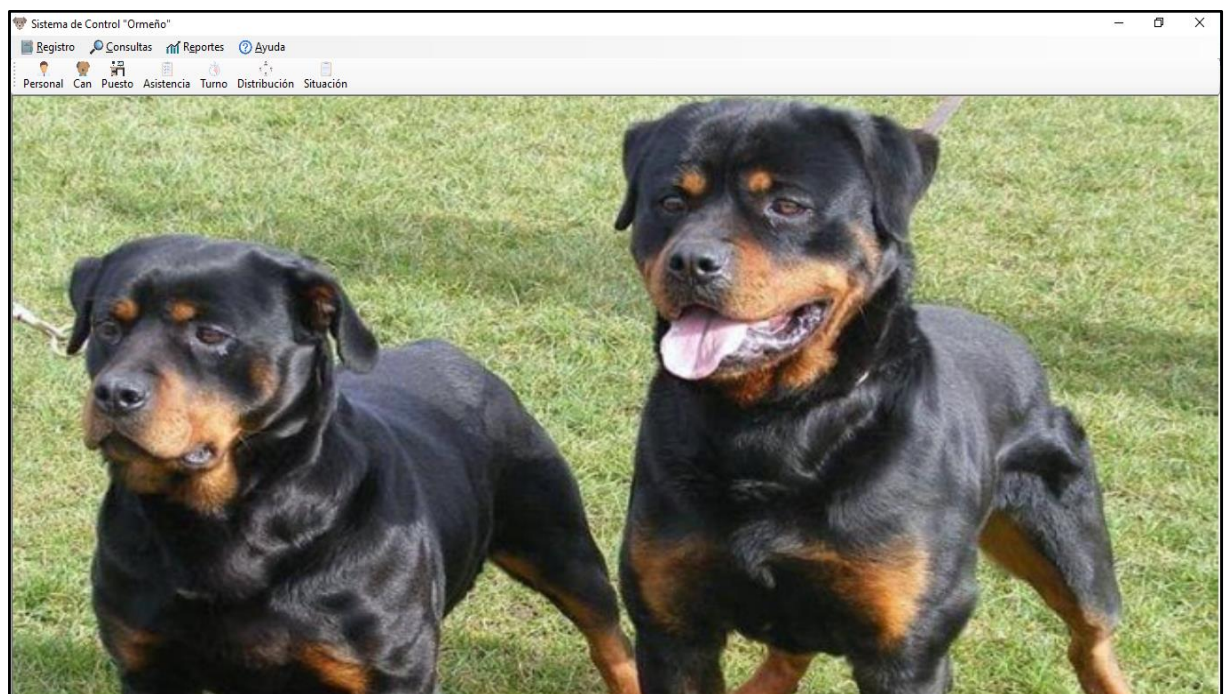




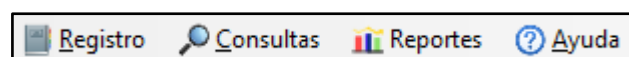
Si el usuario y contraseña son correctas se muestra el siguiente mensaje:



Si damos clic en el botón Aceptar, se muestra la ventana principal del sistema:

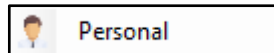
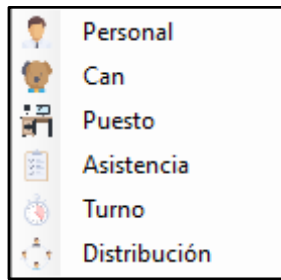


En la parte superior se muestra el siguiente menú:



**Opción Registro**

Al hacer clic aparecen las siguientes sub opciones:



- **Personal:** Aquí en este formulario podemos registrar, modificar y eliminar al personal que trabaja en la empresa.

Personal

## Registro del Personal

DNI:  Foto:  ...

Nombres :

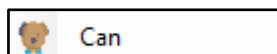
Apellidos :

Telefono :

Fecha de Ingreso : 2021-01-15

Direccion :

	DNI	Nombres	Apellidos	Telefono	Direccion	NomFoto	foto
▶	27077466	Jesus	Castro Salvatierra	9856427155	San Luis		C:\U
	27077467	Jesus	Castro Salvatierra	9856427155	San Luis		C:\U
	27077468	Jesus	Castro Salvatierra	9856427155	San Luis	Imagen	C:\U
	27077469	Jesus	Castro Salvatierra	9856427155	San Luis	1	C:\U
	27077470	Jesus	Castro Salvatierra	9856427155	San Luis	Imagen	C:\U
	27077479	Jesus	Castro Salvatierra	9856427155	San Luis	Imagen	C:\U
	27458212	Jhon	Chinchon Fuentes	963258741	Villa Maria del Tri...	Imagen	C:\U



- **Can:** Permite registrar, modificar y eliminar a los datos de los canes que brindan el servicio de vigilancia canina, datos como nombre, color, raza, fecha de nacimiento, así como la imagen del can.

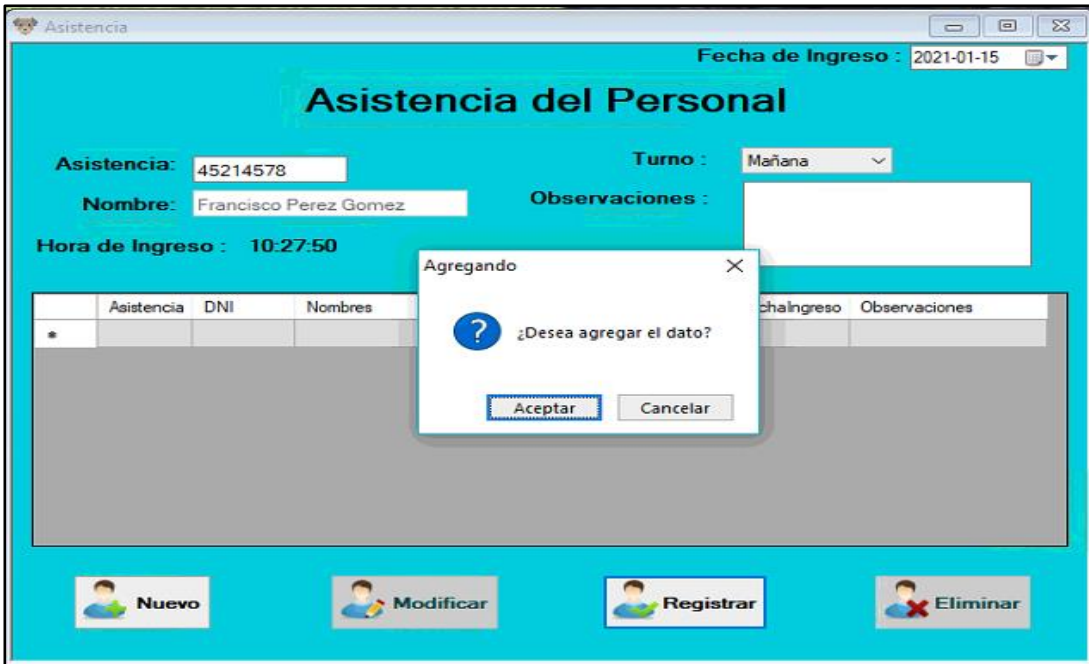


- **Puesto:** Se registran los diferentes puestos donde se presta el servicio de vigilancia canina.

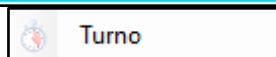
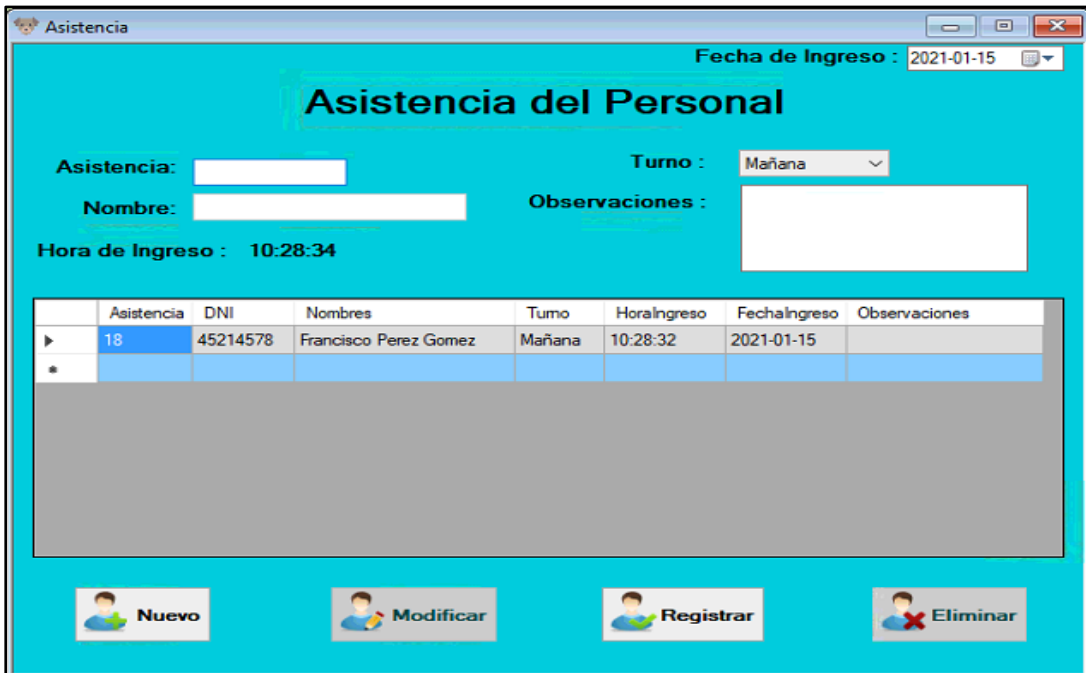


- **Asistencia:** Permite registrar la asistencia del personal, aquí podemos dar mantenimiento a los registros de asistencia del personal.

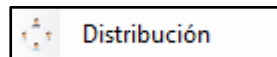
Ingresamos el DNI del personal y luego hacer clic en el botón Registrar, en el cual se muestra un mensaje de para confirmar o cancelar el registro de la asistencia.



Al hacer clic en el botón Aceptar, se muestra la asistencia registrada.



- **Turno:** esta opción nos permite ingresar, modificar y eliminar los turnos del personal cuando prestan el servicio de vigilancia canina, los turnos pueden ser mañana, tarde y noche.



- **Distribución:** Esta opción representa una de las opciones más importantes del sistema, ya que es donde se especifica la distribución tanto del personal como de los canes y en que puesto se brindará el servicio de vigilancia canina.



Una vez que seleccionamos el personal, el puesto y el can, hacemos clic en el botón Registrar, en la cual se muestra un mensaje de confirmación para que se registre dicha distribución.

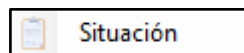


Al hacer clic en el botón Aceptar se registra la distribución respectiva.



**Opción Consultas**

Al hacer clic aparece la siguiente sub opción:



- **Situación:** Permite registrar algún tipo de incidencias ocurridas durante el servicio de vigilancia canina, además nos permite modificarla y eliminarla.

The screenshot shows a web application window titled 'Incidencias'. The main content area has a light blue background. At the top right, there is a date selector showing 'Fecha: 2021-01-15'. The main form contains the following fields and values:

- Incidencias :** INC00-7 (with a 'Nuevo' button)
- Fecha de asistencia:** 2021-01-15
- DNI:** 27458212
- Nombres y apellidos:** (empty text field)
- Estado :** Descanso Medico (dropdown menu)
- Turno:** (empty text field)
- Descripción :** Reprogramación de cita médica

A modal dialog box titled 'Agregando' is centered on the screen, containing a question mark icon and the text '¿Desea agregar el dato?'. Below the text are two buttons: 'Aceptar' and 'Cancelar'. At the bottom of the main window, there are three buttons: 'Modificar', 'Registrar', and 'Eliminar', each with a corresponding icon.

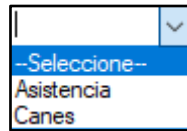
### Opción Reportes

- **Asistencias y Canes:** En esta opción se muestra un reporte de Asistencia y de Canes.

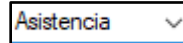
The screenshot shows a web application window titled 'REPORTE DE ASISTENCIA'. The main content area is mostly empty, suggesting a report that has not yet been generated or is currently loading. On the left side, there is a sidebar with the following elements:

- Fechas:** A section containing a dropdown menu set to 'Asistencia', and two date pickers labeled 'FECHA' and 'HASTA', both set to '2021-01-15'.
- VER FECHAS:** A button with a calendar icon.
- IMPRIMIR:** A button with a printer icon.

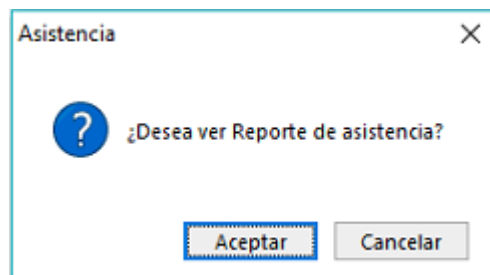
De la lista:



Seleccionamos la opción Asistencia



Y colocamos entre que fechas deseamos ver el Reporte, y damos clic en el botón VER FECHAS, donde se muestra el siguiente mensaje de confirmación.

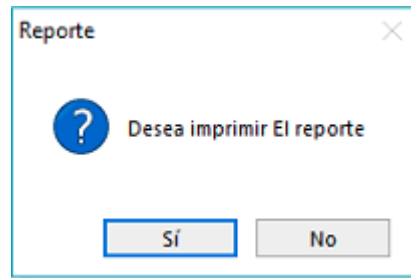


Al hacer clic en Aceptar se muestra el siguiente resultado:

Cod	DNI	Nombres	Direccion	Telefono	Registro	Tumo	Estado	Descripcion	Hora
INC00-6...	76581253	Cristian aROTNCO Torres	z b Lote 1	966975388	26/05/2019	Tarde	Permiso	ttit	23:00:16

Y al hacer clic en el botón IMPRIMIR, se muestra el siguiente mensaje de confirmación.

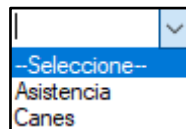




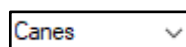
Al hacer clic en el botón Sí, se muestra el siguiente reporte:

COD	DNI	NOMBRES	DIRECCIÓN	TELÉFONO	REGISTRO	DETALLE	TURNO	ESTADO
INC00-6	76581253	Crishian aROTNCO Torres	z b Lote 1	966975388	2021-01-14	ttrrt	Tarde	Permi

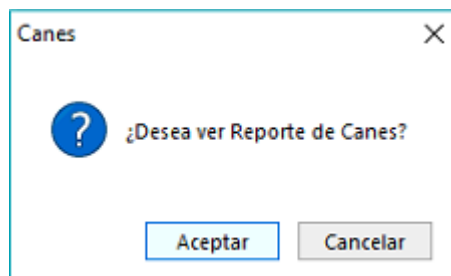
De la lista:



Seleccionamos la opción Canes



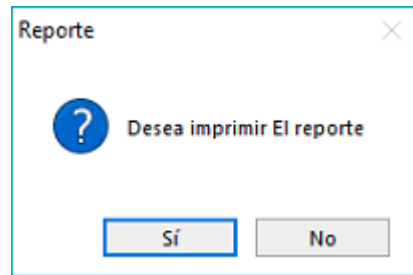
Y colocamos entre que fechas deseamos ver el Reporte, y damos clic en el botón VER FECHAS, donde se muestra el siguiente mensaje de confirmación.



Al hacer clic en Aceptar se muestra el siguiente resultado:

CodCan	Can	Color	Raza	Nacimiento	Tipo	Estado	fecha
39	zzz	221	212	2020-12-09	Veterinario	Activo	2020-12-09
41	zzz	221	212	2020-12-09	Veterinario	Activo	2020-12-09
49	212	1212	122121	2020-12-10	Baño	No Activo	2020-12-10
*							

Y al hacer clic en el botón IMPRIMIR, se muestra el siguiente mensaje de confirmación.



Al hacer clic en el botón Sí, se muestra el siguiente reporte:

REPORTES CAN

de 1 de 1 100% Buscar | Siguiete

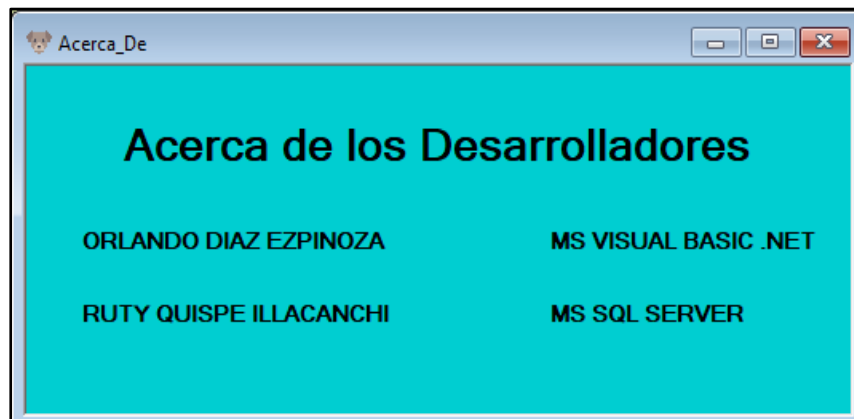
### Reportes de Canes

COD	CAN	COLOR	RAZA	NACIMIENTO	TIPO	ESTADO	FECHA
39	zzz	221	212	09/12/2020	Veterinario	Activo	09/12/2020
41	zzz	221	212	09/12/2020	Veterinario	Activo	09/12/2020
49	212	1212	122121	10/12/2020	Baño	No Activo	10/12/2020

### Opción Ayuda

Al hacer clic se muestra la siguiente sub opción:

➤ **Acerca de:** En esta opción se muestra el grupo de desarrolladores del sistema de control Ormeño.



Así mismo, en la parte superior de la ventana principal se muestra la siguiente barra con diferentes accesos rápidos a diferentes opciones las mismas que ya fueron descritas anteriormente.

