

ESCUELA POLITÉCNICA DEL EJÉRCITO

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

**CARRERA DE INGENIERÍA EN ELECTRÓNICA,
AUTOMATIZACIÓN Y CONTROL**

**PROYECTO DE GRADO PARA LA OBTENCIÓN DEL
TÍTULO EN INGENIERÍA ELECTRÓNICA**

**“DISEÑO E IMPLEMENTACIÓN DE UNA APLICACIÓN
DOMÓTICA EN UN HOGAR TIPO MEDIANTE EL USO DE LA
TECNOLOGÍA LONWORKS”**

**Vinicio Antonio Moya Almeida
Wladimir David Angulo Villacís**

SANGOLQUÍ - ECUADOR

2011

CERTIFICACIÓN

Certificamos que el presente proyecto de grado titulado “Diseño e implementación de una aplicación domótica en un hogar tipo mediante el uso de la tecnología LonWorks” ha sido desarrollado en su totalidad por los Sres. Vinicio Antonio Moya Almeida con C.I. 0501617963 y Wladimir David Angulo Villacís con C.I. 1716642218, bajo nuestra dirección.

Ing. Marcelo Escobar
DIRECTOR

Ing. Jaime Andrango
CODIRECTOR

AGRADECIMIENTO

En primer lugar a Dios, compañero inseparable de estos años de estudio, y quien fue quien me guió y dio la fortaleza para seguir adelante y convertir los obstáculos en oportunidades.

A mis padres quienes me inculcaron la importancia del deber cumplido y me apoyaron en todo momento brindándome su amor y cuidado. A todas las personas que de alguna manera forman parte de mi vida y por su cariño he sentido la responsabilidad de ser un buen ejemplo persona, cristiano y ciudadano.

A la Escuela Politécnica del Ejército, a sus profesores, a mis compañeros quienes han aportado de muchas formas a mi formación académica y personal

Vinicio Moya Almeida

AGRADECIMIENTO

Principalmente a Dios, creador de la vida, que cada día me maravilla al conocer un poco mas de la ciencia y los limites que la sustentan, que ha permitido todas las experiencias de conocimiento y de vida que demuestran su amor y poder, quien sustenta mi vida con amor y es la fuerza que me inspira en cada aspecto de la vida.

A mis padres, que me han provisto de su apoyo en todos los sentidos que les ha sido posible día a día, el ejemplo de vida que me han sabido dar, y los valores que he aprendido de ellos. A mis lideres y maestros que han ayudado a fundamentar en mi una vida basada en convicciones.

A cada miembro de la Escuela Politécnica del Ejército, tanto profesores como compañeros, que intervinieron en el desarrollo de la carrera, que con su experiencia y trato contribuyeron al desarrollo y formación de conocimientos académicos, deportivos y personales.

Wladimir Angulo Villacís

DEDICATORIA

Dedico esta tesis a la Escuela Politécnica del Ejército, institución que me brindó la formación académica que hoy me permite presentar este trabajo de investigación; a todos y cada uno de mis maestros; a mi Director de Tesis Ing. Escobar y a mi Codirector Ing. Andrango por su apoyo en la realización de este trabajo; y a todas las personas que en algún momento pueda servirles y sean aficionados como yo, a la Domótica.

Vinicio Moya Almeida

DEDICATORIA

Dedico esta tesis a las personas cercanas en mi vida tanto familiares como amigos. A la Escuela Politécnica del Ejercito, institución que permitió que todo esto se desarrolle. A mis maestros que ayudaron en la formación académica, y a cada persona que le sea de interés y utilidad la información generada y recopilada en este trabajo de tesis.

Wladimir Angulo Villacís

PRÓLOGO

El presente documento recoge el diseño y la implementación de la domótica de un hogar tipo, la cual puede ser controlada mediante un HMI. Dicha casa se encuentra en la provincia de Pichincha, ciudad Sangolquí, calles Venezuela y Viñedos, Ciudadela Terracota F, casa 31.

Uno de los hechos más importantes del último siglo es el acelerado progreso de la electrónica gracias a la invención y descubrimiento de semiconductores, circuitos integrados, resistencias, entre otros, convirtiéndose en inevitable la introducción de la electrónica en los diferentes aspectos de la vida cotidiana del ser humano común.

La domótica es la rama de la electrónica que se encarga del desarrollo de la tecnología en torno a las casas inteligentes. Esta rama apareció debido a la necesidad de integrar los diferentes sistemas que unas décadas atrás ya existían pero funcionaban por separado, provocando problemas al momento de controlarlos, supervisarlos, brindarles mantenimiento, etc. Es así donde aparecieron diferentes protocolos domóticos entre los cuales los más conocidos y expandidos son LonWorks, Bacnet, X10, ZigBee, KNX, etc.

El presente trabajo se enfoca en la utilización de la tecnología LonWorks, inventada y desarrollada por la empresa Echelon radicada en San José, California, en los Estados Unidos.

El diseño de la red domótica se desarrolla desde el punto de vista de las necesidades de los miembros del núcleo familiar que habitan en el hogar tipo. Se detallan los planos arquitectónicos, eléctricos, de preinstalación domótica, domóticos, y los unifilares. También se realiza un corto análisis sobre el consumo de energía eléctrica y el costo de una instalación domótica.

También se detalla el desarrollo del HMI en Microsoft Visual Basic 6.0 y la conexión con la red domótica mediante el uso de DDE. El HMI permite el control y la supervisión de todos los sistemas que se incluye en el presente trabajo.

ÍNDICE

AGRADECIMIENTO	2
AGRADECIMIENTO	3
DEDICATORIA	4
DEDICATORIA	5
PRÓLOGO	6
ÍNDICE	8
CAPÍTULO I.....	1
MARCO TEÓRICO Y GENERALIDADES.....	1
1.1 DOMÓTICA, INMÓTICA Y HOGAR DIGITAL	1
1.2 CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS	3
1.2.1 CARACTERÍSTICAS.....	3
1.2.2 VENTAJAS	9
1.2.3 DESVENTAJAS.....	11
1.3 ENTORNO SOCIAL, ECONÓMICO Y LEGAL EN EL ECUADOR.....	12
1.3.1 SECTORES INTERESADOS EN LA DOMÓTICA	13
1.4 NIVELES DE AUTOMATIZACIÓN	14
1.5 ARQUITECTURAS DE SISTEMAS DOMÓTICOS.....	19
1.5.1 ARQUITECTURA CENTRALIZADA	19
1.5.2 ARQUITECTURA DISTRIBUIDA:	20
1.6 MEDIOS DE TRANSMISIÓN	21
1.6.1 LÍNEAS DE DISTRIBUCIÓN DE ENERGÍA ELÉCTRICA (CORRIENTES PORTADORAS).....	21
1.6.2 CABLE METÁLICO	22
1.7 TOPOLOGÍAS	27
1.7.1 TOPOLOGÍA DE ESTRELLA	27
1.7.2 TOPOLOGÍA DE BUS	28
1.7.3 TOPOLOGÍA HÍBRIDA.....	28
1.8 PROTOCOLOS DOMÓTICOS	29
1.8.1 SISTEMAS CON PROTOCOLOS PROPIETARIOS.....	29

1.8.2	SISTEMAS CON PROTOCOLOS ESTÁNDARES O ABIERTOS	30
1.8.3	COMPARACIÓN ENTRE PROTOCOLOS.....	31
1.9	TECNOLOGÍA LonWorks.....	33
1.9.1	CONCEPTOS.....	33
1.9.2	ESTRUCTURA.....	35
1.9.3	HARDWARE Y SOFTWARE. NEURON CHIP.....	37
1.9.4	APLICACIONES.....	41
1.10	HMI.....	45
1.10.1	CONCEPTOS	45
1.10.2	FUNCIONES DE UN SOFTWARE HMI	48
1.10.3	MODELO DE DISEÑO DE UNA INTERFAZ HMI.....	49
1.10.4	HMI EXISTENTES EN EL MERCADO	51
CAPÍTULO II.....		57
DISEÑO DOMÓTICO		57
2.1	DESCRIPCIÓN DE LA VIVIENDA	57
2.1.1	PLANOS ARQUITECTÓNICOS.....	58
2.1.2	PLANOS ELÉCTRICOS	60
2.1.3	VIVIENDA ANTES DE LA AUTOMATIZACIÓN	62
2.1.4	ANÁLISIS DE NECESIDADES DE LOS DUEÑOS DEL HOGAR TIPO.....	65
2.2	DISEÑO GENERAL DEL SISTEMA DOMÓTICO.....	66
2.2.1	EQUIPOS LONWORKS A UTILIZARSE.....	66
2.2.2	MEMORIA FUNCIONAL.....	67
2.2.3	PROGRAMACIÓN DE NODOS DOMÓTICOS	79
2.2.4	PLANOS DE PREINSTALACIÓN.....	110
2.2.5	PLANO DOMÓTICO	112
2.2.6	PLANOS UNIFILARES	114
2.3	OBRA CIVIL E INSTALACIÓN DE NODOS DOMÓTICOS	115
2.3.1	TRANSFORMACIONES Y ADAPTACIONES NECESARIAS PREVIAS A LA AUTOMATIZACIÓN.....	116
2.3.2	INSTALACIÓN DE LOS DISTINTOS NODOS LONWORKS, SENSORES Y ACTUADORES NECESARIOS.....	118
2.3.3	ANÁLISIS DE LAS MEDICIONES DEL MEDIDOR DE ENERGÍA ELÉCTRICA	120
2.3.4	ANÁLISIS DE COSTOS, INCLUYENDO OBRA CIVIL	122
CAPÍTULO III		125
DISEÑO DEL HMI		125
3.1	COMPATIBILIDAD CON LONWORKS	125
3.1.1	DDE (Dynamic Data Exchange).....	125

3.1.2	LNS SERVER.....	126
3.1.3	LNS DDE Server.....	126
3.1.4	ANÁLISIS DE COSTOS DE LICENCIAS	127
3.2	SELECCIÓN DEL PROGRAMA A UTILIZAR	128
3.3	REQUERIMIENTOS DEL HMI	129
3.3.1	REQUERIMIENTOS DEL USUARIO	129
3.3.2	REQUERIMIENTOS DEL SISTEMA OPERATIVO Y EQUIPOS	131
3.4	PROGRAMACIÓN DEL HMI.....	133
3.4.1	DIAGRAMA DE FLUJO.....	138
3.4.2	IMPLEMENTACIÓN DE NIVELES DE ACCESO DE ACUERDO A LOS REQUERIDOS POR EL DUEÑO DE CASA	141
3.4.3	DISEÑO E IMPLEMENTACIÓN DE LA VENTANA PRINCIPAL DEL SOFTWARE HMI.....	155
3.4.4	DISEÑO E IMPLEMENTACIÓN DE LA VENTANA GESTIÓN DE ALARMAS.....	204
3.5	INTEGRACIÓN DEL HMI EN EL SISTEMA DOMÓTICO	216
CONCLUSIONES Y RECOMENDACIONES		217
CONCLUSIONES		217
RECOMENDACIONES		219
BIBLIOGRAFÍA		222
ANEXOS.....		224
ÍNDICE DE CUADROS Y FIGURAS.....		237
ÍNDICE DE TABLAS		240
GLOSARIO		241

CAPÍTULO I

MARCO TEÓRICO Y GENERALIDADES

1.1 DOMÓTICA, INMÓTICA Y HOGAR DIGITAL

Domótica, inmótica y hogar digital son términos que poco a poco han dejado de ser desconocidos para pasar a ser parte del cada día más creciente mundo tecnológico en el cual vivimos en la actualidad, donde América Latina no es la excepción.

El vocablo “domótica” fue acuñado originalmente por los franceses, quienes tomando del latín *domus*, que significa “casa”, y el griego *tica*, que hace acepción de algo que “funciona por sí solo”, quisieron dar a entender lo que hoy la Real Academia de la Lengua Española define como el “*conjunto de sistemas que automatizan las diferentes instalaciones de la vivienda*”. Por esto es posible decir que la domótica no es más que la integración de las distintas Tecnologías de la Comunicación con la casa en sí.

Al punto al cual se desea llegar es donde se integren diferentes áreas de las ciencias exactas con el fin de lograr un hogar más confortable, seguro, ahorrativo y lleno de bienestar en comparación con las casas tradicionales. Esto es posible después de la fusión de la Arquitectura e Ingeniería Civil, en cuanto a la construcción de la casa en sí, más la Electrónica, Robótica, Informática, y Eléctrica en cuanto a la

automatización de la vivienda. Esta unificación es lo conocemos como “Hogar Digital”.

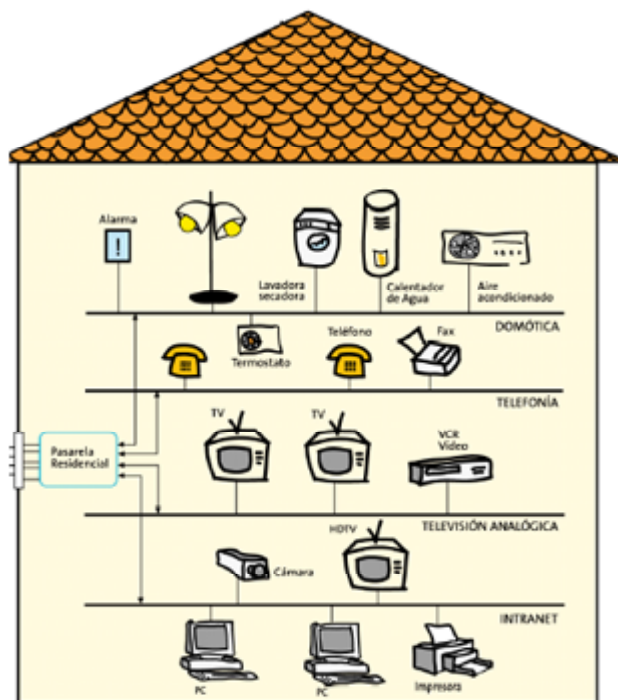


Fig. 1 Elementos de un hogar digital

En este punto es necesario dejar en claro que la integración de todos estos sistemas es por lo general costosa y de difícil implementación desde el punto de vista técnico, por lo cual aún se realiza por separado la instalación de algunos de estos sistemas, sin embargo en un futuro cercano la idea de Hogar Digital será una completa realidad.

Por otro lado podemos definir a la Inmótica como la automatización e integración de los diferentes sistemas de información en todo un edificio de uso terciario o industrial, como por ejemplo hoteles, edificios corporativos, oficinas, entre otros similares. Por lo tanto contará con otro tipo de consideraciones técnicas y de necesidades de los usuarios en comparación con la domótica; por ejemplo aquí el

ahorro energético será uno de los aspectos de mayor consideración al momento del diseño inmótico del edificio.



Fig. 2 Urban Plaza

En la Bienal de Miami Beach en el año 2009 ganó la medalla de oro el edificio Urban Plaza de la constructora Uribe & Schwarzkopf y ubicado en la ciudad de Quito, el cual fue concebido desde sus inicios como un edificio que mezcle modernidad, expresada en inmótica, con su particular arquitectura.

1.2 CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS

1.2.1 CARACTERÍSTICAS

La domótica como parte de las nuevas tecnologías de la información siempre buscará la intercomunicación de todos los equipos asociados a una red de control. Por lo cual sus principales características son: Integración, interrelación, facilidad de manipulación, control a distancia (remoto), fiabilidad, actualización y escalamiento, y ahorro energético.

1.2.1.1 Integración

Es de vital importancia la capacidad de un sistema para que este sea integrable íntegramente, esto quiere decir que no puede existir problemas en las comunicaciones entre distintos equipos de diferentes fabricantes, claro está, siempre y cuando estos funcionen bajo los mismos protocolos de comunicación.

1.2.1.2 Interrelación

Se pueden conectar sobre la misma red a distintos tipos de dispositivos, equipos y sistemas, como por ejemplo, utilizar un mismo sensor de movimiento para los sistemas de iluminación y de seguridad contra intrusos. Sin esta característica los sistemas domóticos simplemente tendrían costos demasiado elevados además de no poder hablarse de la integración del sistema.

1.2.1.3 Facilidad de manipulación

Los sistemas domóticos serán manipulados en alrededor del 95% por personas que carecen de conocimientos en el área de la electrónica o la programación, por lo cual deberán ser completamente amigables y simples de utilizar. El usuario debe estar en la capacidad de que con pocos clics de un ratón, o unos simples botones en una consola, se puedan dominar la totalidad de los sistemas instalados en la vivienda. Además a esto se puede obtener información sobre el estado de la vivienda y el sistema con solo observar en una pantalla instalada en el inmueble.

1.2.1.4 Control a distancia (remoto)

El usuario está en la capacidad de mantener el control y monitoreo de su vivienda las 24 horas del día y desde cualquier lugar tal cual los estuviera realizando localmente, por ello es necesario que se aprovechen otras tecnologías como el celular y el internet para conseguir este fin.

1.2.1.5 Fiabilidad

El sistema domótico funciona, con ayuda de sistemas de alimentación ininterrumpida, manejo inteligente de la energía, baterías de gran capacidad, etc., todo el tiempo y ante cualquier falla del sistema eléctrico externo. Por otra parte se debe garantizar, en el diseño de la red, que esta no vaya a colapsar por exceso de tráfico o cualquier otro motivo similar. De hecho se debe procurar que la única fuente de fallos del sistema sean las conocidas como “falla humana”.

1.2.1.6 Actualización y escalamiento

La red de control, y los equipos utilizados tienen la característica de que se puede modificar su software en el momento deseado sin realizar prácticamente ninguna intervención en el hardware de los mismos. Esto nos da la ventaja de modificar, mejorar o simplemente actualizar los distintos programas que controlan la vivienda sin realizar alteración alguna en el sistema físico domótico.

Es importante mencionar que aunque no todos los estándares domóticos permiten tan fácilmente la escalabilidad, sí se debe procurar que exista dicha capacidad; la cual se trata permite el aumento de la capacidad física del sistema sin que esto involucre mayores cambios en la red instalada.

1.2.1.7 Ahorro energético

Quizás esta es en la actualidad una de las características que más llama la atención de los sectores interesados en la domótica y la inmótica, ya que factores como la contaminación a nivel mundial, la escasez de fuentes de energía, el calentamiento global y la conciencia desarrollada en los últimos años por la humanidad en general sobre el cuidado del mundo ha hecho que poco a poco y en todas las áreas de las ciencias se busque encontrar métodos y tecnologías que hagan las mismas funciones con un consumo menor de energía que sus pares actuales.

En edificaciones en las que se realiza gestión energética se ha llegado a comprobar la existencia de hasta más del 50% de ahorro de energía cuando existe una automatización de alto nivel; esta cifra es más notoria en regiones en las que se utilizan sistemas de climatización, ya que estos son los responsables de alrededor del 75% del gasto energético, frente a un 20% que representa la iluminación y un 5% entre otros rubros.¹

En el año 2002 el Parlamento Europeo promulgo la directiva 2002/91 con el fin de estudiar la calidad energética en edificios. Tema en el cual por obligatoriedad se debe incluir a la inmótica. El encargado de la realización de este estudio fue el Dr. Becker de la Universidad de Biberach, Alemania, tomando en cuenta diferentes condiciones climáticas y los microclimas formados por las construcciones del hombre. Después de comparar un edificio promedio y uno automatizado se llegaron a obtener las siguientes cifras en cuanto a ahorro energético:²

- 1) Iluminación.
 - a. Con regulación constante de LUX: 50%.
 - b. Con regulación por presencia: 15%.

¹ Carlos Navares, *Inmótica y eficiencia energética: Gestión en tiempo real de las aplicaciones (Madrid)*, Artículo, pp. 1-3

² *Estudios de ahorro de costes sobre la eficiencia energética*, 2003, (AVANÇ)

- 2) Climatización.
 - a. Calefacción.
 - i. Regulación por tiempo o presencia: 25%.
 - ii. Regulación con apertura de ventanas: 15%.
 - b. Aire acondicionado.
 - i. Regulación por tiempo o presencia: 15%.
 - ii. Regulación de persianas: 10%.
 - iii. Regulación con ventiladores: 15%.

- 3) Energía solar.
 - a. Foto voltaica: 10%.
 - b. Térmica.
 - i. Cobertura en agua caliente sanitaria: 70 – 80%.
 - ii. Calefacción por suelo radiante: 25 – 30%.

Haciendo una relación a estas cifras anteriores se ha llegado a determinar que el ahorro anual en una estancia de 30 m^2 con control de la gestión energética puede bordear los 55 USD.

A continuación observaremos algunas figuras en las que se representa el ahorro energético obtenido experimentalmente en tres tipos de edificaciones en las áreas de la Industria, Edificios y Residencial.

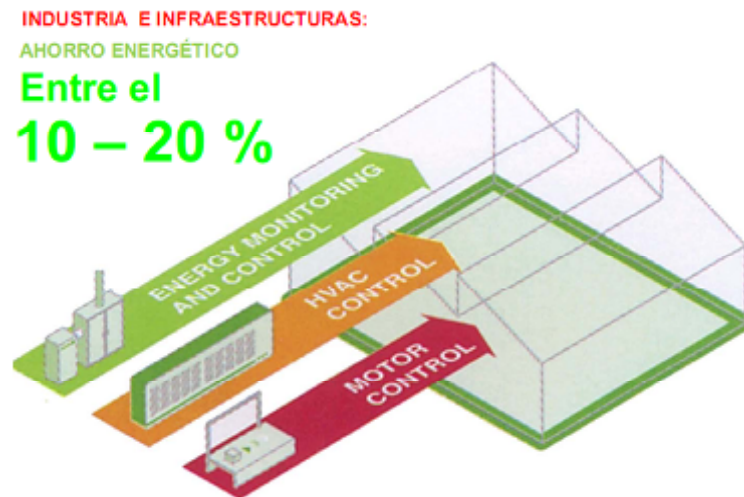


Fig. 3 Ahorro energético en Industrias

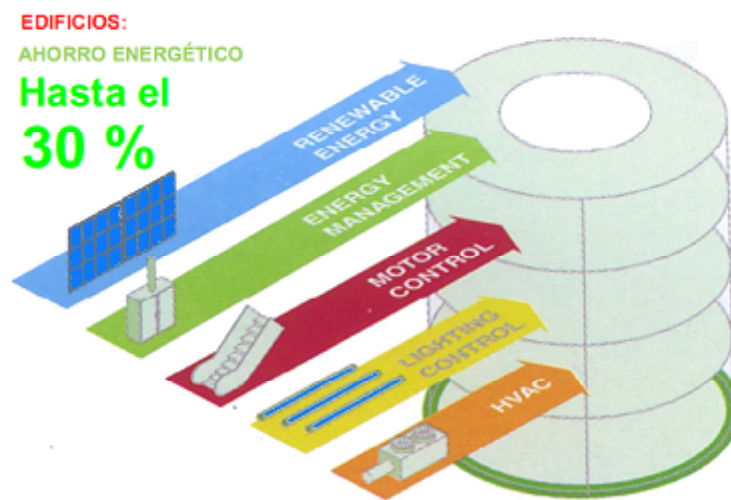


Fig. 4 Ahorro energético en Edificios

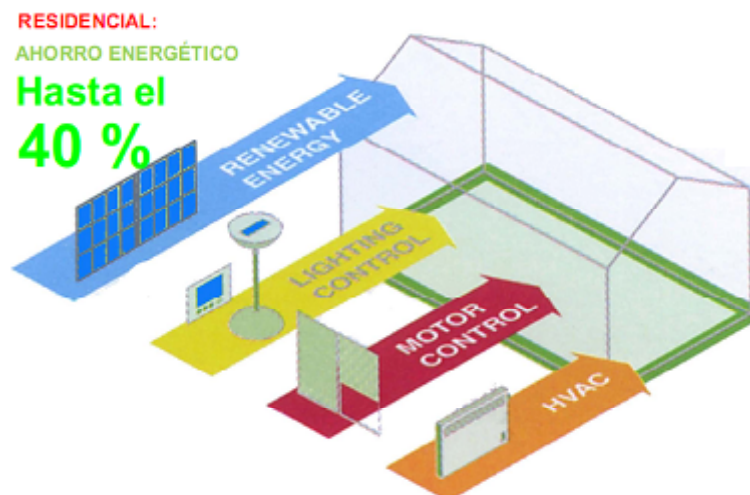


Fig. 5 Ahorro energético en Residencias

1.2.2 VENTAJAS

Por todas las características ya mencionadas anteriormente podemos deducir algunas de las más relevantes ventajas que presenta el utilizar la domótica / inmótica en nuestro hogar u oficina, entre las cuales podemos nombrar la reducción de gastos por manutención y mantenimiento, mayor seguridad, flexibilidad y sencillez, comunicaciones y el confort.

1.2.2.1 Reducción de gastos por manutención y mantenimiento

Al existir un uso eficiente de la energía evidentemente se reflejará en las planillas de electricidad y gas (en el caso de sistemas de calefacción con gas). Aparte de esto el mantenimiento es realmente poco, ya que son sistemas diseñados casi en su totalidad para soportar diferentes situaciones como el sobre voltaje y la sobre corriente.



Fig. 6 Ahorro energético

1.2.2.2 Mayor seguridad

La mayor parte de aspectos de seguridad se verán controlados, como por ejemplo el control de accesos (sobre todo en el caso de edificios), vigilancia con circuitos de CCTV y sensores de movimiento, y sistemas de simulación de presencia, que procuran engañar a cualquier persona asemejando que un ser humano se encuentre prendiendo y apagando luces, climatización, etc., de manera aleatoria.



Fig. 7 Gestión de la seguridad

1.2.2.3 Flexibilidad y sencillez

En muchos casos, dependiendo de algunos factores como el presupuesto, será necesaria la instalación de parte del sistema domótico con miras a ser ampliado en el futuro, y esto sucede sin que sea necesario rediseñar o reestructurar los elementos ya instalados en la red de control. De la misma manera puede darse el caso en el cual el dueño de la edificación desee cambios en el software o la programación de su sistema, y esto es posible sin mayor complicación ni dificultad para el programador.

1.2.2.4 Comunicaciones

Aun cuando todavía en este sentido la domótica no se encuentra completamente desarrollada es la gran tendencia mantener una completa comunicación entre todos los dispositivos electrónicos del hogar, como por ejemplo, poder observar a través de la televisión el estado de las alarmas, controlar la climatización y poder observar el teleportero. Sin embargo al paso agigantado que avanza la tecnología se espera que muy pronto esta integración sea completa y total.

1.2.2.5 Confort

El poder controlar el hogar desde un mismo punto, mientras que la iluminación se coloca al nivel deseado por el usuario y con la música preferida de fondo es una verdadera escena de confort obtenida por estos sistemas. Así se puede experimentar un crecimiento en la calidad y el nivel de vida, ahorrando energía y con un alto nivel de bienestar.

1.2.3 DESVENTAJAS

Definitivamente la única gran desventaja que enfrenta la domótica y la inmótica es el aspecto económico, y aun cuando frente al costo que involucra el diseño y construcción de una casa o un edificio el costo de la

automatización es bajo, todavía este sigue siendo considerable como para que muchos arquitectos e ingenieros civiles lo consideren como algo imprescindible en sus obras.

Con todo esto, hay que considerar que a pesar de que a lo largo de la historia los costos de nuevas tecnologías siempre han sido elevados, con la masificación y los rápidos adelantos tecnológicos estos tienden a decrecer continuamente. Además de esto gracias a la rápida adopción que la sociedad tiene por la tecnología y la necesidad creada por el consumismo se espera que en pocos años más este sector sea tan común como el impacto que en su momento tuvieron aparatos como el celular y la televisión.

1.3 ENTORNO SOCIAL, ECONÓMICO Y LEGAL EN EL ECUADOR

Para el presente proyecto es importante e interesante, aunque no imprescindible, entender el entorno social, económico y legal en el cual se encuentra el Ecuador con respecto a la domótica.

En primer lugar es importante decir que este tipo de instalaciones se encuentran enfocadas a un entorno social de clase media alta hacia alta, por lo cual el mercado de consumidores existentes se ve reducido a personas ante las cuales en muchos casos no existen limitaciones económicas. Por otra parte la mayor parte de este grupo social tienen una gran tendencia hacia adoptar nuevos tipos de tecnologías rápidamente. Sin embargo, son arquitectos, ingenieros civiles y afines aquellos a quienes se les realizarán ventas en primer instancia, y es a estos a quienes se les debe demostrar las diferentes ventajas que la domótica tiene.

Desde el punto de vista legal el Ecuador, como la mayor parte de América Latina, carece de una legislación enfocada directamente hacia la automatización de casas y edificios.

1.3.1 SECTORES INTERESADOS EN LA DOMÓTICA

Los sectores más interesados en la domótica son el área de la construcción, los promotores inmobiliarios, Municipios, Gobierno y los colegios profesionales, ya que todos estos realizan importantes aplicaciones de obra civil.

Los promotores inmobiliarios y las constructoras son el principal mercado, de la domótica y la inmótica, a pesar de que en el país aún existe un desconocimiento general de estos sistemas, lo cual es la principal barrera a ser enfrentada por las empresas integradoras que ofrecen este tipo de servicios.

Por otro lado el Gobierno y los Municipios intervienen en el momento en el cual se realizan planos eléctricos de las edificaciones creadas por las constructoras. Su presencia es indispensable al momento de legislar de tal manera que se creen los mecanismos legales para la existencia de certificaciones de que una obra se encuentra realmente automatizada. En este tema también podría haber una intervención de los Colegios Profesionales de Ingenieros Electrónicos, de Arquitectos e Ingenieros Civiles, con el fin de estandarizar y emitir las certificaciones antes mencionadas. También es importante que se dicten cursos y capacitación sobre el tema con el fin de difundir de mejor manera las bondades de este tipo de tecnologías.

1.4 NIVELES DE AUTOMATIZACIÓN

Los Niveles de Automatización son un conjunto de diferentes valoraciones dados a aplicaciones domóticas con el fin de entender cuán avanzada o compleja es la red de control, y aunque no son de obligado cumplimiento, sí son un buen indicador para poder comparar dos obras diferentes. Nacen como iniciativa de ciertas integradoras y su uso está difundido principalmente en España.

Funciona con una valoración basada en la suma de puntos tomando diversas áreas de la vivienda, además del nivel de complejidad de la red de control. Con esto se ha definido tres niveles de automatización primarios, a pesar de que de ser necesario podrían considerarse otros más, estos son el Nivel 1, Nivel 2 y Nivel3, de los cuales se detalla a continuación:

- Nivel 1: Es cuando existen la mínima cantidad de elementos para que se reconozca a una aplicación como domótica. Suma un mínimo de 13 puntos repartidos en un mínimo de 3 aplicaciones.
- Nivel 2: Nivel medio, debe tener al menos 37 puntos repartidos en un mínimo de 3 aplicaciones diferentes.
- Nivel 3: Nivel alto de dispositivos, tienen un mínimo de 54 puntos y se caracteriza por poseer todas las aplicaciones disponibles.

Estos puntajes se obtienen mediante las valoraciones de aplicaciones y dificultad descritas en la siguiente tabla:

Tabla # 1: Niveles de automatización.

Aplicación	Dispositivos	Ponderación de la aplicación						Funcionalidades				
		No de dispositivos	Puntos	No de dispositivos	Puntos	No de dispositivos	Puntos	Confort	Seguridad	Ahorro energético	Comunicaciones	
Alarma de intrusión	Detectores de presencia.	2	1	1 cada 20 m ²	2	1 por estancia	3		X			
	Teclado codificado, llave electrónica o equivalente.	1	1	1	1	1	1		X			
	Sirena interior.	0		1	2	1	2		X			
	Contactos magnéticos de ventana y/o impactos.	En puntos de fácil acceso		1	En todas las ventanas	2	En todas las ventanas	2		X		
	Sistema de mantenimiento de alimentación en caso de fallo de suministro eléctrico.	0		Sí	2	Sí	2		X			
	Módulo de habla/escucha, destinado a la escucha en caso de alarma.	0		0		Sí	3		X			

	Sistema conectable con central de alarmas.	0	0	Sí	3	X	X	
Alarmas técnicas	Detectores de inundación necesarios en zonas húmedas	Los necesarios	1	Los necesarios	1	Los necesarios	1	X
	Electroválvula de corte de agua con instalación para bypass manual.	Las necesarias	1	Las necesarias	1	Las necesarias	1	X
	Detectores de concentraciones de gas butano y/o natural en zonas donde se prevea que habrá elementos que funcionen mediante gas.	Los necesarios	1	Los necesarios	1	Los necesarios	1	X
	Electroválvula de corte de gas con bypass manual.	Las necesarias	1	Las necesarias	1	Las necesarias	1	X
	Detector de incendios.	1 en cocina	1	1 cada 30 m ²	2	En todas las estancias	3	X
Simulación de presencia		No		Relacionada con las persianas motorizadas	2	Relacionada con las persianas motorizadas y con puntos de luz	3	X X
Videoportero		Sí	1	Sí	1	Sí	1	X
Control de persianas	Motorización y control de persianas.	Cuya superficie	1	Todas	2	Todas	2	X X

		sea mayor a 2 m ²							
Control de iluminación	Regulación lumínica con control de escenas.	No		En dependencias dedicadas al ocio	2	En salón y dormitorio	3	X	X
	En jardín o grandes terrazas mediante interruptor crepuscular o interruptor horario astronómico.	No		Sí	2	Sí	2	X	X
	Conexión / desconexión general de iluminación.	Sí	1	Sí	1	Sí	1		X
	Control de puntos de luz y tomas de corriente más significativas.	No		No		80% puntos de luz y 20% tomas de corriente	3	X	
Control de clima	Cronotermostato.	1 en salón	1	2, zonificando la vivienda en zona día y zona noche	2	Varios cronotermostatos, zonificando la vivienda por estancias	3	X	X
Programaciones	Posibilidad de realizar programaciones horarias sobre los equipos controlados (mínimo 12 programas).	No		Sí	2	Sí	2	X	X
Interfaz usuario	Consola o equivalente.	No		Sí	2	Sí	2	X	X

	Control telefónico bidireccional.	Sí	1	Interacción mediante SMS	2	Interacción mediante SMS	2	X
	Router para control por internet, WAP o equivalentes.	No		No		Sí	3	X
Red multimedia	Tomas SAT y tomas multimedia.	No		3 tomas de satélite + 3 tomas multimedia	2	3 tomas de satélite + 1 toma multimedia en todas las estancias, incluidas terrazas	3	X
	Punto de acceso Wi Fi.	No		Wi Fi	1	Wi Fi	1	X

1.5 ARQUITECTURAS DE SISTEMAS DOMÓTICOS

Para el manejo en sí de cada uno de los dispositivos presentes en un sistema domótico, sean estos actuadores, sensores, interfaces, etc., es necesario especificar las diferentes formas en las cuales pueden estar ubicados e interconectados física y lógicamente entre sí. Por lo cual para los sistemas domóticos se han definido las siguientes arquitecturas: Centralizada, Bus y Libre.

1.5.1 ARQUITECTURA CENTRALIZADA

Caracterizada por la existencia de un controlador centralizado o unidad central, es en donde cada elemento a supervisar y controlar estará conectado hasta tal controlador, el mismo que recibe la información de los sensores, la procesa y da las órdenes de control a los diferentes actuadores.



Fig. 8 Sistemas domóticos centralizados

Este tipo de arquitectura suele ser utilizada en obras domóticas de bajo nivel de automatización debido al bajo costo de los equipos controladores, la poca necesidad de recursos, el pequeño espacio físico a cubrir, sin embargo al ser sólo un dispositivo el encargado de procesar todo el control tiene la obvia limitación de dificultar ampliaciones futuras, aparte de siempre contar con la respectiva dependencia de la unidad central de control.

1.5.2 ARQUITECTURA DISTRIBUIDA:

Como su nombre lo indica, en esta arquitectura, la inteligencia del sistema o capacidad de procesamiento está distribuida por cada uno de los módulos presentes en el sistema, llamados nodos, que se ubican en diferentes lugares en un edificio, deben estar comunicados entre sí mediante un bus de comunicaciones en el que se transporta un protocolo con la información y órdenes pertinentes.

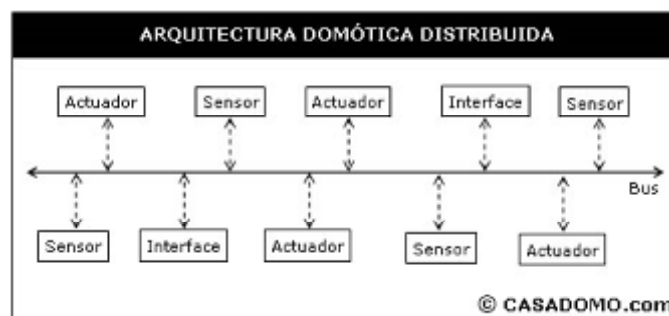


Fig. 9 Sistemas domóticos distribuidos

Los sistemas de control, en general, van tendiendo hacia el control distribuido, debido a la fácil escalabilidad, alta velocidad de respuesta y especialización de ciertos dispositivos, además de no depender de un sólo elemento de control. El caso de la domótica no es la excepción, ya que en las instalaciones domóticas, sobre todo en las de gran escala, se ha tendido a utilizar sistemas distribuidos por las múltiples ventajas que estos presentan, como por ejemplo la reducción del cableado, flexibilidad, robustez, facilidad de instalación, facilidad en el escalamiento, facilidad en el mantenimiento y reposición de nodos dañados. En contraparte, y a pesar de las ventajas que tiene este sistema, la principal desventaja que nos presenta es el aumento en el costo en relación a una arquitectura centralizada, ya que requiere equipos más costosos.

1.6 MEDIOS DE TRANSMISIÓN

Como es lógico los diferentes elementos de un sistema domótico requieren intercambiar información entre sí a través de un medio de transmisión. En la actualidad los medios de transmisión para sistemas domóticos más utilizados son las líneas de distribución eléctrica (power line), cable e inalámbrico.

1.6.1 LÍNEAS DE DISTRIBUCIÓN DE ENERGÍA ELÉCTRICA (CORRIENTES PORTADORAS)

El medio físico utilizado son las líneas de distribución de energía para la transmisión de datos. Este medio tiene una baja velocidad de transmisión. Este sistema no es el más recomendable para la transmisión de datos, ya que depende de la integridad de la señal de la red eléctrica, esta puede llegar a distorsionar la señal domótica y despistar al sistema, lo cual producirá que actúe de forma errónea. Sin embargo es una buena alternativa para instalaciones que necesiten un bajo coste ya que no requiere ningún cableado adicional. Es utilizado comúnmente en viviendas ya construidas.

Un sistema que tenga como base las corrientes de portadoras debe disponer de los siguientes elementos para su funcionamiento:

Unidad de control

Se encarga de gestionar el protocolo, almacenar las ordenes de los elementos de control y transmitirlos por la red para que sean recibidas por cada uno de los dispositivos a los que van dirigidas.

Interface

Elemento que recibe las ordenes de la unidad de control y las ejecuta.

Filtro

Limpia las señales que llegan a la unidad de control, protegen al sistema para que la red eléctrica no provoque daños en la integridad de la señal de control.

1.6.2 CABLE METÁLICO

La información se transmite por medio de ondas de corriente alterna de alta frecuencia, cubre distancias no muy grandes; el metal utilizado casi siempre es el cobre debido a su buena conductividad y coste razonable.

1.6.2.1 Par metálico.

Los cables formados por conductores de cobre pueden dar un amplio rango de variaciones en un entorno doméstico, ya que pueden transportar datos, voz, o servir para la alimentación de corriente. Existen diferentes combinaciones de los cables utilizados, lo cual se detalla a continuación.

- Cables formados por un conductor con aislamiento exterior plástico, comúnmente son utilizados para la transmisión de señales telefónicas.

- Un par de cables, cada uno de ellos posee un arrollamiento helicoidal de varios hilos de cobre, comúnmente se los usa en la distribución de señales de audio.
- Par apantallado, lo forman dos hilos recubiertos por un trenzado conductor en forma de malla cuya misión consiste en aislar las señales que circulan por los cables de las interferencias electromagnéticas exteriores, comúnmente se los puede ver utilizados para la distribución de sonido de alta definición o transporte de datos que necesiten confiabilidad en el envío – recepción de los mismos.
- Par trenzado, conformado por dos hilos de cobre recubiertos cada uno por un trenzado en forma de malla. Se utiliza el trenzado de los cables como un medio de protección ante las interferencias electromagnéticas que pueden recibir de fuentes próximas, el par trenzado es comúnmente utilizado para interconexión entre ordenadores.

Combinando estas configuraciones, se forman los siguientes tipos de cables:

- UTP (Unshielded Twisted Pair), par trenzado sin apantallamiento.
- STP (Shielded Twisted Pair), par trenzado apantallado, este cable es voluminoso debido a que está apantallado y es más costoso en cuanto a precio e instalación.

- FTP (Foil Twisted Pair) o ScTP (Screened Twisted Pair), que emplea una pantalla formada por papel aluminio en lugar de una malla de cobre para reducir el precio y el diámetro del cable.

1.6.2.2 Cable Coaxial.

Está formado por un núcleo de cobre rodeado por un material aislante que está cubierto por una pantalla de material conductor, según el tipo de cable y su calidad, la pantalla puede estar formada por una o dos mallas de cobre, un papel aluminio o ambos. Este material está recubierto por otra capa de material aislante.

El cable coaxial debe manipularse con cuidado evitando golpes o dobleces excesivos ya que esto provocaría una deformación en la malla, lo cual derivará en reducción del alcance del cable. Este tipo de cable permite el transporte de las señales de video y de datos a alta velocidad. Dentro del ámbito de la vivienda, el cable coaxial puede ser utilizado como soporte de transmisión para señales de TV y señales de control y datos a media y baja velocidad.

1.6.2.3 Conexiones Inalámbricas.

Las conexiones inalámbricas se caracterizan por la utilización del aire como medio de comunicación, por lo cual cuenta con ciertas ventajas y desventajas a la vez. Es idealmente utilizado como medio en edificaciones ya construidas en las que prácticamente es imposible la colocación de cables en la estructura de la vivienda. Sin embargo a esta gran ventaja se le enfrenta el alto costo de los equipos inalámbricos, problemas en cuanto a necesidad continua de baterías y por supuesto interferencias electromagnéticas que siempre se encuentran presentes en el ambiente (con excepción de los basados en tecnologías ópticas, como el infrarrojo).

Entre las principales tecnologías aquí utilizadas tenemos a los infrarrojos, la radiofrecuencia y el ya conocido bluetooth, sin embargo este último no es común en tecnologías domóticas, por lo cual no se hablará de él en este documento.

1.6.2.3.1 Infrarrojo

La utilización de mandos a distancia basados en transmisión por rayos infrarrojos se ha extendido ampliamente en el mercado residencial para tele comandar equipos de audio y video.

La comunicación se realiza entre un diodo emisor de luz en la banda de IR, sobre la que se superpone una señal, modulada con la información de control, y un fotodiodo receptor que extrae de la señal recibida la información de control. Los controladores de equipos domésticos basados en la transmisión de ondas en la banda de los infrarrojos, presentan gran comodidad y flexibilidad y admiten un gran número de aplicaciones.

Al tratarse de un medio de transmisión óptico es inmune a las interferencias electromagnéticas producidas por los equipos domésticos o por los demás medios de transmisión. Sin embargo, habrá que tomar precauciones en el caso de las interferencias electromagnéticas que pueden afectar a los extremos del medio.

Las desventajas mayores que posee este medio de transmisión son: La poca distancia de comunicación de los dispositivos a

comunicar, ya que deben estar a 1 m, en algunos casos usando emisores especiales se llega a 10 m pero requieren de un ángulo de visión de 17° a 30°; las bajas velocidades de comunicación del sistema, ya que el sistema IrDA 1.0 llega a velocidades de 115.2 Kb/s y el IrDA 1.1 a 4Mb/s.

Tabla # 2: Espectro de frecuencias

Tipo de radiación	Frecuencia	Longitud de onda
Ionizante	> 3.000 THz	< 100 nm
Ultravioleta	3.000 – 750 THz	100 – 400 nm
Visible	750 – 385 THz	400 – 780 nm
Infrarroja (IR)	385 – 0,3 THz	0,78 – 1.000 μ m
Microondas	300 – 0,3 GHz	1 – 1.000 mm
Radiofrecuencia (RF)	300 – 0,1 MHz	1 – 3.000 m
Bajas frecuencias (ELF)	300 Hz - 0	~ 5.000 Km.

1.6.2.3.2 Radiofrecuencia

La introducción de las radiofrecuencias como soporte de transmisión en la vivienda ha estado acompañada de la proliferación de los teléfonos inalámbricos y de sencillos telemandos. Este medio de transmisión puede parecer, en principio, idóneo para el control a distancia de los sistemas domóticos, dada la gran flexibilidad que supone su uso, sin embargo, resulta particularmente sensible a las perturbaciones electromagnéticas producidas, tanto por los medios de transmisión, como por los equipos domésticos.

Su principal ventaja es la fácil interceptación de las comunicaciones.

Entre las desventajas a tomarse en cuenta en estos sistemas tenemos a la alta sensibilidad a las interferencias y la dificultad para la integración de las funciones de control y comunicación, en su modalidad de transmisión analógica.

1.7 TOPOLOGÍAS

La topología de red se define como la distribución física de los elementos de control con respecto al medio de comunicación. Las principales topologías utilizadas son la de estrella, en bus y libre o híbrida.

1.7.1 TOPOLOGÍA DE ESTRELLA

Todos los nodos están conectados a un nodo central. Todos los nodos periféricos se pueden comunicar con los demás transmitiendo o recibiendo del nodo central solamente. Un fallo en la línea de conexión de cualquier nodo con el nodo central provocaría el aislamiento de ese nodo respecto a los demás, pero el resto de sistemas permanecería intacto.

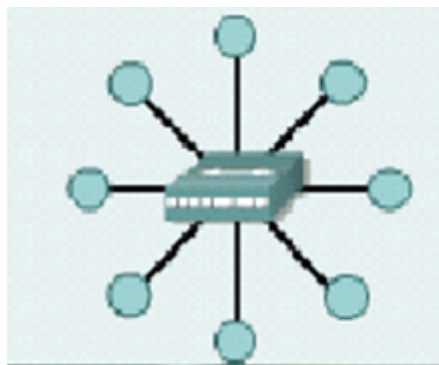


Fig. 10 Topología de estrella

1.7.2 TOPOLOGÍA DE BUS

Los datos de entrada y salida de los sensores y actuadores se transmiten por una misma línea de transmisión, además se transmiten todo tipo de datos como los datos de parametrización, datos de diagnóstico y programas de aplicación. Se requiere de dos terminaciones de red en los extremos del bus para que no se produzcan reflexiones de onda dentro del cable que generen tráfico de red debido a las impedancias adaptadas.

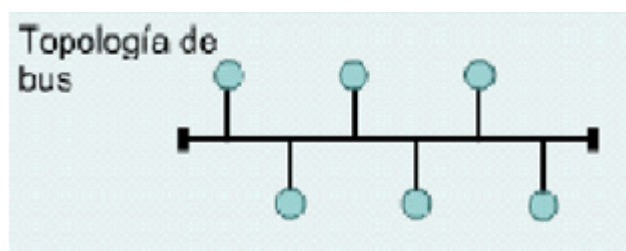


Fig. 11 Topología de bus

1.7.3 TOPOLOGÍA HÍBRIDA

Usa una combinación de dos o más topologías distintas de tal manera que la red resultante no tiene una forma estándar. Por ejemplo, una red en estrella conectada con una red en bus. Una topología híbrida, siempre se produce

cuando se conectan dos topologías de red básicas. En sistemas inmóticos de gran escala es muy probable encontrarse con este tipo de topología debido al tamaño de la obra que genera la necesidad de acoplarse a una determinada topología de acuerdo a la propiedad. Los fabricantes emplean transceivers capaces de procesar las reflexiones de onda que se generen en las topologías llamadas híbridas o libres.

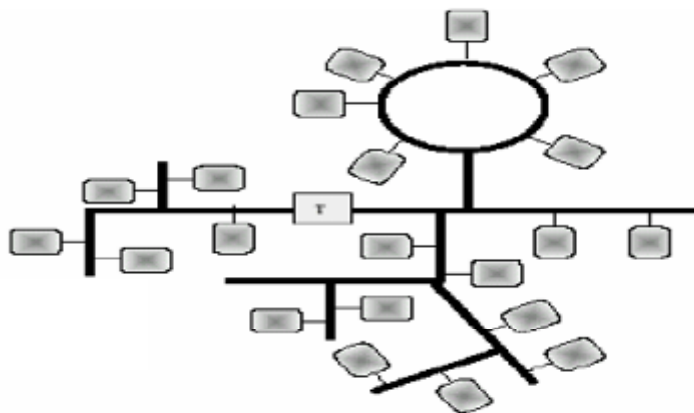


Fig. 12 Topología libre o mixta

1.8 PROTOCOLOS DOMÓTICOS

Con el desarrollo de las redes domésticas se ha permitido la aparición de una serie de tecnologías y protocolos, los cuales pueden ser propietarios o estándares.

1.8.1 SISTEMAS CON PROTOCOLOS PROPIETARIOS

Son aquellos que únicamente permiten la comunicación entre elementos del propio fabricante y por tanto son exclusivos de éste, es decir que los nodos de control en el sistema utilizan un protocolo de comunicaciones cerrado,

creado con el propósito de comunicar únicamente los productos de dicho fabricante.

Los equipos que utilizan este tipo de protocolos suelen ser menos costosos pero al utilizar un protocolo propietario existe una dependencia con el fabricante de los equipos, lo que podría desencadenar problemas como altos costes de mantenimiento o falta de repuestos en caso de cierre de las empresas fabricantes.

Entre los principales protocolos propietarios que se desarrollaron podemos mencionar a Cebus, Modbus, Bus-CAN, BUSing, Dupline, xAP, xPL, etc., capaces de realizar las funciones necesarias para el control domótico. En la actualidad existen protocolos como Honeywell, Bacnet, CeBus.

1.8.2 SISTEMAS CON PROTOCOLOS ESTÁNDARES O ABIERTOS

Son aquellos que permiten la interconexión de elementos de distintos fabricantes que utilizan el mismo protocolo de comunicación. Los protocolos estándares han ido evolucionando al pasar del tiempo, permitiendo un ambiente competitivo y participativo para usuarios, proyectistas, integradores y fabricantes.

Los sistemas con protocolos abiertos tienen un bajo coste de mantenimiento, y al no ser de un sólo fabricante se produce una mayor variedad de equipos alrededor del mundo, asegurándose así la calidad y existencia del producto.

Algunos de los protocolos abiertos más importantes son Konex - EIB, LonWorks, ZigBee y X-10.

1.8.3 COMPARACIÓN ENTRE PROTOCOLOS

Como se mencionó anteriormente los protocolos propietarios requieren que todos los equipos utilizados en la instalación domótica sean del mismo fabricante, ya que no permiten la integración con dispositivos de otras casas fabricantes. Por su parte, los protocolos estándares permiten mayor facilidad de integración entre equipos de diferentes fabricantes.

En cuanto a mantenimiento de los sistemas, en caso de tener uno implementado con protocolos propietarios se depende de los equipos del mismo fabricante y sus respectivos costos, sin la libertad de realizar las modificaciones necesarias o cambios con otros equipos presentes en el mercado, aun cuando estos tengan costos más bajos o un mejor ajuste a los requerimientos del usuario. No es igual caso de los sistemas con protocolos abiertos que permiten mayor versatilidad e integración.

En cuanto a costos de los equipos, el de aquellos que usan protocolos abiertos suele ser más elevado que el de los que trabajan con protocolos propietarios.

En la siguiente tabla se presentan parámetros comparativos entre diferentes tipos de tecnologías utilizadas en domótica:

Tabla # 3: Principales características de las tecnologías de redes domésticas

Tecnología	Medio de Transmisión	Velocidad de Transmisión	Distancia máxima al dispositivo
IEEE 1394	UTP / FO	<ul style="list-style-type: none"> • 400 Mbps (v.a) • 3.2Gbps (v.b) 	<ul style="list-style-type: none"> • 4.5 m / 70 m
USB	USB	<ul style="list-style-type: none"> • 12 Mbps (v. 1.1) • 480 Mb/s (v.2) 	<ul style="list-style-type: none"> • 5 m
Konnex	<ol style="list-style-type: none"> 1. TP0 2. TP1 3. PL100 4. PL132 5. Ethernet 6. Radio 	<ol style="list-style-type: none"> 2. 9600 bps 3. 1200/2400 bps 4. 2.4 Kbps 	<ol style="list-style-type: none"> 2. 1000 m 3. 600 m
Lonworks	<ol style="list-style-type: none"> 1. TP 2. Cable eléctrico 3. Radio 4. Coaxial 5. FO 	<ol style="list-style-type: none"> 1. 78 Kbps – 1.28 Mbps 2. 5.4 Kbps 	<ol style="list-style-type: none"> 1. 500 – 2700 m
X10	Cable eléctrico	<ul style="list-style-type: none"> • 60 bps en EEUU • 50 bps en Europa 	<ul style="list-style-type: none"> • 185 m²
BacNet	<ul style="list-style-type: none"> • Cable Coaxial • TP • FO 	<ul style="list-style-type: none"> • 1 Mbps – 100 Mbps 	<ul style="list-style-type: none"> • Con Ethernet sobre TP: 100 m
EIB	<ol style="list-style-type: none"> 1. TP 2. Cable eléctrico 3. RF 4. Infrarrojos 	<ol style="list-style-type: none"> 1. 9600 bps 2. 1200/2400 bps 	<ol style="list-style-type: none"> 1. 1000 m 2. 600 m 3. 300 m
EHS	<ol style="list-style-type: none"> 1. Cable eléctrico 2. TP 	<ol style="list-style-type: none"> 1. 2.4 Kbps 2. 48 Kbps 	
Batibus	<ul style="list-style-type: none"> • TP 	<ul style="list-style-type: none"> • 4800 bps 	<ul style="list-style-type: none"> • 200 m a 1.500 m en función de la sección de cable
Cebus	<ul style="list-style-type: none"> • TP • Cable eléctrico • Radio • Coaxial • Infrarrojos 	<ul style="list-style-type: none"> • 10.000 bit/s 	<ul style="list-style-type: none"> • En función de las características del medio
DALI	<ul style="list-style-type: none"> • Par de cable 	-----	<ul style="list-style-type: none"> • 200 m
Metasys	<ul style="list-style-type: none"> • N2 Bus 	<ul style="list-style-type: none"> • 9600 bps 	<ul style="list-style-type: none"> • 1219 m
SCP	<ul style="list-style-type: none"> • Cable eléctrico 	<ul style="list-style-type: none"> • <10 Kbps 	-----
Ethernet	<ul style="list-style-type: none"> • UTP / FO 	<ul style="list-style-type: none"> • 100Mbps / 1 Gbps 	<ul style="list-style-type: none"> • 100 m / 15 Km
HomePlug	<ul style="list-style-type: none"> • Cable eléctrico 	<ul style="list-style-type: none"> • 14 Mbps 	<ul style="list-style-type: none"> • 650 m²
HomePNA	<ul style="list-style-type: none"> • Línea telefónica 	<ul style="list-style-type: none"> • 10 Mbps 	<ul style="list-style-type: none"> • 304.8 m • 929 m²
IEEE 802.11	<ul style="list-style-type: none"> • Inalámbrico 	<ul style="list-style-type: none"> • 54 Mbps (v.a y v.g) • 11 Mbps (v.b) 	<ul style="list-style-type: none"> • 33 m (v.a) • 100 m (v.b)
Bluetooth	<ul style="list-style-type: none"> • Inalámbrico 	<ul style="list-style-type: none"> • 1 Mbps (v. 1) • 10 Mbps (v. 2) 	<ul style="list-style-type: none"> • 10 m (v.1) • 100 m (v.2)
HiperLAN/2	<ul style="list-style-type: none"> • Inalámbrico 	<ul style="list-style-type: none"> • 54 Mbps 	<ul style="list-style-type: none"> • 100 m
IRDA	<ul style="list-style-type: none"> • Inalámbrico 	<ul style="list-style-type: none"> • 9600 bps – 4 Mbps 	<ul style="list-style-type: none"> • 2 m
Home RF	<ul style="list-style-type: none"> • Inalámbrico 	<ul style="list-style-type: none"> • 10 Mbps 	<ul style="list-style-type: none"> • 38 m
ZigBee	<ul style="list-style-type: none"> • Inalámbrico 	<ul style="list-style-type: none"> • 20 Kbps-250Kbps 	<ul style="list-style-type: none"> • 10 m – 75 m
GSM	<ul style="list-style-type: none"> • Inalámbrico 	<ul style="list-style-type: none"> • 9600 bps 	-----

1.9 TECNOLOGÍA LonWorks.

LonWorks tiene una relevancia especial dentro de los sistemas domóticos debido a que posee una arquitectura abierta y una implementación de sistemas domóticos a pequeña y gran escala, lo cual abarca desde casas, aeropuertos, rascacielos, hasta complejos deportivos, entre muchas obras más.

1.9.1 CONCEPTOS.

LonWorks es una plataforma de control creada por la compañía norteamericana Echelon, que como creadora del protocolo y sistema LonWorks ofrece herramientas de desarrollo, gestión y configuración de aplicaciones y redes.

Como respuesta a la imperativa necesidad de definir estándares y marcos para que LonWorks sea un sistema abierto pero estandarizado surge la asociación LonMark.

LonMark es la asociación de usuarios de la tecnología Lon, siendo creada por las empresas líderes en los diferentes sectores de la aplicación de la tecnología Lon tales como; domótica, inmótica, control industrial y de transporte.

A continuación se hará referencia a LonMark España:

“A LonMark España puede pertenecer todo aquel que esté interesado en el conocimiento, difusión y utilización de la tecnología LON: fabricantes, integradores, instaladores, prescriptores, centros docentes y usuarios.

Actualmente, son 8 las empresas que forman parte de la Asociación como Miembros de Pleno Derecho:

- EBV Elektronik GmbH
- Echelon Europe Ltd
- ERCO Iluminación
- ISDE.
- Schneider Electric
- TEMPER Clima
- e-Controls
- Unitronics

Y otras 5 empresas como Miembro Asociado:

- Aditel
- Dinycon Sistemas
- Kamstrup España
- SCS
- Future Technologies, SL.³

El protocolo LonTalk implementa las siete capas del modelo OSI, y lo hace usando una mezcla de hardware y firmware sobre un chip de silicio, evitando cualquier posibilidad de modificación. Se incluyen características como gestión acceso al medio, reconocimiento y gestión punto a punto, y servicios más avanzados tales como autenticación de remitente, detección de mensajes duplicados, colisión, reintentos automáticos, soporte de cliente-servidor, transmisión de tramas no estándar, normalización e identificación de tipo de dato, difusión unicast / multicast, soporte de medios mixtos y detección de errores.

³ España, L. (s.f.). *LonMark España*. Recuperado el 26 de 07 de 2010, de LonMark España: http://www.lonmark.es/que_es_asociacion.asp

Las redes de control LonWorks pueden variar en tamaño desde 2 hasta 32.000 dispositivos y se pueden usar en cualquier aplicación desde supermercados a plantas de petrolíferas, desde aviones hasta ferrocarriles, desde medición por láser a máquinas de mecanizado, desde rascacielos a viviendas particulares.

1.9.2 ESTRUCTURA.

Una red LonWorks está compuesta por un conjunto de elementos inteligentes denominados nodos, los cuales se comunican entre sí mediante el protocolo LonTalk, que está basado en el modelo OSI, empleando diversos medios físicos. Además, existen también elementos específicos para encaminamiento por la red, los cuales se llaman ruteadores, y se dispone de múltiples medios de transmisión e interfaces.

En una red LonWorks los subsistemas básicos son:

- Controladores Neuron Chip y firmware asociado, el mismo que incluye soporte para protocolo LonTalk.
- Transceivers que permiten la conexión a los distintos medios físicos de transmisión.
- Módulos de control LonWorks que incluyen al Neuron Chip, transceivers y memoria externa.
- Routers que sirven de encaminadores entre diferentes subredes y/o medios de transmisión.

- Interfaces de red, mediante los cuales se pueden interconectar otros elementos no basados en el protocolo LonTalk como pueden ser: ordenadores personales, PLCs, GSM, etc.
- Herramientas de instalación, gestión y diagnóstico tanto de los nodos como de la red de control: DDE server, LonMaker.
- Herramientas de desarrollo de aplicaciones, compilador Neuron C, enlazador, depuradores, etc.

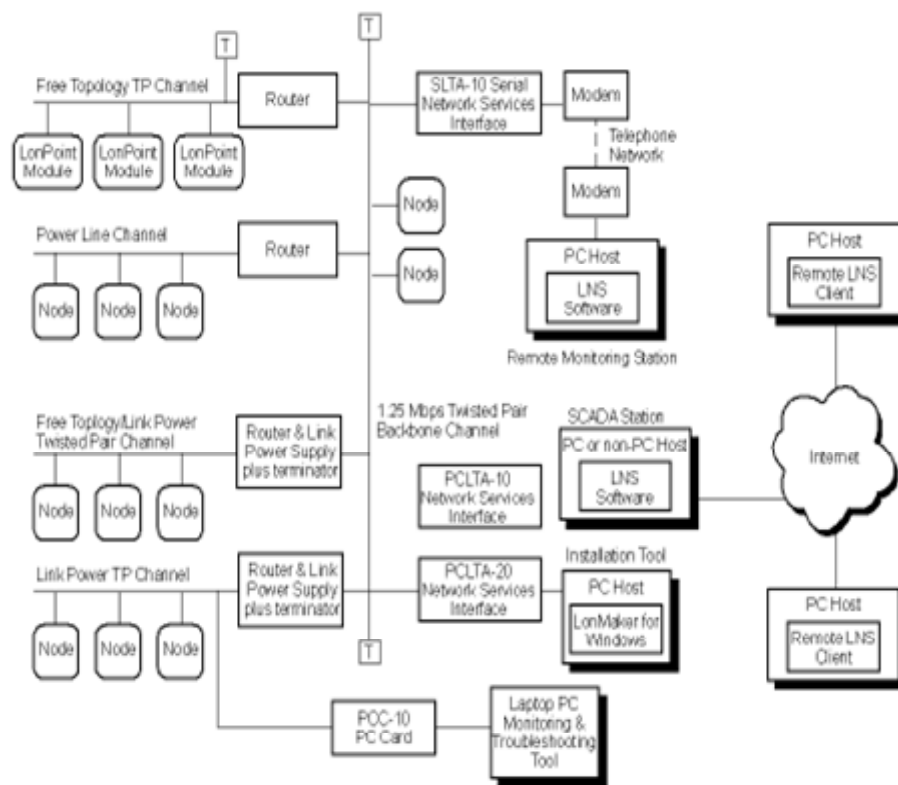


Fig. 13 Estructura de una red LonWorks

Existen nodos de control con memoria y capacidad de procesamiento para calcular la ley de control y nodos simples que funcionan como transductores / actuadores leyendo variables de los transductores adecuados y enviándolas a través de variables de red al nodo inteligente.

Es posible utilizar routers para aislar zonas de la red con mayor tráfico y crear subredes.

1.9.3 HARDWARE Y SOFTWARE. NEURON CHIP.

El Neuron Chip fue diseñado por Echelon para proporcionar un elemento capaz de, de manera similar a las neuronas de un cerebro, intercambiar información entre dispositivos conectados, cada uno con funciones específicas para las que fue diseñado. El Neuron Chip integra las seis capas inferiores del modelo OSI del protocolo LonTalk; es decir desde la física hasta la de presentación, lo que permite al fabricante desarrollar cualquier aplicación utilizando el interfaz de presentación de que dispone y preocupándose de cualquier tarea relacionada con la implementación del sistema LonWorks.

Internamente, un Neuron está constituido por tres microprocesadores en uno. Dos microprocesadores están optimizados para ejecutar el protocolo de comunicaciones, y el tercero a ejecutar el programa de control del nodo. De esta manera se asegura que la complejidad del programa no afecta negativamente a la respuesta de la red y viceversa. Además, el hecho de encapsular ambas funciones; comunicación y aplicación, en un solo chip ahorra tiempos de diseño y producción.

Más específicamente los tres procesadores son:

- CPU de acceso al medio, que controla la E/S de información a través del puerto de comunicaciones; implementa las capas OSI uno y dos.
- CPU de control de red, encargada de las funciones destinadas al control de conexiones, direccionamiento de mensajes conforme a la estructuración de la red, temporización de diálogos, estructuración de datos, etc. Implementa las capas OSI tres a la seis.
- CPU de aplicación, que ejecuta el programa implantado por el fabricante y que determina la funcionalidad final del dispositivo.

La memoria principal se encuentra distribuida en forma de:

- Memoria RAM, necesaria para la ejecución de las aplicaciones.
- Memoria de sólo lectura ROM donde se encuentra el firmware del Neuron Chip: sistema operativo y funciones que implementan el protocolo LonTalk.
- Memoria EEPROM destinada a guardar los datos de configuración que se ponen a disposición del usuario del dispositivo, así como el programa de aplicación para que éste pueda ser actualizado por ejemplo a través de la red.

En la siguiente figura se puede observar el diagrama de bloques de los elementos internos en un Neuron Chip.

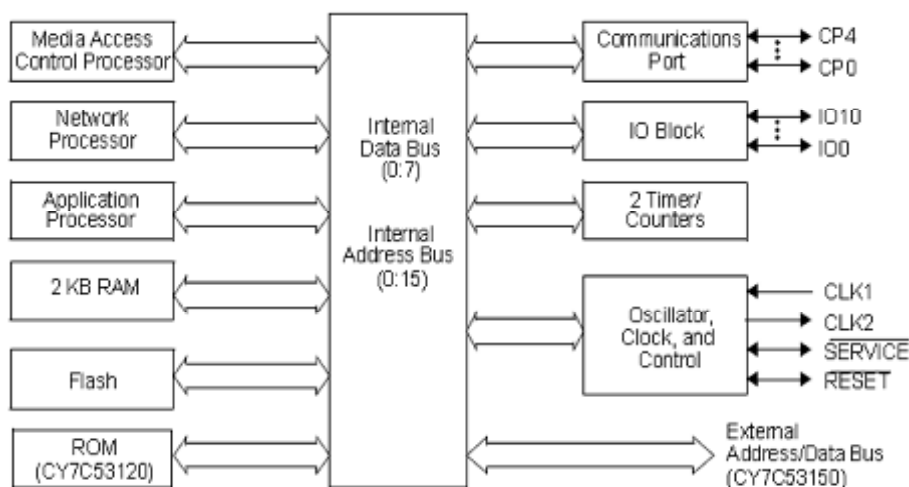


Fig. 14 Diagrama de bloques interno de un Neuron Chip

En caso de que el programa de aplicación sea demasiado extenso o complicado y requiera de capacidades superiores de velocidad, procesamiento y memoria, el Neuron Chip dispone de un interfaz de bus paralelo de alta velocidad, al que se le puede conectar un procesador más potente. Si este fuera el caso, el Neuron chip realizaría únicamente las funciones de las 6 primeras capas del modelo OSI, es decir, se encargaría de las funciones de red y el procesador anexo ejecutaría el programa de aplicación.

Todos los Neuron Chips disponen de un identificador único de 48 bits llamado Neuron-ID, que es implantado en el proceso de fabricación. El mismo que permite identificar inequívocamente un dispositivo de una red al momento de conectarlo; sin embargo, debido a que se puede implementar un complejo nivel de direccionamiento en las redes LonWorks, cuando el dispositivo forma parte de una red basada en dominios, el Neuron-ID deja de ser utilizado como dirección de dispositivo, pasando a utilizar una dirección de red.

En la figura siguiente se puede observar el Neuron ID de un determinado nodo.



Fig. 15 Neuron ID

Los programas de aplicación que definen le funcionalidad del dispositivo se escriben en el lenguaje Neuron C, que está basado en ANSI C con determinadas características que le hacen más apropiado para aplicaciones basadas en intercambio de información a través de redes. Éste programa dispone de tres importantes extensiones:

- Sentencia when, para introducir eventos y definir el orden de ejecución del programa. Difiere de las aplicaciones estructuradas comunes en C en que las aplicaciones en este caso están dirigidas a eventos, que ocurren en la red o en el propio dispositivo.
- 37 tipos nuevos de datos, 35 objetos de E/S y 2 objetos temporizadores.
- Funciones para el intercambio de información en forma de mensajes y variables de red.

En la red, cada dispositivo incorpora un transceptor o transeiver, que proporciona un interfaz físico de comunicación del dispositivo con la red LonWorks. Dos dispositivos con diferente transceptor pueden comunicarse entre sí, pero es necesario interponer un componente enrutador, esto permite

poder definir segmentos de redes diferenciados por el medio físico utilizado: subredes basadas en enlaces de radiofrecuencia, fibra óptica, par de hilos, etc.

Cada dispositivo de la red incorpora el transceptor y el Neuron Chip, aunque dependiendo de la aplicación a la que se destine, puede incluir sensores, actuadores, interfaces con sensores y/o actuadores de terceros, interfaces de comunicación serie o paralelo, etc.

1.9.4 APLICACIONES.

Entre las aplicaciones básicas de la tecnología LonWorks se pueden destacar:

- Ahorro Energético.
- Nivel de Confort.
- Seguridad.
- Comunicaciones.

A continuación se mencionan ejemplos de casos en los que se ha utilizado la tecnología LonWorks en la domotización de lugares.

1.9.4.1 Ahorro energético.

Uno de los lugares donde se ha implantado la tecnología LonWorks es la plaza de toros de Navacarnero con capacidad para 7500 personas.

La singularidad de esta plaza recae en el hecho que su iluminación, climatización y ventilación dispone de un sistema de control implantado por Electromontajes Mendiola con los productos del fabricante ISDE, bajo el protocolo abierto e interoperable LonWorks ®.

La plaza de toros de Navalkanero supondrá un ahorro energético superior al 35% en comparación a un recinto de similares características que no cuente con un sistema de control distribuido. La capacidad de controlar los tiempos de encendido de los equipos de climatización junto con un mayor aprovechamiento de la luz natural son los responsables de gran parte de este ahorro, que aparte de beneficiar al propio municipio, contribuye ecológicamente reduciendo considerablemente las emisiones de CO2 al medio ambiente.”⁴



Fig. 16 Plaza de toros de Navalkanero

⁴ ISDE. (12 de Marzo de 2008). *Proyectos Domótica*. Recuperado el 07 de Junio de 2010, de Proyectos Domótica: <http://www.proyectosdomotica.com/articulos-domotica.php?hogar-digital=164>

1.9.4.2 Iluminación

“La Villa Olímpica en Beijing utilizó tecnología de LonWorks para crear un sistema de control de iluminación que integró todos los subsistemas para optimizar el uso de energía, manteniendo al mismo tiempo un lugar seguro y de agradable ambiente para los atletas.”⁵

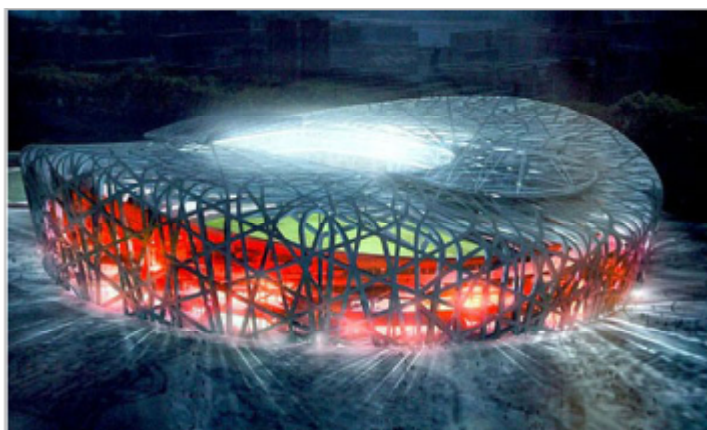


Fig. 17 Villa Olímpica de Beijing

Dentro del ámbito de la iluminación pueden realizarse acciones como; apagado general de todas las luces de la vivienda, automatización del apagado/ encendido en cada punto de luz, regulación de la iluminación según el nivel de luminosidad ambiente.

Gracias a la utilización de sensores, las luces se encenderán solamente cuando sean realmente requeridas, como por ejemplo al entrar en las habitaciones. Un detector de presencia se encarga de encenderlas. También se podrá regular la intensidad de luz o encender a intervalos los diferentes interruptores de la casa.

⁵ Iluminet. (24 de Agosto de 2008). *ILUMINET*. Recuperado el 07 de Junio de 2010, de ILUMINET: <http://www.iluminet.com.mx/articulos-destacados/sobre-la-iluminacion-en-beijing/>

1.9.4.3 Control a distancia

El control remoto potencia el desarrollo de nuevos servicios en el hogar. Hoy ya es posible controlar cualquier propiedad domótica mediante Internet o mensajes SMS. La aplicación de las tecnologías de la información al hogar será un aspecto básico y crucial para distintos sectores:

1.9.4.4 Aparatos domésticos

La domótica utiliza la electrónica para controlar el funcionamiento y activación de los dispositivos presentes en el hogar tales como la lavadora o el lavaplatos en tiempos en que hay una disminución en la tarifa de la energía.

1.9.4.5 Climatización

Mediante la implementación domótica adecuada, se puede tener un control total de climatización en una casa, inclusive en cada estancia mediante la programación. Existen aplicaciones como bajar la temperatura interna si la temperatura en el exterior de la vivienda aumenta. Se puede controlar además este tipo de aplicaciones a distancia con la utilización de SMS o llamadas.

1.9.4.6 Equipos de audio y video

Se puede controlar el ajuste adecuado sea para que se goce de una calidad de audio y video adecuada para un auditorio de modo que el usuario pueda ajustar estos equipos de acuerdo a sus preferencias en cualquier lugar de la casa.

1.9.4.7 Sistemas de vigilancia y seguridad

“Los más sofisticados sistemas de protección de la vivienda contra robo, catástrofes e incendios también pueden ser inteligentes. La notificación puede realizarse incluso a través del correo electrónico. Una cámara de televisión o una webcam permiten controlar un inmueble desde cualquier lugar del mundo. Con la video vigilancia remota se puede visualizar visualice un lugar desde un PC o PDA en cualquier navegador de Internet, incluso desde algunos teléfonos móviles. También se puede automatizar el accionamiento de rejas protectoras y cierres a las horas deseadas.”⁶

1.10 HMI

1.10.1 CONCEPTOS

Es un acrónimo en idioma inglés, cuyas siglas representan la frase: (Human Machine Interface), traducido al español como interfaz humano máquina, utilizado para referirse a una interacción entre el ser humano y un sistema usualmente de automatización de procesos. Se debe considerar que el diccionario de la Real Academia de la Lengua Española define la palabra interfaz como: “Conexión física y funcional entre dos aparatos o sistemas independientes”. Es conocido además que la palabra interfaz proviene de la palabra inglesa interface, formada del prefijo inter (entre) y face (cara) significando en forma etimológica, “entre caras” o también entendida como “superficie de contacto”.

⁶REPSOL. (31 de 08 de 2005). *Repsol*. Recuperado el 26 de 07 de 2010, de Repsol: http://www.repsol.com/es_es/casa_y_hogar/sin_salir_de_casa/reportajes/domotica/domotica.aspx

Un HMI, también es conocido como: “interfaz de usuario”, siendo el medio con el que el usuario puede comunicarse con una máquina o sistema, comprende todos los puntos de contacto entre el usuario y el equipo. También hace referencia al conjunto de métodos para lograr interactividad entre un usuario y una computadora. Una interfaz puede ser del tipo GUI (Interfaz Gráfica de Usuario), línea de comandos, etc. También puede ser a partir de un hardware, por ejemplo, el monitor, el teclado y el mouse, son interfaces entre el usuario y el ordenador.

1.10.1.1 Evolución de la Interacción Hombre Máquina

Marshall McLuhan; un educador, filósofo y estudioso canadiense, profesor de teoría de la comunicación, clasifica las eras de la humanidad de tal forma que se llega a determinar lo que es conocido actualmente como Interacción Hombre Máquina, estas son:

- La era preliteraria o tribal.
- La era de Imprenta.
- La era Electrónica.

En la era preliteraria o tribal, él se refiere a un tiempo en el que la comunicación era realizada de un modo verbal donde se vivía en un espacio acústico con sonidos asociados a objetos.

En la era de la Imprenta se inicia un segundo estadio de la civilización con la creación de la escritura, llegando a un estado más racional y funcional requiriendo una comprensión uniforme, concatenada y continua. La escritura provocó el espacio del conocimiento en algo limitado, lineal, estructurado, sistemático y visual.

La era electrónica o retribalización viene marcada por la aparición de medios tecnológicos en la comunicación. La aparición de la radio que McLuhan considera como una extensión del oído y la televisión como extensión del tacto restituyen al individuo a una relativa totalidad de sus sensaciones, además la tecnología electrónica no es dependiente de las palabras transmitidas oralmente, y si se considera a la computadora como extensión del sistema nervioso central habría como extender la conciencia.

McLuhan concebía cualquier tipo de tecnología como una extensión del cuerpo, mente y ser por ejemplo: la ropa como una extensión de la piel la bicicleta como una extensión de los pies, etc. Así como la computadora sería una extensión del sistema nervioso central o cerebro. Las consecuencias de la tecnología serían derivadas al orden social y natural del hombre, alcanzando la conciencia su conciencia conceptos más allá de los contemporáneos.

De esta forma va haciéndose cada día más común y necesaria una interacción del ser humano con las computadoras, y las máquinas en general, provocando que se trate de diseñar interfaces que hagan el manejo de todo dispositivo de control

fácilmente accesible para mayorías. El HMI viene a ser un software con la función de integrar al ser humano al sistema.

1.10.2 FUNCIONES DE UN SOFTWARE HMI

El ser humano está constantemente interactuando con los objetos que le rodean, y tiene expectativas del comportamiento de los mismos en base a experiencias anteriores. En cuanto a las funciones del software HMI están las siguientes:

- Monitoreo de la planta respectiva en tiempo real para poder obtener y mostrar los datos que se generen, de forma que se permita una lectura más fácil de interpretar, sea números texto o gráfico.
- Supervisión de las condiciones de trabajo del proceso y su respectivo ajuste de forma directa desde la computadora.
- Alarmas basadas en límites de control preestablecidos. Alarma se define como la capacidad de reconocer eventos excepcionales dentro del proceso y reportar estos eventos.
- Control: Capacidad de aplicar algoritmos en el proceso para mantener los valores de las variables del proceso en límites establecidos, hay que considerar que si se realiza el control únicamente desde una PC puede estar limitado en base a la capacidad de la computadora y su funcionamiento estaría condicionado a la conexión con la PC.
- Capacidad de mostrar y almacenar archivos históricos del proceso con determinada frecuencia de almacenamiento, que servirá para verificar el funcionamiento del proceso y su comportamiento a lo largo de cierto

intervalo de tiempo, siento una herramienta poderosa para poder optimizar y corregir el proceso.

Las tareas de un Software de Supervisión y Control son:

- Permitir una comunicación con dispositivos de campo.
- Actualizar una base de datos dinámicamente con las variables del proceso.
- Visualizar las variables mediante pantallas con objetos animados (mímicos).
- Permitir que el operador pueda enviar señales al proceso, mediante botones, controles ON/OFF, ajustes continuos con el mouse o teclado.
- Supervisar niveles de alarma y alertar/actuar en caso de que las variables excedan los límites normales.
- Almacenar los valores de las variables para análisis estadístico y/o control.
- Controlar en forma limitada ciertas variables de proceso.

1.10.3 MODELO DE DISEÑO DE UNA INTERFAZ HMI

Cuando se diseña una interfaz humano-máquina, se requiere crear un modelo de diseño, uno de usuario, la percepción del usuario del sistema y una imagen del sistema. Debe tomarse en cuenta que el interfaz diseñado

debe obtener un equilibrio en estos modelos para obtener un interfaz consistente.

El modelo de diseño del sistema incorporará representaciones arquitectónicas, de procedimientos de software, de interfaces, dispositivos.

El modelo de usuario muestra las características de los usuarios finales del sistema. El diseño debe comenzar conociendo el perfil de los usuarios a los que va dirigido, nivel de conocimientos, edad, sexo, capacidades físicas. Los usuarios pueden estar en una de las siguientes categorías:

- Novatos, sin conocimientos del sistema, escaso conocimiento de la aplicación o de la computadora en general.
- Usuarios esporádicos, con conocimiento razonable de la aplicación, pero que recuerda vagamente la información para usar la interfaz.
- Usuarios frecuentes, buenos conocimientos de la aplicación, buen manejo de la información.

La imagen del sistema combina el aspecto y sensación de la interfaz, con toda la información que proporciona el sistema. Cuando coinciden la imagen del sistema y la percepción del sistema, los usuarios se sienten cómodos con el software y lo utilizan eficazmente. Para conseguir esta fusión de modelos, el modelo de diseño debe desarrollarse para acomodar la información sintáctica y semántica sobre la interfaz.

Los modelos de diseño de un HMI son abstracciones de lo que está haciendo el usuario o de lo que piensa que debería estar haciendo cuando usa una aplicación. De esta manera se logra facilitar el manejo y la interacción del

usuario con un sistema determinado para que tenga una sensación de controlar y monitorear cada parte intuitivamente y con facilidad.

1.10.4 HMI EXISTENTES EN EL MERCADO

Existen una gran variedad de HMI presentes en el mercado, los mismos que están compuestos por un conjunto de programas y archivos. Hay programas para diseño y configuración del sistema y otros que son el motor mismo del sistema.

Entre los tipos de software de supervisión y control para computadora se tiene:

- Lenguajes de programación visual tales como Visual C++ o Visual Basic, los mismos que se utilizan para desarrollar software HMI de acuerdo a la necesidad específica del usuario. Una vez que se ha desarrollado el software se genera un programa de aplicación transparente para el usuario, que no tiene la posibilidad de ser re-programado.

- Paquetes de desarrollo orientados a tareas HMI que pueden ser empleados para desarrollar un HMI acorde a la necesidad específica del usuario, y además también puede ser utilizado para ejecutar un HMI desarrollado para el usuario. El usuario puede re-programarlo si tiene el software utilizado. Entre los más usados están: FIX Dynamics, Wonderware, PCIM, Factory Link, WinCC, LabView, etc.

Ha de considerarse que para trabajar desde un sistema operativo con una red se requiere un protocolo de comunicación entre aplicaciones, para el intercambio de datos. A continuación se dará una breve descripción de algunos de los softwares continuamente utilizados para supervisión y control:

1.10.4.1 Wonderware

El software de Wonderware facilita el diseño, construcción, despliegue y mantenimiento de aplicaciones seguras y estandarizadas para las operaciones de fabricación e infraestructura. Las soluciones de software HMI de Wonderware son de las más populares en el mercado mundial de automatización y control. De la página de Wonderware para América latina ellos describen las Soluciones de Software HMI/SCADA de Wonderware de la siguiente forma:

“Las soluciones HMI/SCADA a menudo imponen demandas complejas a las arquitecturas de software. InTouch HMI Visualization de Wonderware, combinado con la premiada Wonderware System Platform, basada tecnología ArchestrA, se encuentra posicionado de manera única para superar estos retos.

Las soluciones construidas sobre la tecnología ArchestrA se benefician de una arquitectura de software única, abierta y escalable que puede conectarse a prácticamente cualquier sistema de automatización, unidad terminal remota (RTU), dispositivo electrónico inteligente (IED), controlador lógico programable (PLC), base de datos, historiador o sistema de negocios en uso hoy en día. La naturaleza abierta de esta plataforma les permite a los usuarios expandir sus sistemas existentes sin necesidad de adquirir nuevo hardware o sistemas de control.

La implementación de aplicaciones geográficamente dispersas, desde unos cuantos cientos hasta un millón de I/O, y desde un solo nodo hasta cientos de estaciones, puede realizarse de manera rápida y segura.”

Beneficios

- Fáciles de usar e implementar.
- Fácil configuración y mantenimiento.
- Alta seguridad y disponibilidad.
- Escalabilidad prácticamente ilimitada.

Capacidades

- Visualización HMI y SCADA distribuido geográficamente.
- Mantenimiento y desarrollo a base de plantillas.
- Despliegue de aplicaciones remotas y gestión de cambios.
- Seguridad de niveles de datos integrada al sistema.
- Definición de alarmas fácil y flexible.
- Análisis y recolección de datos para sistemas nuevos y existentes.
- Generación de reportes fácil de usar.

- Acceso abierto a datos históricos.”⁽⁷⁾

Además haciendo referencia específicamente al software InTouch HMI se dice:

“El software InTouch ofrece funciones de visualización gráfica que llevan sus capacidades de gestión de operaciones, control y optimización a un nivel completamente nuevo. Aquello que ahora se conoce en la industria como HMI (Human Machine Interface) comenzó hace más de veinte años con el software InTouch.

Ningún otro HMI en el mercado puede compararse al software InTouch en términos de innovación, integridad de arquitectura, conectividad e integración de dispositivos, ruta de migración de versiones de software sin interrupciones y facilidad de uso.

Esto se traduce en sistemas basados en estándares que permiten incrementar al máximo la productividad, optimizar la efectividad del usuario, mejorar la calidad y reducir los costos operacionales, de desarrollo y de mantenimiento.”

Beneficios:

- Facilidad de uso que le permite a desarrolladores y operarios ser más productivos de manera simple y rápida.
- Gran integración de dispositivos sin y conectividad a prácticamente todos los dispositivos y sistemas.

⁷© 2002-2011 Invensys Systems, Inc. (s.f.). *Soluciones de Software HMI/SCADA*. Recuperado el 03 de 03 de 2011, de <http://global.wonderware.com/LA/Pages/WonderwareHMISCADA.aspx>

- Sus capacidades de representación gráfica y la interacción con sus operaciones permiten entregar la información correcta a las personas correctas en el momento correcto.
- Migración de versiones de software sin interrupción, lo que significa que la inversión en sus aplicaciones HMI está protegida.

Capacidades:

- Gráficos de resolución independiente y símbolos inteligentes que visualmente dan vida a su instalación directamente en la pantalla de su computadora.
- Sofisticado sistema de scripting para extender y personalizar aplicaciones en función de sus necesidades específicas.
- Alarmas distribuidas en tiempo real con visualización histórica para su análisis.
- Graficación de tendencias históricas integrada y en tiempo real.
- Integración con controles Microsoft ActiveX y controles .NET.
- Librería extensible con más de 500 de objetos y gráficos prediseñados, “inteligentes” y personalizables” (⁸).

Otra de las opciones que se podría utilizar es Visual Basic, que sin ser una herramienta especializada para desarrollar HMI es un programa que permite diseñar, ejecutar y desarrollar otros programa, es un lenguaje de aplicaciones, que permite crearlas.

⁸© 2002-2011 Invensys Systems, Inc. (s.f.). *Wonderware.com*. Recuperado el 03 de 03 de 2011, de <http://global.wonderware.com/LA/Pages/WonderwareInTouchHMI.aspx>

1.10.4.2 Visual Basic

Es un lenguaje de programación y una aplicación que fue desarrollada por Allan Cooper para Microsoft, nació a partir del BASIC (Beginner's All-purpose Symbolic Instruction Code), el mismo que fue creado originalmente en el Dartmouth College. Evolucionó de ser un lenguaje para principiantes hasta llegar a ser un poderoso lenguaje de programación. Visual que permite desarrollar programas robustos, es una alternativa buena para programadores de cualquier nivel que deseen desarrollar aplicaciones mayormente para Windows.

La aplicación Visual BASIC, permite crear ventanas, botones, menús, etc. De manera considerablemente sencilla con solo arrastrar y soltar los elementos en el entorno de trabajo. Luego se pueden definir las apariencias, posiciones y comportamientos tanto de forma visual como utilizando códigos de programación. Cuando se crea un programa en Visual Basic deben considerarse los siguientes pasos:

- Creación del interface de usuario. La parte que hace transparente la comunicación entre el usuario y el sistema domótico implementado, al poder seleccionar las acciones a seguir, y visualizar el estado de los diferentes elementos controlados en ventanas, a través de los diferentes controles, de un modo amigable con el usuario.
- Escritura y generación del código por procedimiento, asociando de objetos a eventos, etc. De acuerdo a lo requerido en el programa.
- Generación de código de bloques, módulos, funciones, procedimientos de eventos producidos durante la ejecución del programa.
- Generación de aplicaciones ejecutables.

CAPÍTULO II

DISEÑO DOMÓTICO

2.1 DESCRIPCIÓN DE LA VIVIENDA

Una de las consideraciones más importantes al momento de realizar una instalación domótica es conocer el lugar físico en el cual esta se realizará, así podremos conocer aspectos importantes e interesantes como el costo, números de nodos a utilizarse, tipo de nodos necesarios, entre otras cosas. Normalmente el estudio de las necesidades técnicas a ser satisfechas se realizará en base al estudio de los planos eléctricos y arquitectónicos.

De una manera general, las integradoras, prefieren realizar la domotización de viviendas en etapa de construcción para evitar arreglos mayores en el inmueble, sin embargo en esta ocasión se realizará el diseño sobre una vivienda ya construida.

El hogar tipo es del tipo rústico, ubicado en la ciudad de Sangolquí, Ciudadela Terracota F casa 31. Se divide en dos pisos; en el inferior se encuentran la sala, el comedor, cocina, baño social, estudio y un patio posterior; en el piso superior se ubican 3 dormitorios, dos baños, y una sala de estar. El detalle de la vivienda se puede encontrar en los planos arquitectónico y eléctrico que se encuentran a continuación.

La siguiente figura muestra la localización del bien inmueble con respecto al resto de la ciudadela Terracota F.



Fig. 18 Posición del inmueble con respecto al resto de la ciudadela Terracota F

2.1.1 PLANOS ARQUITECTÓNICOS

En las figuras siguientes se observan los planos eléctricos, sin embargo se los puede observar con mayor detalle en el ANEXO #1

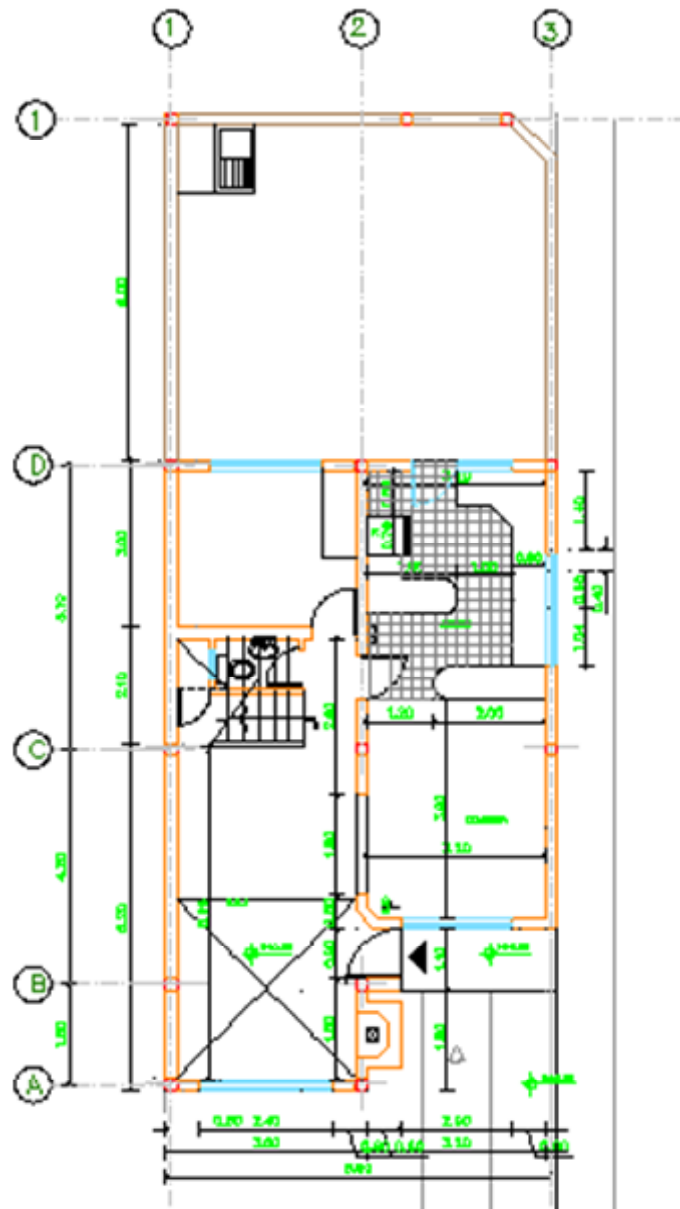


Fig. 19 Plano arquitectónico. Planta baja.

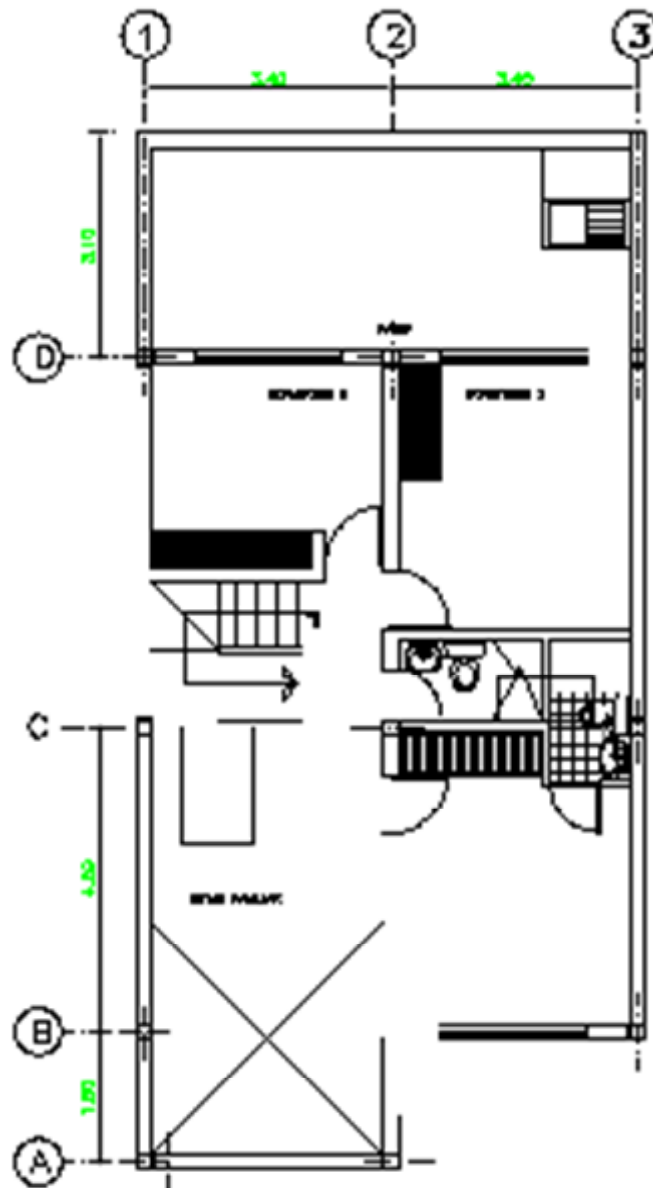


Fig. 20 Plano arquitectónico. Planta alta.

2.1.2 PLANOS ELÉCTRICOS

En las figuras siguientes se observan los planos eléctricos, sin embargo se los puede observar con mayor detalle en el ANEXO #2.

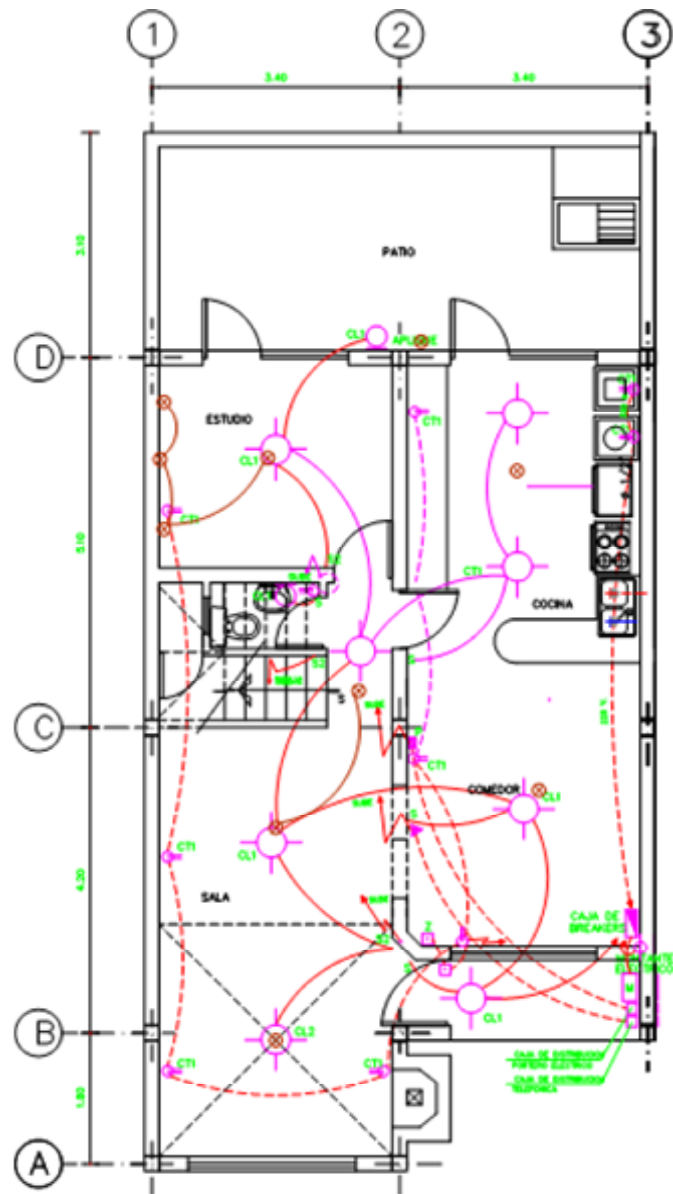


Fig. 21 Plano eléctrico. Planta baja.

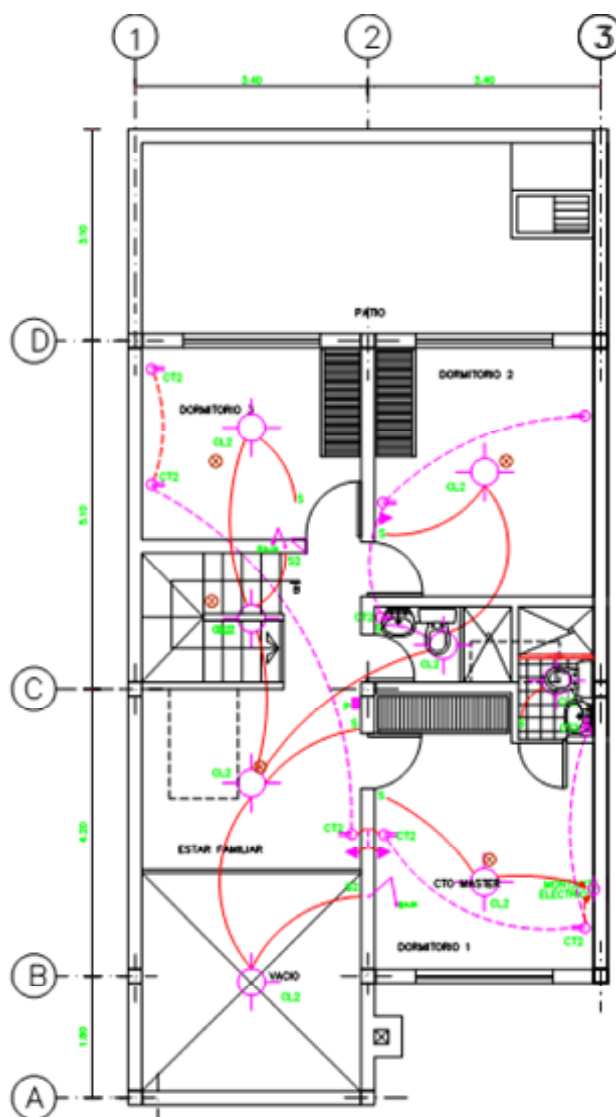


Fig. 22 Plano eléctrico. Planta alta.

2.1.3 VIVIENDA ANTES DE LA AUTOMATIZACIÓN

En el afán de poder comparar la estética y funcionalidad del inmueble antes y después de la domotización se ha incluido un compendio de fotos, las cuales se han tomado en puntos clave y en donde posteriormente se localizarán los nodos domóticos.

En primer lugar podemos observar la sala y el comedor, lugares en los cuales irán circuitos inteligentes de iluminación y varios sensores.



Fig. 23: Sala y comedor

En esta foto se puede observar el muro del estudio, en donde se colocará un nodo dedicado a la iluminación y a la alarma de fuga de gas, la cual está colocada en la cocina y conectada con el nodo a través de una tubería insertada en el muro.



Fig. 24 Estudio

Esta foto corresponde a una de las habitaciones ubicadas en el piso superior de la vivienda. En este lugar irá colocado uno de los sensores de movimiento.



Fig. 25 Cuarto

En esta foto se puede observar el muro del estudio, en donde se colocará un nodo dedicado a la iluminación y a la alarma de fuga de gas, la cual está colocada en la cocina y conectada con el nodo a través de una tubería insertada en el muro.



Fig. 26 Sala de estar

En el cuarto Máster irá ubicado únicamente un sensor de movimiento que controlará un circuito de iluminación.

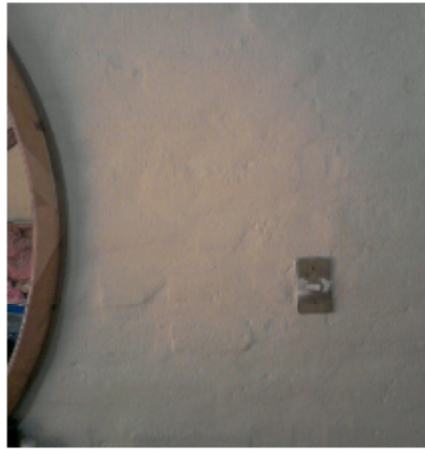


Fig. 27 Cuarto Máster

2.1.4 ANÁLISIS DE NECESIDADES DE LOS DUEÑOS DEL HOGAR TIPO

Después de conversar y recopilar información sobre las diferentes inquietudes por parte de los miembros del núcleo familiar y que aportarían a la selección de los distintos sistemas a instalarse destacan ampliamente la instalación de circuitos inteligentes de iluminación y el área de la seguridad, es decir, todo lo relacionado con las diferentes alarmas técnicas, alarma contra intrusión, simulación de presencia y la recepción de avisos vía telefónica.

Entre los miembros de mayor edad de la familia se encuentra un mayor interés por todos los sistemas relacionados con la seguridad y el ahorro energético, mientras que los de menor edad manifestaron un mayor interés por los sistemas enfocados al confort.

Debido a los costos y a la influencia de los miembros del hogar de mayor edad se realizará la instalación del sistema de iluminación y de seguridad.

2.2 DISEÑO GENERAL DEL SISTEMA DOMÓTICO

Después de analizar los requerimientos del usuario se ha optado por la instalación de los siguientes sistemas:

- a) Sistema de iluminación, encendido y apagado inteligente.
- b) Sistema de seguridad integral.
- c) Control telefónico.
- d) Control mediante HMI.
- e) Back up de energía mediante batería de alimentación.

2.2.1 EQUIPOS LONWORKS A UTILIZARSE

Para cubrir la necesidad de los sistemas a instalarse se requerirá la utilización de los siguientes equipos LonWorks:

Tabla # 4: Nodos a emplearse en el sistema domótico

Cantidad	Nodo ISDE	Función
2	Avanzado INS – 451	Nodo de propósito general, 4 salidas y 6 entradas.
2	Estándar INS – 231	Nodo de propósito general, 2 salidas y 6 entradas.
1	Fuente de alimentación IFA – 200	Nodo que provee de alimentación de 12V a los demás nodos de la red domótica.
1	Integral de supervisión INM – 011	Nodo que contiene las alarmas de incendios, fuga de gas y contra intrusos, aparte de realizar la “Simulación de presencia”.
1	Telefónico sin domoportero INM – 020	Nodo que permite la comunicación vía telefónica entre el usuario y el sistema domótico.
1	Medidor de luz INM –	Nodo que mide diferentes umbrales de

	030	luminosidad con el fin de determinar la necesidad de encender o apagar el sistema de iluminación.
1	USB LonWorks	Permite la comunicación USB – LonWorks entre la PC y el sistema domótico.

Los detalles técnicos de cada nodo se los puede encontrar en el ANEXO #3.

2.2.2 MEMORIA FUNCIONAL

El sistema domótico fue diseñado en función de dos plantas: baja y alta, los circuitos de iluminación y de la disposición del tablero eléctrico y la línea telefónica, como se observa en las siguientes figuras:

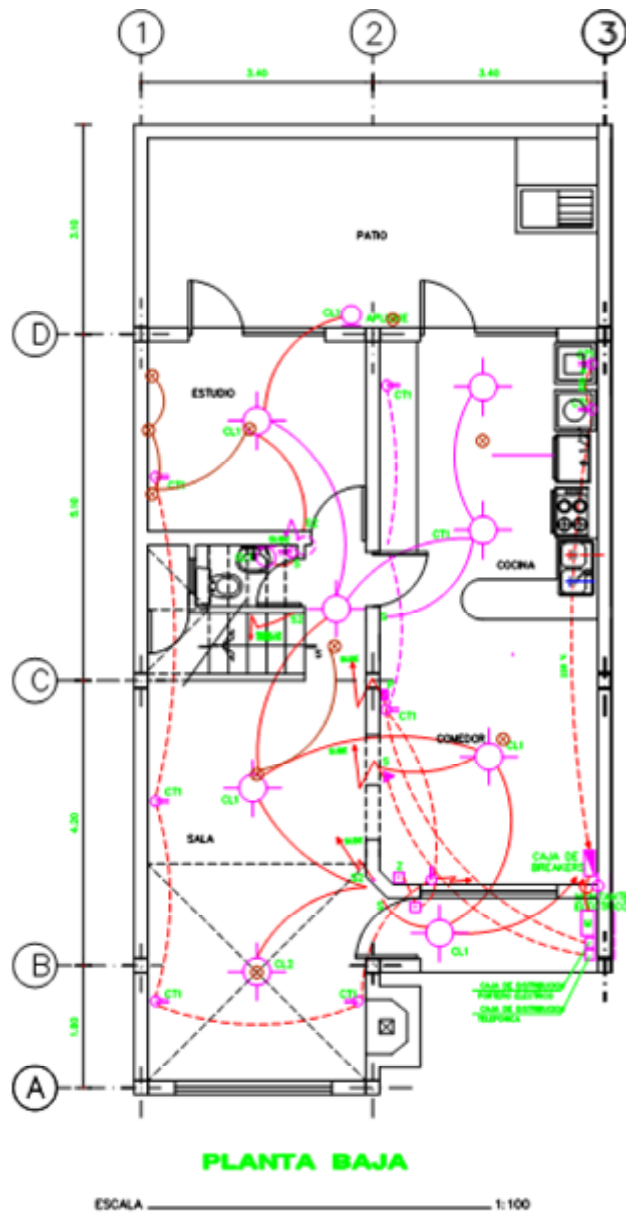


Fig. 28 Planta baja. Plano eléctrico.

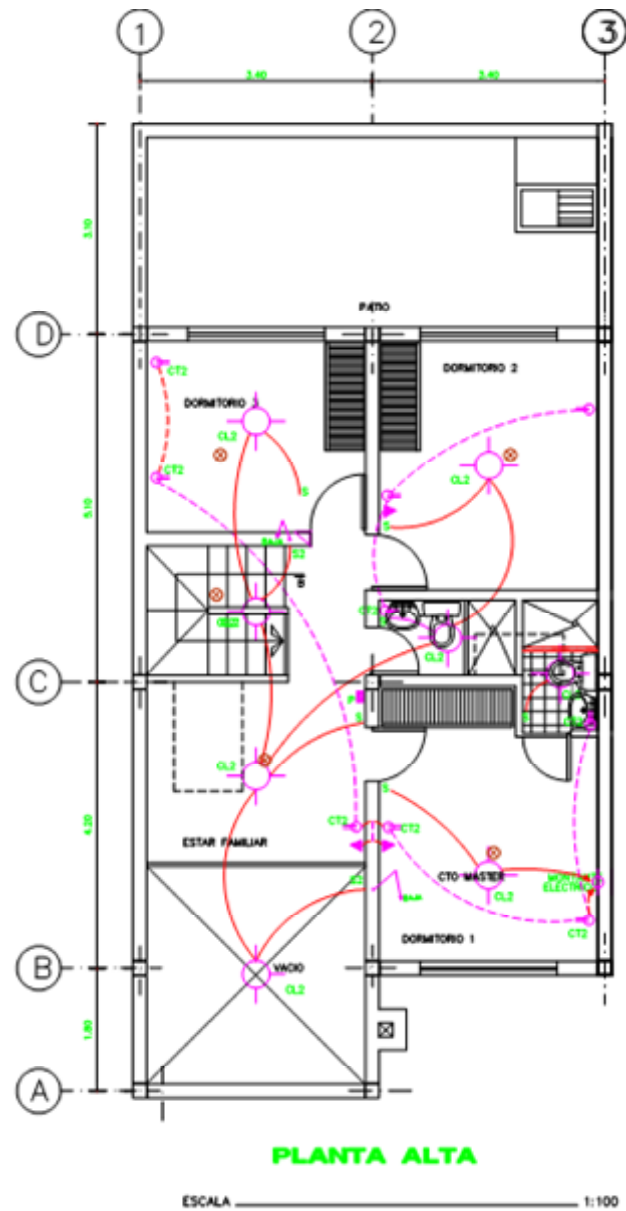


Fig. 29 Planta alta. Plano eléctrico.

Para ver los planos eléctricos con más detalle se puede acudir al ANEXO # 2.

El sistema domótico será diseñado en función de las necesidades de los miembros del hogar tipo y cuenta con la gestión de los siguientes servicios:

Tabla # 5 Sistemas presentes en el presente proyecto

SERVICIOS		DESCRIPCIÓN	
Detección fallo suministro eléctrico	del de	<p>Cuando se produce fallo de suministro eléctrico entra en funcionamiento la batería auxiliar.</p> <p>Aviso telefónico del fallo eléctrico al número de teléfono programado.</p>	
Gestión seguridad	de	<p>Gestión de alarmas técnicas.</p> <p>Detección de intrusión.</p> <p>Las alarmas técnicas son notificadas para que sean atendidas.</p> <p>Simulación de presencia.</p>	<p>Fuga de gas.</p> <p>Presencia de Humo.</p>
Gestión energética		Control de iluminación.	Encendido / apagado de circuitos de luz por detección de presencia o manual.
Gestión confort	del	Encendido automático de los circuitos de luz por detección de presencia y según iluminación exterior.	
Gestión comunicaciones	de	Control telefónico del sistema mediante módulo telefónico, desde un teléfono interior o exterior.	
Interfaces usuarios	de	<p>A través de los botones e iconos del módulo de supervisión se observa el estado de la vivienda y se actúa sobre el sistema domótica.</p> <p>A través del teléfono, pulsando distintos códigos numéricos se realizan distintas acciones y se obtiene información del sistema.</p>	

De una manera más específica el sistema domótico cuenta con 4 sistemas principales, 1) Iluminación inteligente, 2) Alarmas, 3) Control telefónico, y

4) HMI (Interfaz Humano Máquina). Los nodos *supervisión*, *telefónico* y *fuelle* se encuentra en el cuadro de control ubicado en la estancia *Comedor*.

El sistema de iluminación inteligente cuenta con dos formas de activación, a través de sensores de presencia y de pulsadores. Mediante sensor de presencia y pulsador, los circuitos de iluminación de las siguientes estancias:

- Sala.
- Comedor.
- Estudio.
- Todas las habitaciones.
- Sala de estar.

Por otra parte se cuenta con la automatización de los circuitos de iluminación activados por medio de pulsador de las siguientes estancias:

- Cocina.
- Escaleras.
- Patio trasero.

Sin embargo, las estancias *Escaleras* y uno de los circuitos de *Sala* se activan también con el sensor de presencia de las estancias *Sala de estar* y *Sala*, respectivamente.

El sistema de alarma contra intrusión ocupará los sensores de presencia ubicados en las estancias *Sala* y *Comedor*. El nodo de *supervisión* puede ser controlado desde su panel ubicado en la parte frontal del mismo, a través

de la HMI en la PC y finalmente mediante el nodo *telefónico*. El panel frontal podemos apreciarlo en la siguiente figura:



Fig. 30 Panel frontal nodo de supervisión.

Este panel está dividido en tres grupos detallados como se aprecia en la siguiente figura:

ZONA ALARMAS TÉCNICAS		ZONA CONTROL ILUMINACIÓN EXTERIOR	ZONA SIMULACIÓN Y VIGILANCIA		
1	Estado Vigilancia de Fuga de Agua	5	Indica estado luz exterior en Modo Automático.	7	Estado Simulación de Presencia.
2	Estado Vigilancia de Fuga de Gas	6	Indica estado luz exterior en Modo Manual.	8	Prueba de sonido de Buzzer Interno.
3	Estado Vigilancia de Presencia de Humo			9	Estado de vigilancia de Intrusión.
4	Estado Vigilancia de Presencia de Fuego			10	Activación / Desactivación de Vigilancia de Intrusión.

Fig. 31 Detalle de los grupos del panel frontal del nodo de supervisión.

El sistema actual utilizará las alarmas técnicas de *Fuga de gas* y de *Presencia de humo*, también contará con el servicio de *Simulación de Presencia*. Las alarmas son activadas mediante la pulsación de los botones en el nodo *supervisión*; la activación se representa mediante un led indicador.

El nodo *telefónico* sirve para controlar a través del teléfono de la vivienda o llamando desde un teléfono exterior (el teléfono debe ser *multifrecuencia*), elementos como la iluminación y las alarmas técnicas. Por otra parte llamará al teléfono que se haya programado, para comunicar mediante VOZ, cualquier incidencia que se registre en la vivienda. A continuación se observará el detalle de la parte frontal del nodo *Telefónico*:



Fig. 32 Panel frontal nodo telefónico.



STOP. El led se enciende para indicar que se ha realizado tres intentos fallidos de código de acceso. Al producirse esta alarma el nodo se bloquea durante 10 minutos, durante el cual no será posible el intento de acceso, tanto en modo local o remoto. Después de este tiempo se puede volver a intentar ingresar al nodo, mas no se desactivará el led hasta que se teclee el código 98#.



RING. Icono informativo, no contiene pulsador. Se activa cada vez que el nodo detecta un ring del operador de telefonía.



SALIDA DE TELÉFONO. DESVÍO DE LLAMADAS.

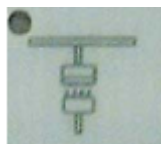
Es el único de este bloque que tiene tecla asociada. Si está activado, significa que se realizarán llamadas de alarma a los números programados, caso contrario, almacenará las alarmas provenientes de la red en su memoria interna (las últimas 5 producidas). Al mantener pulsada esta tecla durante más de 6 segundos, se configura el modo de funcionamiento de “sin línea exterior”. El icono de red (bloque batería y comunicaciones) se apagará. Si se vuelve a pulsar esta tecla durante más de 6 segundos se encenderá el icono de red, configurándose en modo “con línea exterior”. Esta función se utiliza para indicar si tenemos línea telefónica exterior, o no.



DESCOLGAR. Se activa cuando el nodo *telefónico* ha descolgado para realizar una llamada o porque se ha realizado una llamada desde el exterior y detecta el número de rings programado (cada vez que se produzca un ring se enciende el led).



BATERÍA. Si el led se mantiene encendido fijo indica que la batería funciona sin ningún problema. Al mantenerse parpadeando una vez por segundo indica que la batería se encuentra en proceso de carga. Parpadeando tres veces por segundo indica que no existe comunicación con el nodo fuente de alimentación. Apagado, indica que la batería se encuentra en mal estado.



RED. Este icono establece dos estados bien diferenciados. Encendido, indica que el nodo está configurado en modo “CON LÍNEA TELEFÓNICA EXTERIOR”, mientras que apagado, indica que el nodo está configurado en modo “SIN LÍNEA TELEFÓNICA EXTERIOR”.

Este nodo tiene cuatro modos de funciones: *GENÉRICAS*, *CONTROL*, *PROGRAMACIÓN* y *CONSULTA*. En la siguiente tabla se observan las funciones utilizadas y habilitadas para el presente sistema:

Tabla # 6 Funciones del nodo TELEFÓNICO

GENÉRICAS			
1#	Ingreso al nodo telefónico.	99#	Salida del sistema.
89#	Incidencias pendientes.	4321#	Código de acceso (de fábrica).
98#	Borrado del led STOP, del nodo.		
CONTROL			
12#	Vigilancia activada.	13#	Vigilancia desactivada.
18#	Simulación de presencia activada.	19#	Simulación de presencia desactivada.
20#	Encendido luz de la estancia sala.	21#	Apagado luz de la estancia sala.
26#	Encendido general de las luces.	27#	Apagado general de las luces.
30#	Aviso telefónico activo.	31#	Aviso telefónico desactivo.
34#	Vigilancia de gas activa.	35#	Vigilancia de gas desactivada.
36#	Alarma de humo activa.	37#	Alarma de humo desactivada.

60#	Activación de vigilancia Zona 1.	61#	Desactivación vigilancia Zona 1.
PROGRAMACIÓN			
70#	Cambia número de <i>rings</i> al cabo al cabo de los cuales el nodo, cuando es llamado, descuelga (como un contestador automático). Se pueden programar desde 1 hasta 25 rings.	71#	Cambia el número del primer teléfono al cual llamará el nodo en caso de alarma. Podemos introducir cualquier número no superior a 20 dígitos y no inferior a 1.
72#	Cambia el número del segundo teléfono al cual llamará el nodo en caso de alarma. Podemos introducir cualquier número no superior a 20 dígitos y no inferior a 1.	75#	Modifica el código de acceso al sistema, el cual está compuesto por cuatro dígitos.
76#	Modificar el código de marcación que va a usar el nodo cuando llame al exterior. A continuación, se pulsará el valor 1#, si se desea que la marcación se realice por tonos mientras que se pulsará el valor 2#, si queremos que la marcación al exterior se realice por cortes.		
CONSULTA			
80#	Consulta el número de <i>rings</i> que actualmente	81#	Consultar el número del primer teléfono al cual

	están programados.		llamará en caso de alarma.
82#	Consulta el número del segundo teléfono al cual llamará en caso de alarma.	85#	Consulta el código de acceso actual.
86#	Consulta el tipo de marcación que se va a utilizar para llamada al exterior. Si la marcación es de tonos, el mensaje que se oirá será: “CODIGO UNO” Si la marcación es de cortes, el mensaje que se oirá será: “CODIGO DOS”	89#	Consulta las incidencias pendientes, osea, las que no han podido ser comunicadas por teléfono. Las alarmas, cuando se producen, se almacenan en la memoria interna del nodo en forma de cola. Cuando se consultan, van repitiéndose en el mismo orden que se produjeron.
96#	Conoce el estado del circuito de iluminación. Si el circuito de iluminación esta encendido, el mensaje que se oirá será: “LUZ ACTIVADA” Si el circuito de iluminación esta desactivado, el mensaje que se oirá será: “LUZ DESACTIVADA”	101#	Conoce el estado de la alarma de gas.
102#	Conoce el estado de la alarma de humo.	106#	Conoce el estado de la simulación controlada por el supervisor.
107#	Permite conocer el estado de la vigilancia de la zona 1.		

El uso del nodo telefónico se lo realiza prácticamente en su totalidad con el uso del teléfono y mediante la utilización de dos modalidades distintas, 1) modo Local, y 2) modo Remoto, es decir, usando un teléfono ubicado físicamente dentro de la casa o uno ubicado fuera de la casa. El teléfono por obligatoriedad debe ser multifrecuencia (tonos).

Para el manejo del nodo en modo local se deben seguir los siguientes pasos:

- 1) Descolgar el teléfono.
- 2) Se pulsa el código de acceso al nodo: 1#.
- 3) Se escuchará el siguiente mensaje: *“INTRODUZCA FUNCIÓN”*.
- 4) Se ingresará la función que se desea el nodo realice (observar la Tabla de funciones del nodo TELEFÓNICO). En el caso de que la función sea de seguridad se pedirá el código de acceso, el cual es de fábrica 4321#, sin embargo es fácilmente configurable por el usuario.
- 5) Para finalizar la comunicación se marcará la función 99#.

En el modo remoto el manejo del sistema es muy similar al modo local con las siguientes diferencias:

- 1) Marcar el número telefónico del hogar. Cuando hayan pasado el número de rings programados el nodo descuelga y solicita el código de acceso. El cual es de fábrica 4321#, sin embargo es fácilmente configurable por el usuario.
- 2) Si el código es el correcto se escuchará el siguiente mensaje: *“INTRODUZCA FUNCIÓN”*.

- 3) El manejo del nodo telefónico se convierte en exactamente el mismo que en el modo local a raíz de este punto.
- 4) Para finalizar la comunicación se marcará la función 99#.

Nota: Si la función ingresada no se encuentra activada en la red domótica se escuchará el mensaje: “*CÓDIGO NO VALIDO*”.

2.2.3 PROGRAMACIÓN DE NODOS DOMÓTICOS

La programación de los nodos domóticos LonWorks se realiza en el software llamado LonMaker, de la empresa Echelon, del cual existen dos versiones: Estándar y Profesional. El programa funciona como una extensión de Microsoft Visio, utilizando así su lenguaje gráfico.

Al abrir el LonMaker se encuentra la pantalla que se observa en la siguiente figura, se distinguen algunas pestañas en las cuales se encuentran las diferentes opciones con las que cuenta el programa; también observamos la opción *New Network* en donde se procederá a escribir el nombre de la red que se desea, posteriormente se dará un clic en el botón *Create Network*.



Fig. 33 Ventana principal del LonMaker.

De la misma manera se observa en la parte inferior una barra verde que representa la cantidad de Créditos LonMaker, lo cual muestra el número de dispositivos (nodos domóticos) que pueden ser *Comisionados* (activados). Estos Créditos necesariamente deben ser comprados a la Echelon Corporation mediante un formulario que incluye el LonMaker.



Fig. 34 Créditos LonMaker.

Posteriormente se abrirá una nueva pantalla en la cual se ingresan con más detalle el nombre de la red, y su localización en el disco duro.

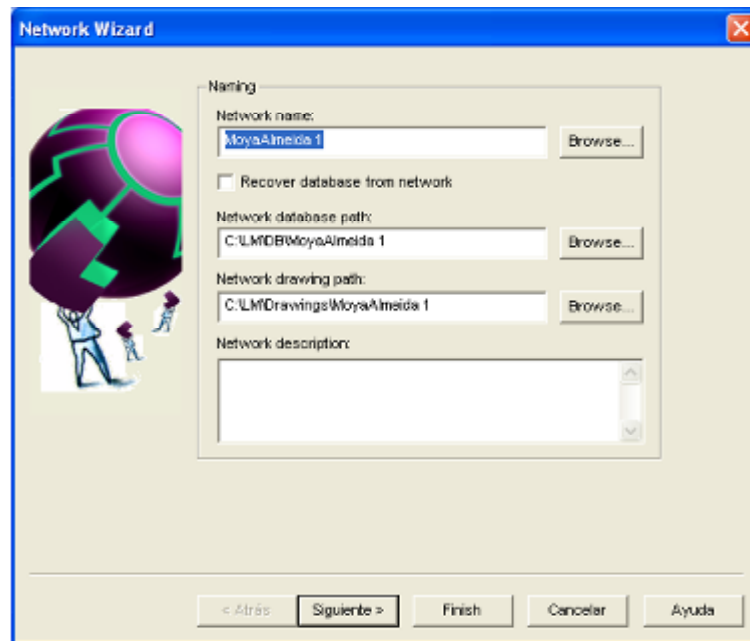


Fig. 35 Network Wizard. Nombre de la red y su ubicación en disco duro.

En la siguiente ventana se escogerá la opción *Network attached*, la cual debe ser activada si la PC se encuentra en ese momento conectada a la red domótica. Esto es completamente necesario para realizar el *Comisionado* de los nodos. Caso contrario se puede realizar la programación de la red domótica para posteriormente conectarla y que así se activen los nodos y se propaguen los valores de configuración y la información sobre la conexión de las variables de red. El menú *Network Interface Name* muestra el nombre del dispositivo físico mediante el cual se realiza el enlace, es decir el *LNS Network Interface*, el que para este caso es el *LONI*.

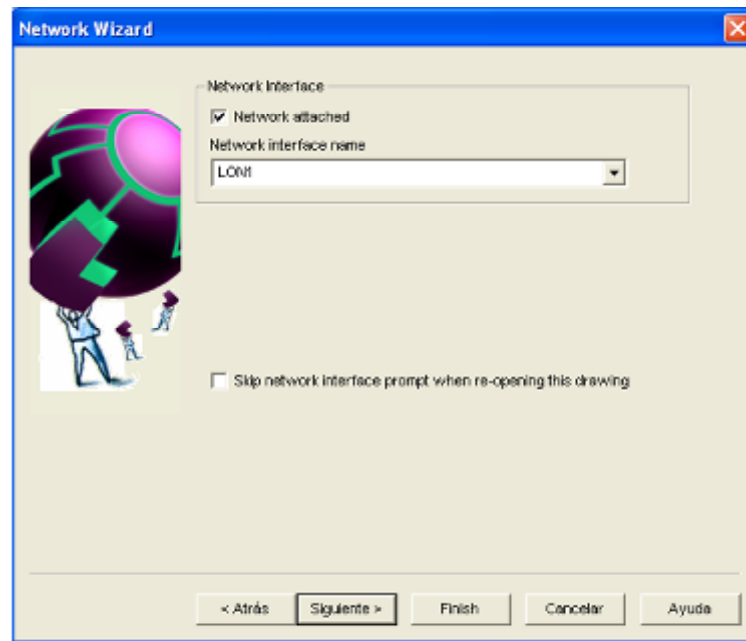


Fig. 36 Network Wizard. Network interface.

En la pantalla *Management Mode* se encuentran las opciones *OnNet* y *OffNet*, las cuales indican si la propagación de la información sobre conexiones y datos de configuración que se programen debe ser en tiempo real (*OnNet*), o bien deben mantenerse en stand by para que se realice manualmente (*OffNet*).

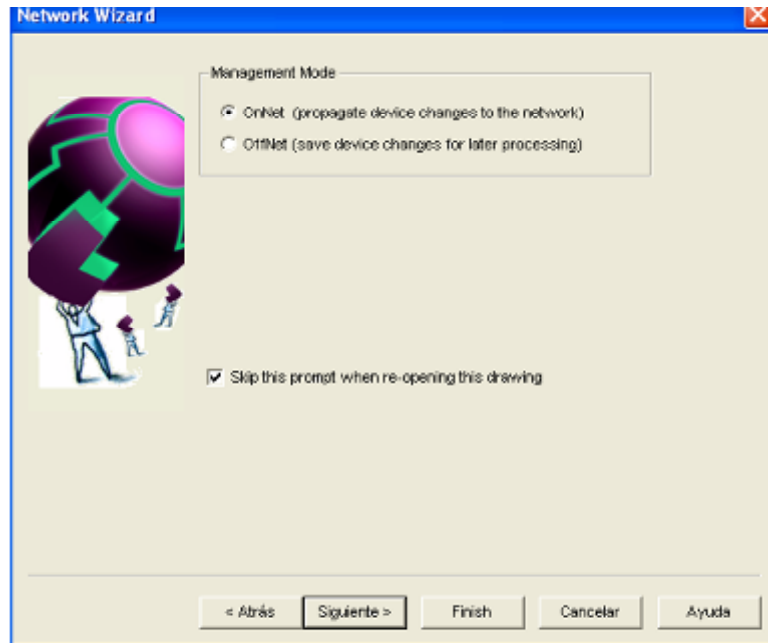


Fig. 37 Network Wizard. Management Mode.

Posteriormente se preguntará qué servicios o *Plug-ins* se desean agregar a la red. Se ha escogido todos los servicios, sin embargo el más importante es el *Echelon LonMaker Browser*, ya que este permitirá ingresar a las bases de datos de los diferentes nodos con el fin de observar su comportamiento y de configurar los valores necesarios.

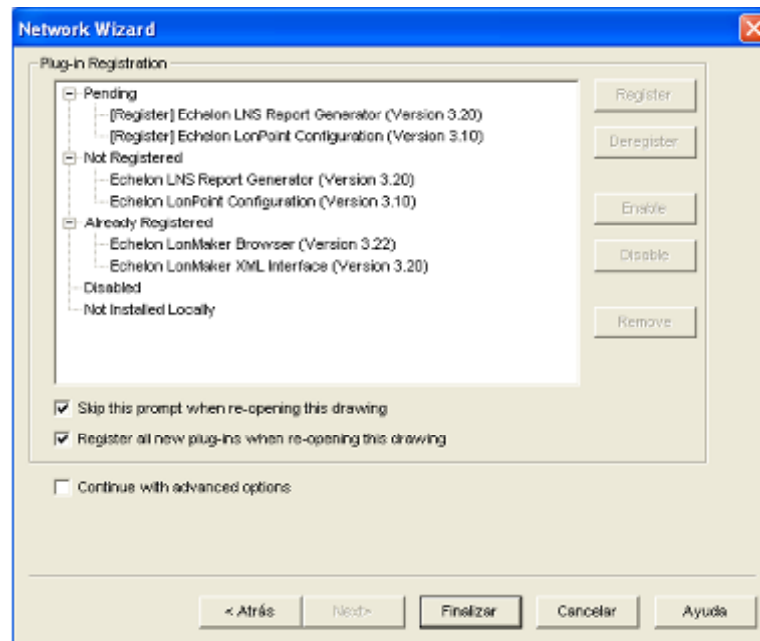


Fig. 38 Network Wizard. Plug-in Registration.

Finalmente se abrirá el *Microsoft Visio* y se desplegará en la parte izquierda las diferentes herramientas del *LonMaker*.

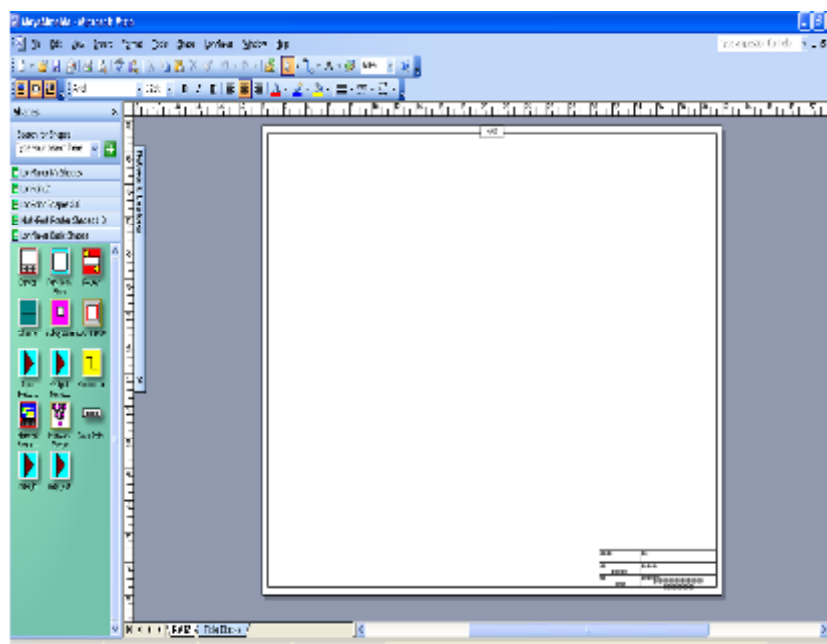


Fig. 39 Microsoft Visio. Pantalla principal.

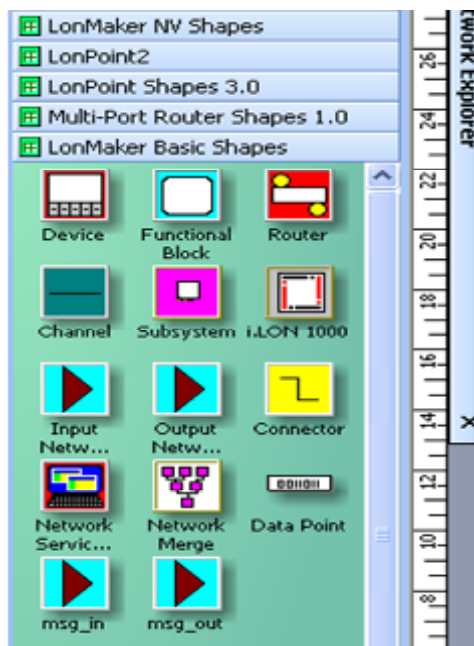


Fig. 40 Microsoft Visio. Principales herramientas del LonMaker.

Para poder agregar un nodo domótico se selecciona el botón *Device* y se lo arrastra hasta el lienzo blanco de la pantalla del *Microsoft Visio*.



Fig. 41 Botón Device.

A continuación se observará la pantalla que permite agregar un nuevo nodo domótico. Si se encuentra conectada la red domótica físicamente se puede seleccionar la opción *Comission Device*, con el fin de que este se active automáticamente. Es importante seleccionar la opción *Create new device*

template, ya que esta le indica al programa que estamos ingresando un nodo con un firmware nuevo, es decir, si se desea agregar más de un nodo del mismo tipo no se debe activar dicha opción. También se muestra las opciones para indicar el tipo de canal que se está utilizando, su tipo y su nombre; para este caso se utilizará el canal *Channel 1* del tipo *TP/FT-10*.

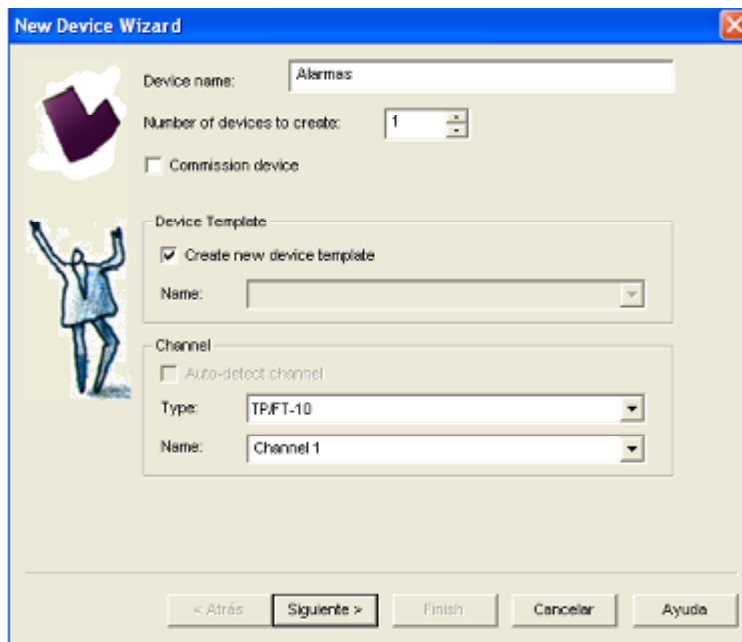


Fig. 42 Device Wizard. Pantalla principal.

Al activar la opción *Create new device template* tenemos la opción de elegir qué tipo de firmware utilizaremos, así que seleccionaremos la opción *Load XIF*, y mediante el botón *Browse* buscaremos el XIF más apropiado para nuestras necesidades entregado por el proveedor de nodos domóticos. Los XIFs utilizados se pueden observar en la siguiente tabla:

Tabla # 7 Nombres de los archivos XIF

NODO	XIF
Fuente. IFA 200.	A9A0700000102

Telefónico. INM-020	A831200000313
Alarmas. INM-011	A9B1202000503
Medidor de Luz INM-030	A850700000402
Iluminación 01. INS-231	A020700000713
Iluminación 02. INS-231	A020700000610
Iluminación 03. INS-451	A030700000203
Iluminación 04. INS-451	A030700000203

Las características particulares de cada archivo .XIF se las encuentra en el ANEXO # 5.

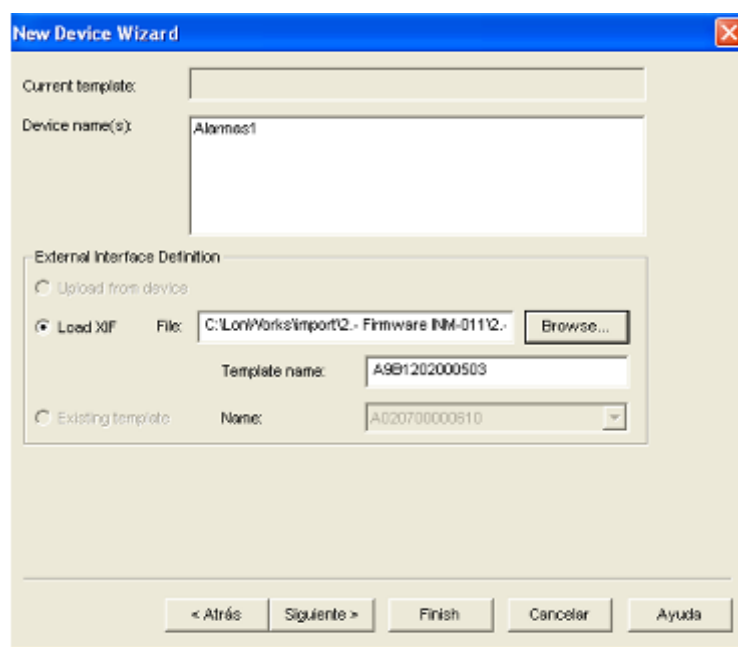


Fig. 43 Device Wizard. Selección del firmware.

La siguiente pantalla simplemente mostrará cierta información sobre el nodo además de permitirnos seleccionar el Intervalo de Pings que se desea dar al nodo con el fin de verificar que este se encuentre activado y/o presente en la red.

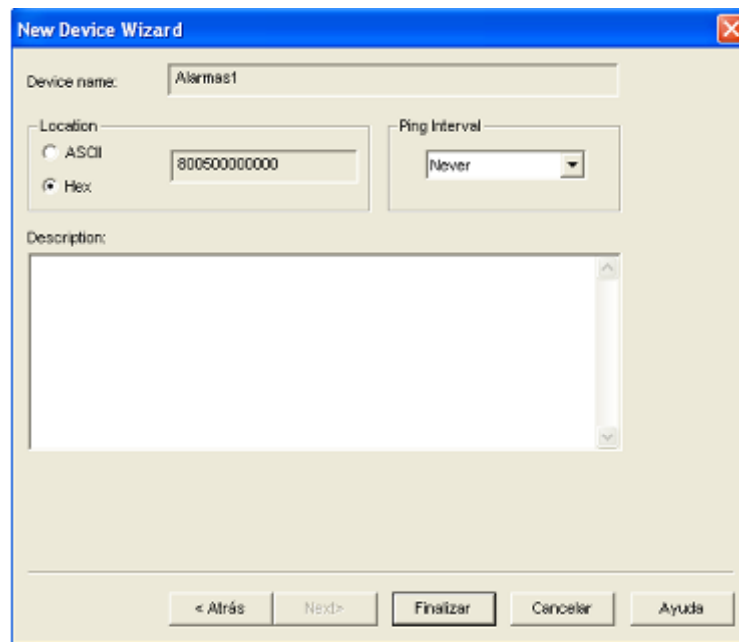


Fig. 44 Device Wizard. Información y Ping Interval.

Finalmente se observará una pantalla en la cual se informará acerca de la creación del nodo domótico.

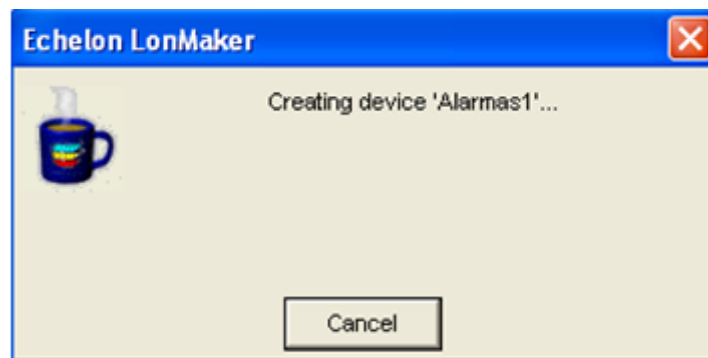


Fig. 45 Device Wizard. Pantalla informativa final.

Para el presente proyecto se repetirá el proceso anterior durante 7 veces más con el fin de agregar todos los nodos domóticos. El bloque *LNS Network*

Interface se refiere al adaptador que permite la comunicación PC – Red LON. Se observa en la siguiente figura a los diferentes dispositivos agregados, los de color verde representan que se encuentran comisionados y conectados, mientras que los de color amarillo pálido representan una de dos opciones 1) que no han sido comisionados o 2) que si lo están no se encuentran conectados; de cualquier manera simboliza que no se puede establecer comunicación con ellos.



Fig. 46 Nodos domóticos agregados.

Los nodos pueden ser comisionados (activados) mediante dos métodos: 1) Utilizando el pin de servicio, o 2) mediante la escritura del Neuron ID. En el presente caso se ingresaron cada uno de los 7 Neuron ID, los cuales se muestran en la siguiente tabla:

Tabla # 8 Neuron IDs

NODO	NEURON ID
Fuente. IFA 200.	027507130200
Telefónico. INM-020	026CAFC10200
Alarmas. INM-011	02EC5C5E0100
Medidor de Luz INM-030	02AC90030200
Iluminación 01. INS-231	02B85A860200
Iluminación 02. INS-231	0292FC2D0200
Iluminación 03. INS-451	029204330200
Iluminación 04. INS-451	02F435860200

Cada nodo domótico deberá por obligatoriedad estar relacionado a uno y sólo uno *Bloque Funcional*, quien será el que contenga las diferentes variables de red, configuraciones y conexiones. Estos representan en sí mismos a los nodos y sus funciones. El botón para agregar a estos bloques es el siguiente:



Fig. 47 Botón Funtional Block.

Después de arrastrar el ícono hasta el lienzo blanco se desplegará la pantalla de configuración del *Funtional Block*, en el cual podremos asignarle un nombre, el nodo al cual pertenece, entre otros datos.

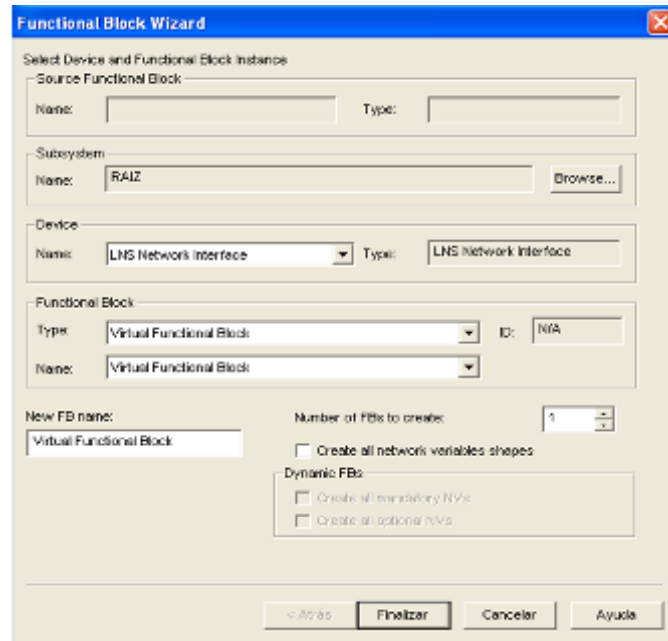


Fig. 48 Functional Block Wizard. Pantalla principal.

Posteriormente podremos agregar variables de red de entrada o de salida y de conectores mediante el uso de los siguientes botones:



Fig. 49 Botones para agregar variables de red y conexiones.

Después de agregar la variable de red *nvoFaltan220v*, de la cual se hablará más tarde, al nodo Fuente se podrá visualizar el *Functional Block* de la siguiente forma:

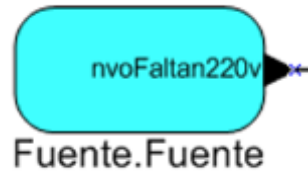


Fig. 50 Funtional Block del nodo Fuente.

Finalmente después de agregar todos los *Funtional Block* necesarios y realizar las conexiones lógicas entre sus diferentes variables de red la red LON queda activada y levantada como se visualiza en la siguiente imagen:

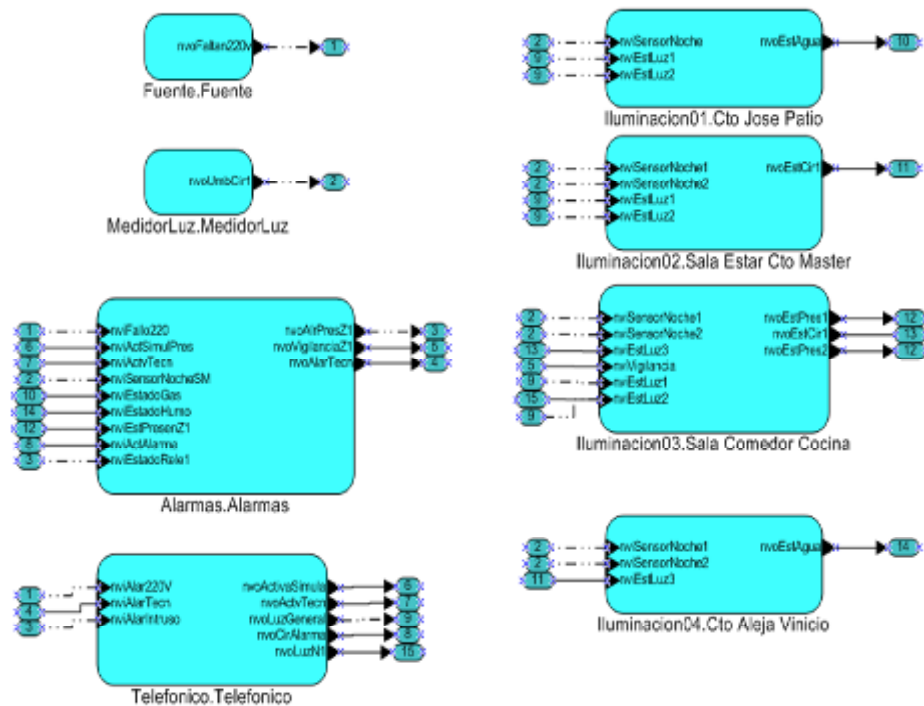


Fig. 51 Funtional Blocks de todo el sistema domótico.

Para observar en mayor detalle el gráfico anterior remitirse al ANEXO # 4.

A continuación se encontrará una tabla en la cual se puede identificar los diferentes nodos y conexiones realizados. El detalle de las variables de red de entrada y salida y los valores de configuración se los encuentra en el ANEXO # 5.

Tabla # 9 Conexiones de variables de red de entrada y salida en cada nodo.

CONEXIONES DE VARIABLES DE RED DE ENTRADA Y SALIDA EN CADA NODO				
Número de conexión	Variable de red de salida	Variable de red de entrada	Nodo entrante	Función
NODO FUENTE IFA – 200				
1	nvoFaltan220v	nviFallo220	Supervisión	Informa sobre un corte en el suministro eléctrico.
		nviAlar220V	Telefónico	
NODO MEDIDOR DE LUZ EXTERIOR. INM – 030				
2	nvoUmbCir1	nviSensorNocheSM	Supervisión	Informa sobre una baja del nivel de luz exterior con respecto a un umbral fijado en la variable de configuración cfgUmbralN1.
		nviSensorNoche	Iluminación 01	
		nviSensorNoche1	Iluminación 02	
		nviSensorNoche2	Iluminación 03	
		nviSensorNoche1	Iluminación 03	
		nviSensorNoche2	Iluminación 03	
		nviSensorNoche1	Iluminación 04	

		nviSensorNoche2		
NODO DE SUPERVISIÓN INM - 011				
3	nvoAlrPresZ1	nviAlarIntruso	Telefónico	Informa sobre una detección de presencia en la Zona 1.
		nviEstadoRelé1		
5	nvoVigilanciaZ1	nviVigilancia	Iluminación03	Informa al nodo destino sobre la activación del sistema de vigilancia en la Zona 1.
4	nvoAlarTecn	nviAlarTecn	Telefónico	Informa sobre la activación de las alarmas técnicas.
NODO TELEFÓNICO INM – 020				
6	nvoActivaSimula	nviActSimulPres	Supervisión	Informa sobre la activación vía telefónica de la simulación de presencia.
7	nvoActvTecn	nviActTecn	Supervisión	Informa sobre la activación vía telefónica de la supervisión de las alarmas técnicas.
9	nvoLuzGenera1	nviEstLuz1	Iluminación02	Informa sobre la activación vía telefónica de los
		nviEstLuz2		
		nviEstLuz1	Iluminación	

		nviEstLuz2 nviEstLuz2	03	circuitos de iluminación asociados a él.
15	nvoLuzN1	nviEstLuz2	Iluminación 03	Informa sobre la activación vía telefónica de los circuitos de iluminación asociados a él.
8	nvoCirAlarma	nviActAlarma	Supervisión	Informa sobre la activación vía telefónica de la vigilancia contra intrusión.
NODO ILUMINACIÓN 01 INS – 231				
10	nvoEstAgua	nviEstadoGas	Supervisión	Informa sobre la detección de una fuga de gas.
NODO ILUMINACIÓN 02 INS – 231				
11	nvoEstCir1	nviEstLuz3	Iluminación 04	Genera un paralelo virtual entre los circuitos de iluminación del nodo de origen y de destino.
NODO ILUMINACIÓN 03 INS – 451				
12	nvoEstPres1	nviEstPresenZ 1	Supervisión	Informa sobre una detección de presencia.
	nvoEstPres2			
13	nvoEstCir1	nviEstLuz3	Iluminación 03	Genera un paralelo virtual entre los

				circuitos de iluminación del nodo de origen y de destino.
NODO ILUMINACIÓN 04 INS – 451				
14	nvoEstAgua	nviEstadoHum o	Supervisión	Informa sobre la detección de humo.

También es importante tomar en cuenta de que cada nodo tiene ciertas variables de configuración que le indican a este la manera en la cual debe funcionar.

Tabla # 10 Variables de configuración de cada nodo.

VARIABLES DE CONFIGURACIÓN DE CADA NODO		
Variable de configuración	Valor de configuración	Función
NODO FUENTE IFA – 200		
cfgBackUp Valores: (100, 1 0), (0, 0 0)	(0,0 0)	Configura si la fuente es la principal o de Backup (fuente de apoyo).
cfgTimeFalta220v Valores: 0 - 65535 (seg)	30,0	Tiempo para aviso de falta suministro eléctrico.
cfgTimeAlrMan Valores: 0 - 65535 (seg)	1,5	Tiempo para aviso de alarma de mantenimiento.
cfgTimeEstable Valores: 0 - 65535 (seg)	1,0	
cfgIdEquipo	01	Número de equipo (si hay varios)

Valores: 0 - 255		iguales).
cfgConfiguracion	93BD4D7E293 B4DCECDCD2 9806BDDOE40 710	Para detalle Ver ANEXO # 5.
NODO MEDIDOR DE LUZ EXTERIOR. INM – 030		
cfgUmbralN1 - N6 Valores: 0 - 65535 (lux)	750 1000 2000 3000 4000 5000	Indica el umbral por encima del cual debe considerar que es de día o de noche. El valor real máximo permitido es de 16000 lux.
cfgOffsetLuz	200	Histéresis del algoritmo de detección.
cfgLuxCambio	100	Indica la mínima intensidad que es necesario superar desde la última vez que se propagó la variable nvoLuxCambio para una nueva propagación de la intensidad luminosa.
cfgTimeoutCir1 - 6 Valores: 0 ... 6553.4 (seg)	(0,0) (0,0) (0,0) (0,0) (0,0) (0,0)	Utiliza específicamente el tiempo (seg.) que se debe mantener activada la variable nvoUmbCir1 , desde que el nivel de luz llegó a su umbral, hasta apagarse.
cfgOffDecorC1 Valores: 0 ... 6553.4 (seg)	(1080,0)	Tiempo que permanecerá activada la variable nvoUmbDecor desde su puesta a (100,0 1).
cfgOnDecorC1 Valores: 0 ... 6553.4 (seg)	(180,0)	Tiempo que transcurre desde que la variable cfgUmbDecor detecta que el nivel de luz medido coincide con el de su umbral hasta que

		realmente se pone a (100,0 1). Es un retardo de activación del circuito de decoración.
cfgUmbRefDeco	01	Indica el nivel umbral de los 6 disponibles que se toma como referencia para la variable de decoración.
cfgValorUmbX1 Valores: 0 ... 3 X = 1 ... 6	00	Indica el valor que debe tomar la variable nvoValorX y bajo qué condición debe producirse este cambio de valor.
cfgFactorUmbX2 Valores: X = 1 ... 6	01	Factor por el que es necesario multiplicar para ampliar el tiempo de estado en modo noche para cada uno de los umbrales de los que dispone el firmware.
cfgFactorDecOn	01	Factor por el que es necesario multiplicar para ampliar el tiempo para que el umbral de decoración indique que es de noche referido a un umbral cualquiera.
cfgFactorDecOff	01	Factor por el que es necesario multiplicar para ampliar el tiempo de duración de indicación modo noche para el umbral de decoración.
NODO SUPERVISIÓN. INM – 011		
cfgTimeEntrada1-3 Valores: 0 ... 6553.4 (seg)	(45,0) (40,0) (40,0)	Tiempo de entrada para poder desconectarla vigilancia de las zonas 1, 2 y 3.
cfgTimeSalida Valores:	(120,0)	Tiempo (sg) de salida para que se active la vigilancia.

0 ... 6553.4 (seg)		
cfgTimePostAlar Valores: 0 ... 6553.4 (seg)	(240,0)	Tiempo que estará activado el buzzer desde que se produce una alarma. Transcurrido este tiempo sólo indicarán alarma las señales luminosas del panel.
cfgTimeRearme Valores: 0 ... 6553.4 (seg)	(30,0)	Tiempo de rearme de la vigilancia de las distintas alarmas técnicas. Empieza a contar desde el momento en que se reciba un (0,0 0) en la variable de estado de GAS, HUMO o FUEGO.
cfgTimeoutRel1- 2 Valores: 0 ... 6553.4 (seg)	(240,0) (2,0)	Tiempo de desactivación del relé 1 ó 2. Este tiempo es de activación si cfgInvNivelRel1 o cfgInvNivelRel2 están a (100,0 1).
cfgPreAlarma Valores: (0,0 0) , (100,0 1)	(100,0 1)	Configura la posibilidad de aviso acústico de la prealarma de vigilancia.
cfgBuzSalida Valores: (0,0 0) , (100,0 1)	(100,0 1)	Configura la posibilidad de aviso acústico durante el tiempo de salida de la vigilancia de presencia.
cfgInvNivelRel1-2 Valores: (0,0 0) , (100,0 1)	(0,0 0) (0,0 0)	Cuando está a (100,0 1) se invierten los niveles de los relés 1 ó 2.
cfgSimulDia	051E01	Esta variable establece los límites de tiempo entre los que va a fluctuar la simulación de presencia de día (en segundos o en minutos).
cfgSimulNoche	051E01	Esta variable establece los límites de tiempo entre los que va a fluctuar la simulación de presencia de noche (en segundos o en

		minutos).
cfgNumDetZ	0202020200000 0	Esta variable indica el número de detecciones de presencia que serán consideradas como una alarma de presencia en la zona 1, 2 y 3. Cada uno de los bytes de la estructura corresponde con una zona de vigilancia. En el manual del nodo se explica de mejor manera la configuración de esta variable.
NODO TELEFÓNICO INM – 020		
cfgMarcarDTMF Valores: (100,0 1), (0,0 0)	(100,0 1)	El valor (100,0 1) indica que la marcación se va a realizar por tonos; el valor (0,0 0) informa que la marcación se va a realizar por pulsos.
cfgErrorCom Valores: (100,0 1), (0,0 0)	(0,0 0)	El valor (0,0 0) informa sobre la existencia de errores de comunicaciones; el valor (100, 0 1) no informa de errores de comunicaciones aunque no se realicen las órdenes dadas.
cfgTimePuerta Valores:	(2,0)	
cfgTimeRing Valores:	(5,0)	
NODO ILUMINACIÓN 01 INS – 231		
cfgActuaLocal1-2 Valores: (100,0 1), (0,0 0)	(100,0 1) (100,0 1)	Se utiliza para especificar el modo de funcionamiento de los circuitos nº1 o nº2. Si está a (100,0 1), la actuación sobre el pulsador de este

		circuito afecta a la conmutación R1. Si se pone a (0,0 0), no hay actuación y la actuación sobre el pulsador no tiene ningún efecto local sobre la salida. Solo vale de cara a la red a través de las variables de salida nvoPulCir1-2 y nvoPulMan1-2.
<p>cfgInvNivelCir1-2</p> <p>Valores: (100,0 1), (0,0 0)</p>	<p>(0,0 0)</p> <p>(0,0 0)</p>	<p>Esta variable indica la inversión del funcionamiento del circuito de conmutación 1 ó 2. Si está a (0,0 0) la conmutación es normal. Si se pone a (100,0 1), se invierte la conmutación; es decir, si recibe una activación se abrirá el relé y si recibe una desactivación se cortocircuitará.</p>
<p>cfgInicioSensor</p> <p>Valores: (100,0 1), (0,0 0)</p>	<p>(0,0 0)</p>	<p>Se utiliza para poder configurar el modo de encendido automático de la luz del circuito nº1 con el detector de presencia.</p>
<p>cfgAlmacenaCIR1</p> <p>-2</p> <p>Valores: (100,0 1), (0,0 0)</p>	<p>(0,0 0)</p> <p>(0,0 0)</p>	<p>Se utiliza para indicar al nodo que se desea que el equipo, cuando se produzca un reset, inicialice el valor de CIR1-2 con el último valor que hubiera tenido la variable nviEstLuz1-2.</p>
<p>cfgAlmacenaVal</p> <p>Valores: (100,0 1), (0,0 0)</p>	<p>(0,0 0)</p>	<p>Se utiliza para indicarle al nodo que en sus salidas se van a colocar una electroválvula de dos posiciones y que en caso de producirse un reset las salidas adquieran el último valor que</p>

		hubiera tenido la variable nviValvula.
<p>cfgValInversor</p> <p>Valores: (100,0 1), (0,0 0)</p>	(0,0 0)	Esta variable se utiliza para especificar qué tipo de electroválvula se va a colocar a la salida del equipo. El valor (0,0 0) indica que se va a colocar una electroválvula de dos posiciones, una para abrir y otra para cerrar, mientras que el valor (100,0 1) especifica que la electroválvula a utilizar va a ser de dos posiciones pero con un inversor de polaridad.
<p>cfgSleep</p> <p>Valores: (100,0 1), (0,0 0)</p>	(0,0 0)	Se utiliza para habilitar o deshabilitar el modo sleep o bajo consumo en que entra el nodo cuando está en reposo. Cuando esta variable tiene el valor (100,0 1) el nodo entra en bajo consumo mientras que cuando tiene el valor (0,0 0) esta opción se encuentra deshabilitada.
<p>cfgTimeoutLuz1-2</p> <p>Valores: 0... 6553.4 seg.</p>	(300,0) (600,0)	Se utiliza para especificar el tiempo (seg) que se debe mantener encendido los circuitos de salida nº1 ó nº2 sin que exista ninguna detección de presencia.
<p>cfgTimeRele1-2</p> <p>Valores: 0... 6553.4 sg.</p>	(0,0) (0,0)	Esta variable indica el tiempo que permanece activada o desactivada la salida nº1 ó nº2 (en función de la variable cfgInvNivelCir1 o cfgInvNivelCir2) cuando la variable nviEstLuz1 o nviEstLuz2

		recibe el valor (100,0 1).
<p>cfgTimeFiltroSLL</p> <p>Valores:</p> <p>0... 6553.4 sg.</p>	(60,0)	Esta variable indica cada cuánto tiempo como mínimo debe actualizarse la variable de salida nvoEntLluvia.
<p>cfgTimeoutSens</p> <p>Valores:</p> <p>0... 6553.4 sg.</p>	(60,0)	Tiempo que debe transcurrir desde la última detección de presencia para que la variable nvoApagSensor envíe el valor (0,0 0) como finalización de detección de presencia en la habitación.
<p>cfgTimerValvula</p> <p>Valores:</p> <p>0... 6553.4 sg.</p>	(40,0)	Tiempo durante el cual se encuentran activadas las salidas del equipo para habilitar o deshabilitar la electroválvula situada a sus salidas. Este tiempo únicamente tiene validez cuando se actualiza la variable de red nviValvula.
<p>cfgTimeFiltro</p> <p>Valores:</p> <p>0.. 65535 (mseg)</p>	000A	Esta variable indica el tiempo en milisegundos que transcurren desde la producción de un evento en la entrada de un nodo hasta que verdaderamente se evalúa la actualización de la entrada.
<p>cfgPulsoEst1-2</p> <p>Valores:</p> <p>0... 3</p>	03 03	Esta variable indica si la actuación sobre CIR1-2 es temporizada, cuando se actualiza la variable nviEstLuz1 o nviEstLuz2. Cuando se recibe el valor (100,0 1): - 0 y 2: actuación temporizada. - 1 y 3: actuación no temporizada. Cuando se recibe el valor (0,0 0): - 1 y 2: actuación temporizada.

		- 0 y 3: actuación no temporizada.
<p>cfgPulsoReset1-2</p> <p>Valores: 0... 3</p>	<p>03</p> <p>03</p>	<p>Esta variable indica si la actuación sobre CIR1 o CIR2 es temporizada cuando se produce un reset. Se repite la última operación especificada por nviEstLuz1 o nviEstLuz2 y la variable cfgAlmacenaCIR1 o cfgAlmacenaCIR2 debe tener el valor (100,0 1). Cuando se recibe el valor (100,0 1) en nviEstLuz1 o nviEstLuz2:</p> <ul style="list-style-type: none"> - 0 y 2: actuación temporizada. - 1 y 3: actuación no temporizada. <p>Cuando se recibe el valor (0,0 0):</p> <ul style="list-style-type: none"> - 1 y 2: actuación temporizada. - 0 y 3: actuación no temporizada.
<p>cfgResetVal</p> <p>Valores: 0... 3</p>	<p>03</p>	<p>Esta variable indica si la actuación sobre CIR1 y CIR2 es temporizada cuando se produce un reset. Se repite la última operación especificada por nviValvula y la variable cfgAlmacenaVal debe tener el valor (100,0 1). Cuando se recibe el valor (100,0 1) en nviValvula :</p> <ul style="list-style-type: none"> - 0 y 2: actuación temporizada CIR1. - 1 y 3: actuación no temporizada. <p>Cuando se recibe el valor (0,0 0):</p> <ul style="list-style-type: none"> - 1 y 2: actuación temporizada CIR2. - 0 y 3: actuación no temporizada.

cfgFactorUmb1-2	01 01	Esta variable se utiliza para aumentar el tiempo de encendido de la iluminación. El tiempo final especificado por la variable cfgTimeoutLuz1 o cfgTimeoutLuz2 será ese valor multiplicado por cfgFactorUmb1 o cfgFactorUmb2. Igual para cfgTimeRele1 o cfgTimeRele1, es decir, el tiempo final será cfgTimeRele1 o cfgTimeRele2 multiplicado por cfgFactorUmb1 o cfgFactorUmb2.
NODO ILUMINACIÓN 02 INS – 231		
cfgTimeoutLuz1-2 Valores: 0... 6553.4 seg.	(300,0) (300,0)	Se utiliza para especificar el tiempo (seg) que se debe mantener encendido los circuitos de salida nº1 ó nº2 sin que exista ninguna detección de presencia.
cfgTimeRele1-2 Valores: 0... 6553.4 sg.	(0,0) (0,0)	Esta variable indica el tiempo que permanece activada o desactivada la salida nº1 ó nº2 (en función de la variable cfgInvNivelCir1 o cfgInvNivelCir2) cuando la variable nviEstLuz1 o nviEstLuz2 recibe el valor (100,0 1).
cfgTimeoutSens1-2 Valores: 0... 6553.4 sg.	(60,0) (60,0)	Tiempo que debe transcurrir desde la última detección de presencia para que la variable nvoApagSensor1-2 envíe el valor (0,0 0) como finalización de detección de presencia en la habitación.

<p>cfgInicioSensor1-2</p> <p>Valores: (100,0 1), (0,0 0)</p>	<p>(0,0 0)</p> <p>(0,0 0)</p>	<p>Se utiliza para poder configurar el modo de encendido automático de la luz de los circuito nº1 o nº2 con el detector de presencia.</p>
<p>cfgSensorNocheC</p> <p>Valores: (100,0 1), (0,0 0)</p>	<p>(100,0 1)</p>	<p>Esta variable se utiliza para habilitar el cambio del valor de la variable nviSensorNoche2 tome el valor nviSensorNoche1.</p>
<p>cfgInvNivelCir1-2</p> <p>Valores: (100,0 1), (0,0 0)</p>	<p>(0,0 0)</p> <p>(0,0 0)</p>	<p>Esta variable indica la inversión del funcionamiento del circuito de conmutación 1 ó 2. Si está a (0,0 0) la conmutación es normal. Si se pone a (100,0 1), se invierte la conmutación; es decir, si recibe una activación se abrirá el relé y si recibe una desactivación se cortocircuitará.</p>
<p>cfgRellano1-2</p> <p>Valores: (100,0 1), (0,0 0)</p>	<p>(100,0 1)</p> <p>(100,0 1)</p>	<p>Esta variable indica si los circuitos de iluminación nº1 y nº2 deben funcionar con doble pulsación y encendido automático (100,0 1) o sin doble pulsación de encendido y modo automático (modo rellano, 0,0 0).</p>
<p>cfgTimeFiltro</p> <p>Valores: 0.. 65535 (mseg)</p>	<p>000A</p>	<p>Esta variable indica el tiempo en milisegundos que transcurren desde la producción de un evento en la entrada de un nodo hasta que verdaderamente se evalúa la actualización de la entrada.</p>
<p>cfgFactorUmb1-2</p>	<p>01</p> <p>01</p>	<p>Esta variable se utiliza para aumentar el tiempo de encendido de la iluminación. El tiempo final</p>

		especificado por la variable <code>cfgTimeoutLuz1</code> o <code>cfgTimeoutLuz2</code> será ese valor multiplicado por <code>cfgFactorUmb1</code> o <code>cfgFactorUmb2</code> . Igual para <code>cfgTimeRele1</code> o <code>cfgTimeRele2</code> , es decir, el tiempo final será <code>cfgTimeRele1</code> o <code>cfgTimeRele2</code> multiplicado por <code>cfgFactorUmb1</code> o <code>cfgFactorUmb2</code> .
NODO ILUMINACIÓN 03 INS – 451		
<code>cfgActuaLocal1-2-3-4</code> Valores: (100,0 1), (0,0 0)	(100,0 1) (100,0 1) (100,0 1) (100,0 1)	Se utiliza para especificar el modo de funcionamiento de los circuitos nº1, nº2, nº3 o nº4. Si está a (100,0 1), la actuación sobre el pulsador de los circuitos correspondientes afectan a la conmutación CC1, CC2, CC3 o CC4. Si se pone a (0,0 0) no hay actuación y la actuación sobre el pulsador no tiene ningún efecto local sobre la salida. Solo vale de cara a la red a través de las variables de salida <code>nvoPulCir1-2-3-4</code> y <code>nvoPulMan1-2-3-4</code> .
<code>cfgInvNivelCir1-2-3-4</code> Valores: (100,0 1), (0,0 0)	(0,0 0) (0,0 0) (0,0 0) (0,0 0)	Esta variable indica la inversión del funcionamiento de los circuitos de conmutación 1, 2,3 ó 4. Si está a (0,0 0) la conmutación es normal. Si se pone a (100,0 1), se invierte la conmutación; es decir, si recibe una activación abrirá los relés y si recibe una desactivación se cortocircuitarán.

cfgInicioSensor1-2 Valores: (100,0 1), (0,0 0)	(0,0 0) (0,0 0)	Se utiliza para poder configurar el modo de encendido automático de la luz de los circuito nº1 o nº2 con el detector de presencia.
cfgRellano1-2-3-4 Valores: (100,0 1), (0,0 0)	(100,0 1) (100,0 1) (100,0 1) (100,0 1)	Esta variable indica si los circuitos de iluminación nº1, nº2, nº3 o nº4 deben funcionar con doble pulsación y encendido automático (100,0 1) o sin doble pulsación de encendido y modo automático (modo rellano, 0,0 0).
cfgTimeoutLuz1-2-3-4 Valores: 0... 6553.4 seg.	(300,0) (600,0) (600,0) (600,0)	Se utiliza para especificar el tiempo (seg) que se debe mantener encendido los circuitos de salida nº1, nº2, nº3 o nº4 sin que exista ninguna detección de presencia.
cfgTimeFiltroSLL Valores: 0... 6553.4 sg.	(60,0)	Esta variable indica cada cuánto tiempo como mínimo debe actualizarse la variable de salida nvoEntLluvia.
cfgModoEstLuzG Valores: 0... 15	0F	Indica qué circuitos se deben conmutar cuando se actualiza la variable de entrada nviEstLuzg.
cfgTimeFiltro Valores: 0.. 65535 (mseg)	000A	Esta variable indica el tiempo en milisegundos que transcurren desde la producción de un evento en la entrada de un nodo hasta que verdaderamente se evalúa la actualización de la entrada.
NODO ILUMINACIÓN 04 INS – 451		
cfgActuaLocal1-2-3-4 Valores:	(100,0 1) (100,0 1) (100,0 1)	Se utiliza para especificar el modo de funcionamiento de los circuitos nº1, nº2, nº3 o nº4. Si está a (100,0

(100,0 1), (0,0 0)	(0,0 0)	1), la actuación sobre el pulsador de los circuitos correspondientes afectan a la conmutación CC1, CC2, CC3 o CC4. Si se pone a (0,0 0) no hay actuación y la actuación sobre el pulsador no tiene ningún efecto local sobre la salida. Solo vale de cara a la red a través de las variables de salida nvoPulCir1-2-3-4 y nvoPulMan1-2-3-4.
cfgInvNivelCir1-2-3-4 Valores: (100,0 1), (0,0 0)	(0,0 0) (0,0 0) (0,0 0) (0,0 0)	Esta variable indica la inversión del funcionamiento de los circuitos de conmutación 1, 2,3 ó 4. Si está a (0,0 0) la conmutación es normal. Si se pone a (100,0 1), se invierte la conmutación; es decir, si recibe una activación abrirá los relés y si recibe una desactivación se cortocircuitarán.
cfgInicioSensor1-2 Valores: (100,0 1), (0,0 0)	(0,0 0) (0,0 0)	Se utiliza para poder configurar el modo de encendido automático de la luz de los circuito nº1 o nº2 con el detector de presencia.
cfgRellano1-2-3-4 Valores: (100,0 1), (0,0 0)	(100,0 1) (100,0 1) (100,0 1) (100,0 1)	Esta variable indica si los circuitos de iluminación nº1, nº2, nº3 o nº4 deben funcionar con doble pulsación y encendido automático (100,0 1) o sin doble pulsación de encendido y modo automático (modo rellano, 0,0 0).
cfgTimeoutLuz1-2-3-4 Valores:	(300,0) (300,0) (300,0)	Se utiliza para especificar el tiempo (seg) que se debe mantener encendido los circuitos de salida

0... 6553.4 seg.	(60,0)	n°1, n°2, n°3 o n°4 sin que exista ninguna detección de presencia.
cfgTimeFiltroSLL Valores: 0... 6553.4 sg.	(60,0)	Esta variable indica cada cuánto tiempo como mínimo debe actualizarse la variable de salida nvoEntLluvia.
cfgModoEstLuzG Valores: 0... 15	0F	Indica qué circuitos se deben conmutar cuando se actualiza la variable de entrada nviEstLuzg.
cfgTimeFiltro Valores: 0.. 65535 (mseg)	000A	Esta variable indica el tiempo en milisegundos que transcurren desde la producción de un evento en la entrada de un nodo hasta que verdaderamente se evalúa la actualización de la entrada.

2.2.4 PLANOS DE PREINSTALACIÓN

En las figuras siguientes se observan los planos de preinstalación domótica, sin embargo se los puede observar con mayor detalle en el ANEXO # 6.

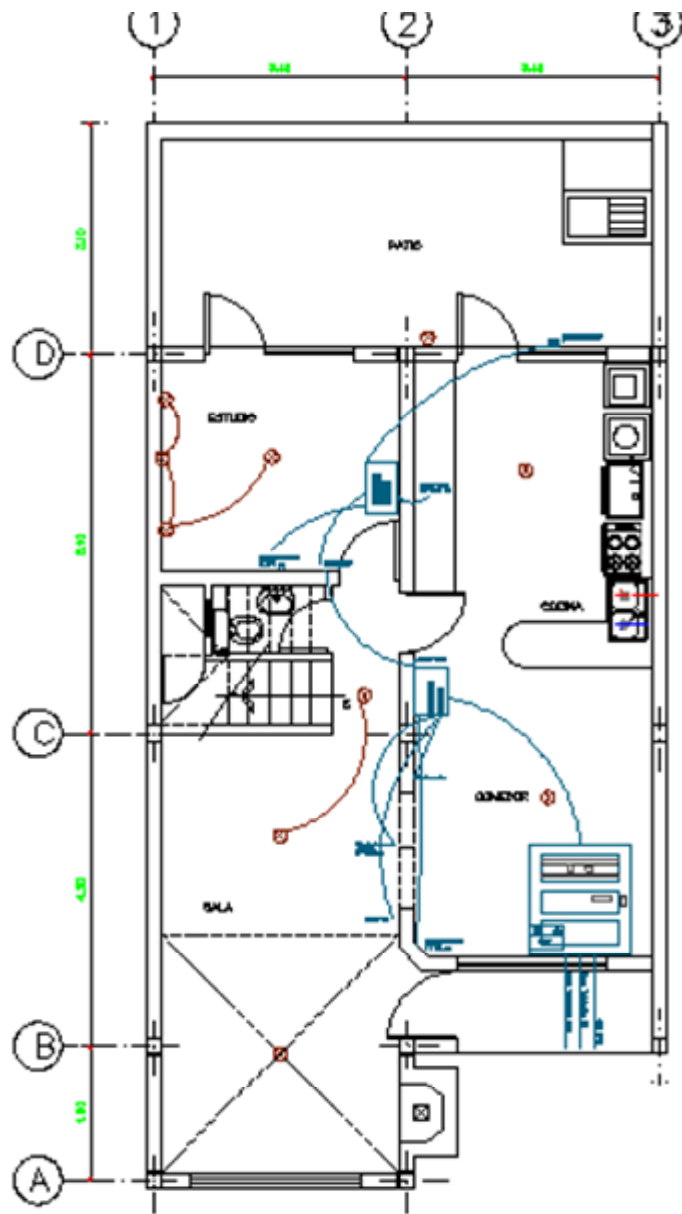


Fig. 52 Plano de preinstalación domótica. Planta baja.

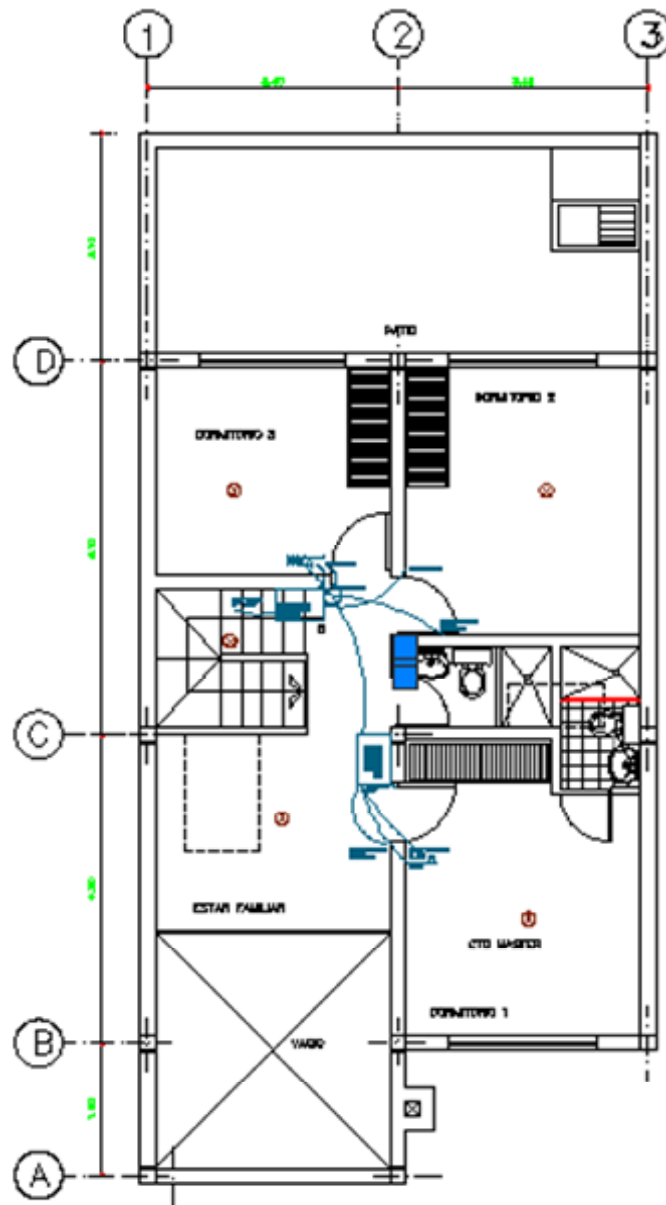


Fig. 53 Plano de preinstalación domótica. Planta alta.

2.2.5 PLANO DOMÓTICO

En las figuras siguientes se observan los planos domóticos, sin embargo se los puede observar con mayor detalle en el ANEXO # 7.

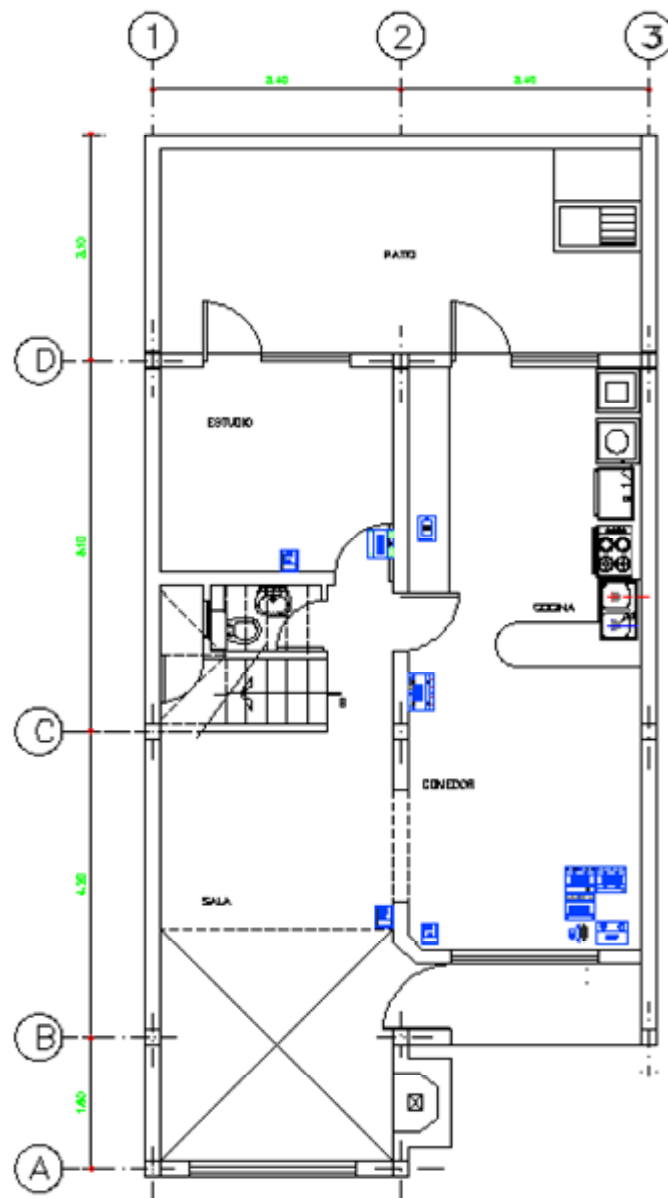


Fig. 54 Plano domótico. Planta baja.

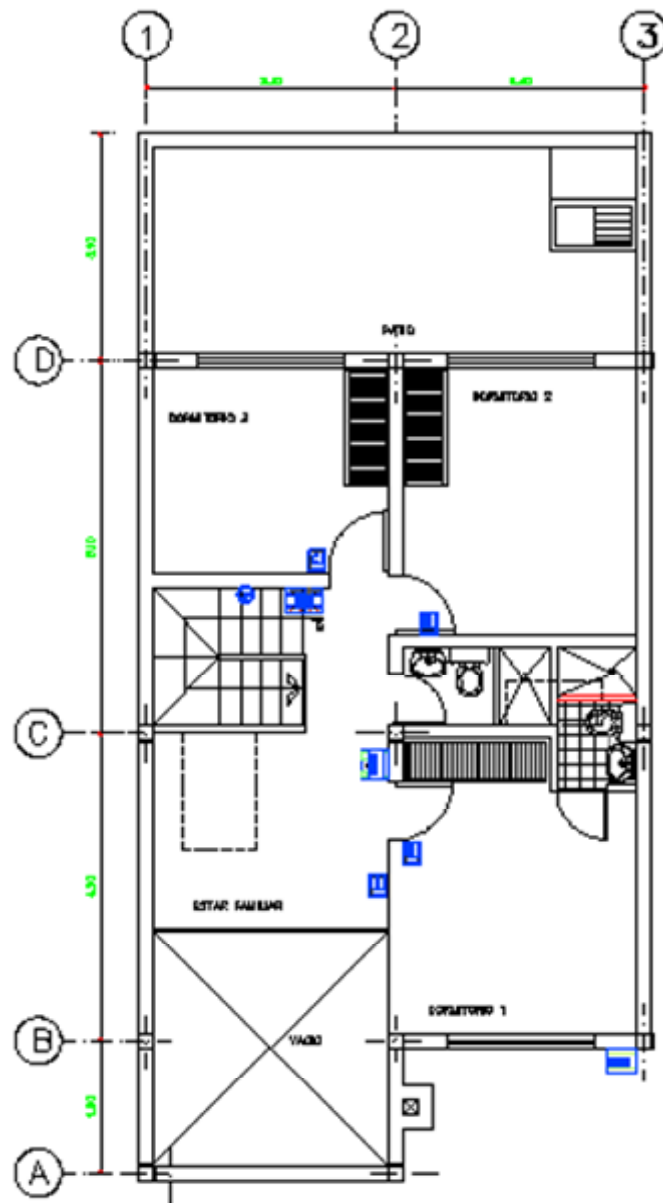


Fig. 55 Plano domótico. planta alta.

2.2.6 PLANOS UNIFILARES

En las figuras siguientes se observan el plano unifilar, sin embargo se los puede observar con mayor detalle en el ANEXO # 8.

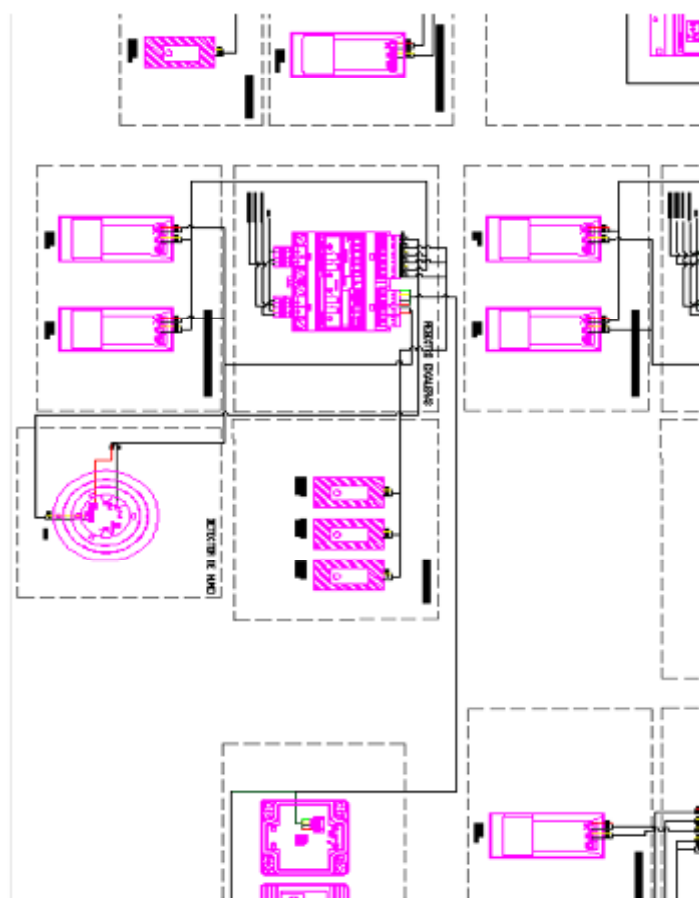


Fig. 56 Plano unifilar.

2.3 OBRA CIVIL E INSTALACIÓN DE NODOS DOMÓTICOS

Para el proceso de domotización de un inmueble es necesario realizar ciertas adecuaciones en la obra civil, por lo cual podemos dividir a este proceso en Preinstalación e Instalación.

En el caso de inmuebles ya construidos se tienen dos posibilidades para la preinstalación del sistema domótico, en primer lugar la “picada” de los muros y en segundo la colocación de canaletas; las dos con el fin de poder colocar los cables de comunicaciones, sensores, poder y eléctricos. Para el presente proyecto se decidió la colocación de canaletas y la utilización de las mangueras ya existentes en el inmueble.

2.3.1 TRANSFORMACIONES Y ADAPTACIONES NECESARIAS PREVIAS A LA AUTOMATIZACIÓN

Las principales transformaciones realizadas en la vivienda son la instalación de las diferentes cajas de empotrar, una al costado de la caja de breakers y cuatro más distribuidas a lo largo de la vivienda, en las estancias *Sala*, *Sala de estar*, y *Estudio*. Para tal efecto fue necesario picar los muros en los puntos especificados por el del inmueble.

Para la colocación de los diferentes cables se ubicó una serie de canaletas ubicadas entre las cajas de empotrar y en ciertos puntos con el fin de ocultar cables eléctricos y/o de sensores.

A continuación se puede observar el agujero realizado para la colocación de la caja principal en la cual irán ubicados los nodos *Fuente*, *Telefónico* y de *Supervisión*. Se puede apreciar el lugar por el cual irán los cables de alimentación desde la central de carga.



Fig. 57 Agujero de la Caja principal.

En la siguiente fotografía se observa uno de los agujeros realizados para la colocación de una de las cajas de registros. Específicamente se trata de la caja de registro de la estancia *Escalera*.



Fig. 58 Agujero para la Caja de registro de la estancia *Escaleras*.

Para minimizar el impacto sobre la obra civil se procuró la utilización de las diferentes tuberías eléctricas instaladas por donde antes se ubicaba el cable del retorno eléctrico proveniente de los interruptores y en donde ahora se ubicarán los cables de sensores. A la vez se reemplazaron enteramente los interruptores por pulsadores.



Fig. 59 Agujero para el aprovechamiento de las tuberías instaladas.

2.3.2 INSTALACIÓN DE LOS DISTINTOS NODOS LONWORKS, SENSORES Y ACTUADORES NECESARIOS

Los nodos LonWorks fueron dispuestos de la siguiente manera, según las diferentes estancias del hogar tipo:

Tabla # 11 Distribución de nodos en las distintas estancias.

DISTRIBUCIÓN DE NODOS		
NODOS	ESTANCIAS	CAJA DE REGISTRO
Fuente de alimentación IFA – 200	Sala	Principal
Integral de supervisión INM – 011	Sala	Principal
Telefónico sin domoportero INM – 020	Sala	Principal
Medidor de luz INM – 030	Patio	Patio
Estándar INS – 231	Sala de estar	Sala de estar
Estándar INS – 231	Estudio	Estudio
Avanzado INS – 451	Comedor	Comedor
Avanzado INS – 451	Escaleras	Escaleras

Se colocaron un total de 7 sensores de movimiento, 1 sensor de detección de gas, un sensor de detección de humo, 11 pulsadores, una bocina, una batería de respaldo y la conexión telefónica interior según la siguiente distribución:

Tabla # 12 Distribución de los sensores, actuadores y demás dispositivos en los nodos.

DISTRIBUCIÓN DE LOS SENSORES, ACTUADORES Y DEMÁS DISPOSITIVOS	
SENSORES, ACTUADORES Y DEMÁS DISPOSITIVOS	NODOS
1 Batería de respaldo.	Fuente de alimentación IFA – 200
1 Bocina.	Integral de supervisión INM – 011
Línea telefónica.	Telefónico sin domoportero INM – 020
Ninguno.	Medidor de luz INM – 030
2 Sensores de movimiento.	Estándar INS – 231. Sala de estar.

2 Pulsadores.	
1 Sensor de movimiento. 1 Sensor de detección de gas.	Estándar INS – 231. Estudio.
2 Sensores de movimiento. 3 Pulsadores.	Avanzado INS – 451. Comedor.
3 Sensores de movimiento. 4 Pulsadores. 1 Sensor de detección de humo.	Avanzado INS – 451. Escaleras.

La batería utilizada es la VISIÓN CP 1270, la cual es una batería recargable de plomo – dióxido de plomo que utiliza ácido sulfúrico como electrolito. Está batería es de 12V y produce 7 A/h. Las características técnicas más específicas de la batería y demás sensores y/o actuadores ver en el ANEXO #9.

Para observar el detalle de la ubicación de los sensores y actuadores se puede acceder a los Planos de Preinstalación en el ANEXO # 6.

2.3.3 ANÁLISIS DE LAS MEDICIONES DEL MEDIDOR DE ENERGÍA ELÉCTRICA

Para el análisis del ahorro energético se utilizó la *Historia de facturaciones* que se encuentra en el ANEXO # 10. Sin embargo es importante anotar que difícilmente se puede realizar un análisis en base a una línea de tendencia a lo largo de un año calendario, ya que cada época de este tiene características especiales que modifican el consumo típico de energía eléctrica, como por ejemplo en los meses de noviembre y diciembre en donde existirá un inminente aumento del consumo. También hay que tener especial atención en las fechas de corte de las planillas por parte de la Empresa Eléctrica

Quito, ya que existen meses en los cuales se realizó la medición el día 19 y en otros el 24, lo cual da un margen de error de 5 días.

Con el análisis anterior se concluye que lo correcto es comparar meses iguales de año a año, específicamente de septiembre a diciembre del 2009 y enero del 2010 con los meses de septiembre a diciembre del 2010 y enero del 2011. En la siguiente tabla y el correspondiente gráfico se encontrará que hubo una disminución del consumo eléctrico en el orden del 23,2%.

Tabla # 13 Ahorro energético por mes.

FECHAS				AHORRO ENERGÉTICO
Septiembre 2009	276	Septiembre 2009	204	26,1%
Octubre 2009	300	Octubre 2010	198	34,0%
Noviembre 2009	236	Noviembre 2010	201	14,8%
Diciembre 2009	276	Diciembre 2010	214	22,5%
Enero 2010	258	Enero 2011	202	21,7%
Ahorro energético promedio				23,2%

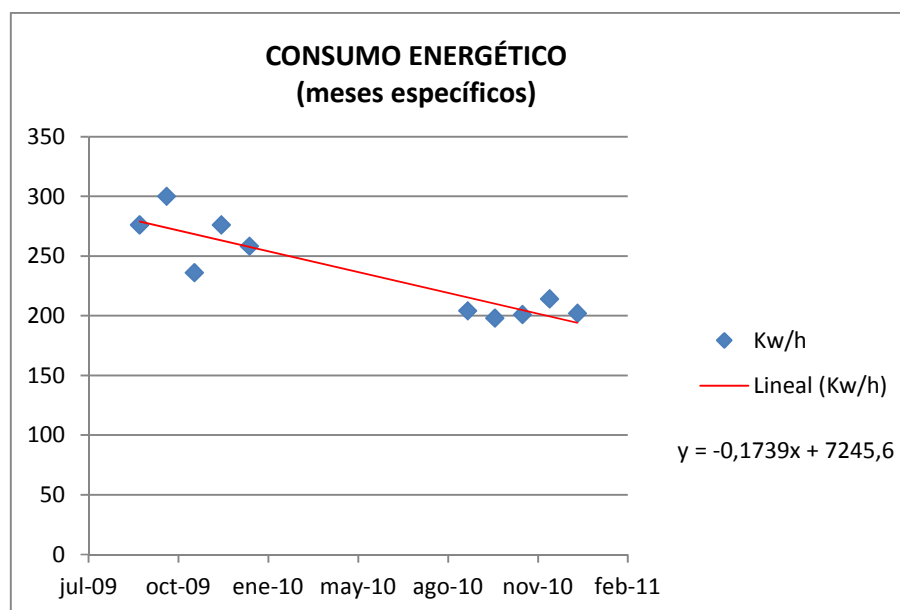


Fig. 60 Histórico del consumo energético. Comparación entre meses correspondientes al mismo periodo.

2.3.4 ANÁLISIS DE COSTOS, INCLUYENDO OBRA CIVIL

Para poder realizar un análisis del costo de la instalación del sistema domótico es necesario detallar una lista de los distintos materiales utilizados en la misma y descritos en la siguiente tabla:

Tabla # 14 Detalle de materiales utilizados en el sistema domótico.

MATERIALES UTILIZADOS EN EL SISTEMA DOMÓTICO				
DETALLE		Cantidad	PU	Precio final
Preinstalación	Caja principal de registro	1	\$ 25,00	\$ 25,00
	Cable de sensores (m)	100	\$ 0,17	\$ 17,00
	Cable de bus (m)	100	\$ 1,63	\$ 163,00
	Terminación de red	2	\$ 35,67	\$ 71,34
	Canalizaciones (m)	40	\$ 2,45	\$ 98,00
	Cajas de registros	3	\$ 12,00	\$ 36,00
	Pulsadores	11	\$ 3,50	\$ 38,50
Cuadro de control	Nodo integral de supervisión	1	\$ 328,15	\$ 328,15

	Nodo telefónico sin domoportero	1	\$ 358,75	\$ 358,75
	Nodo Fuente de alimentación	1	\$ 409,29	\$ 409,29
	Batería 12 V	1	\$ 12,00	\$ 12,00
	Sirena interior	1	\$ 13,00	\$ 13,00
Nodos de control	Nodo de control estándar 231	2	\$ 201,47	\$ 402,94
	Nodo de control avanzado 451	2	\$ 269,15	\$ 538,30
Sensores	Detector de gas	1	\$ 110,58	\$ 110,58
	Detector de humo	1	\$ 89,23	\$ 89,23
	Detector de presencia de pared	7	\$ 48,59	\$ 340,13
	Nodo medidor de luz exterior	1	\$ 261,96	\$ 261,96
Otros	Interface USB a FTT - 10	1	\$ 397,33	\$ 397,33
TOTAL				\$ 3710,50

Para la preinstalación fue necesaria la contratación de un albañil, además del instalador, como se detalla en la siguiente tabla:

Tabla # 15 Detalle de la mano de obra utilizada en el sistema domótico.

MANO DE OBRA INVOLUCRADA			
DETALLE	Cantidad (Horas de trabajo)	PU	Precio final
Albañil	12	\$ 2,50	\$ 30,00
Instalador	8	\$ 6,25	\$ 50,00
Programador HMI	50	\$ 20,00	\$ 1 000,00
TOTAL			\$ 1080,00

En el caso específico de este trabajo las horas de trabajo invertidas en la instalación del sistema domótico se encuentran estimadas al caso de un albañil, instalador y programador experimentados, ya que en la realidad el

tiempo necesario fue mucho mayor debido a las diversas pruebas y tests que se realizaron para observar el correcto funcionamiento del sistema.

La suma de los diferentes valores anteriormente mencionados señala una inversión total de \$4790,50.

CAPÍTULO III

DISEÑO DEL HMI

3.1 COMPATIBILIDAD CON LONWORKS

3.1.1 DDE (Dynamic data exchange).

O intercambio dinámico de datos. Creado por Microsoft. Es un protocolo usuario-servidor para intercambio de datos entre aplicaciones, con el paso del tiempo Microsoft lo sustituyó por COM y OLE pero a pesar de eso, se utiliza siempre en algunas partes de Windows.

DDE permite abrir una sesión con otra aplicación, enviar un pedido a la aplicación que trabaja como servidor y recibir una respuesta. No permite incorporar el interfaz gráfico de la aplicación servidor en la aplicación cliente. Para poder utilizar correctamente DDE, la aplicación cliente debe conocer los pedidos disponibles sobre la aplicación servidor que generalmente no se estandarizan.

En este caso se utilizará el DDE para establecer una comunicación entre la red LonWorks y el HMI desarrollado en Visual BASIC.

3.1.2 LNS SERVER

El sistema operativo de red LNS® proporciona un conjunto completo de herramientas de software que permite a múltiples aplicaciones de red realizar una amplia gama de servicios más sobre LONWORKS® y las redes IP. Estos servicios incluyen gestión de redes (instalación de red, configuración, mantenimiento y reparación), el monitoreo y control de todo el sistema.

3.1.3 LNS DDE Server

El LNS DDE Server funciona como un controlador para varias aplicaciones HMI y SCADA tales como Wonderware InTouch ®. Cualquier operador basado en Windows o una aplicación SCADA que apoye el protocolo de Intercambio Dinámico de Datos (DDE) funciona muy bien con el LNS DDE Server. Cuando se combina con InTouch, el LNS DDE Server utiliza DDE Wonderware SuitLink saltando al DDE haciendo de esta una solución de ultra alto rendimiento, en lo referente al seguimiento y solución de control de redes LonWorks.

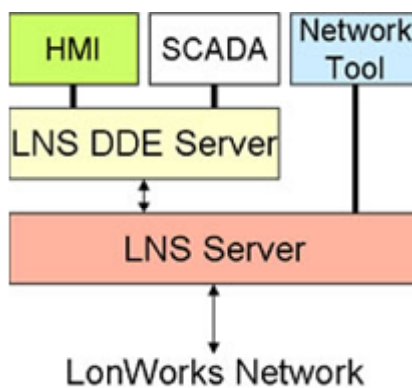


Fig. 61 Funcionamiento de LNS Server.

Características

- Leer, controlar y modificar el valor de cualquier variable de red.

- Supervisar y cambiar las propiedades de configuración.

- Recibir y enviar mensajes de aplicación.

- Probar, activar, desactivar, y reemplazar los objetos LonMark.

- Probar, wink, y controlar dispositivos.

- Compatible con LNS 3 o LNS Turbo Edition bases de datos (data base).

- Conecta con hasta 100 bases de datos (LNS) al mismo tiempo.

3.1.4 ANÁLISIS DE COSTOS DE LICENCIAS

De acuerdo a lo anteriormente descrito se evidencia que son requeridos dos programas para que funcione el HMI: la aplicación LNS Server, y el controlador LNS DDE Server, la aplicación LNS Server viene incluida en el LNS DDE Server, el costo de la licencia de tal programa se encuentra en el internet en un precio actual de 862.65 € lo cual varía de acuerdo a la cotización del Euro y costo del envío, el programa fue comprado a ISDE-ECUADOR a un precio aproximado de 1000USD lo que implica un costo inferior en la licencia, para esta aplicación y con las consideraciones del caso.

La herramienta de integración empleada para instalar la red fue el programa InstallLON_V1_0_2, facilitado por la empresa ISDE-ECUADOR, debido a que la programación de la red se realizó utilizando LonMaker.

Para que la programación del HMI pueda ser realizado, de tal modo que se pueda integrar la red con el usuario, es requerido un programa, de entre los recomendados por Echelon (creadores del protocolo LonWorks, de LNS DDE Server y LNS Server) está InTouch, el mismo que como ya se mencionó al inicio de la sección 4.2 es diseñado precisamente con esta finalidad, por lo cual se han averiguado los precios de la licencia de este programa, siendo conocido que el valor económico a cancelar por las licencias más básicas que se podría tener bordea los 3500USD, lo cual en caso de requerir su implementación implicaría una fuerte inversión para el integrador, y debería tomarse en cuenta la frecuencia de utilización que posteriormente dará al programa.

Como también se revisó al inicio de la sección 4.2 el programa Visual Basic a pesar de no ser una solución específica, al ser un lenguaje de aplicaciones, que permite desarrollar ejecutables con una interfaz visual totalmente entendible para el usuario, y comunicación con DDE, es una herramienta muy útil a considerar para este proyecto. Visual Basic no se vende independientemente sino que es vendido en un entorno de desarrollo integrado llamado Visual Studio. Las licencias para Visual Studio 2010 edición Profesional cuesta alrededor de 1300USD y en una de las versiones más simples llega a costar aproximadamente 700USD, por lo cual en caso de implementarse, debe considerarse los valores aproximados de cada software.

3.2 SELECCIÓN DEL PROGRAMA A UTILIZAR

En base a las facilidades de utilizar software de prueba en cuanto al programa Visual Basic para realizar el HMI a implementar, debido a que permite una solución Interactiva, visual y a medida del Usuario; considerando además que de requerir las licencias para una implementación futura en el caso de un negocio, y habiendo verificado los precios de los softwares más cercanamente utilizable como es el programa InTouch de Wonderware y el programa Visual Basic. Los mismos que pueden comunicarse con el LNS DDE Server. Se han considerado las importantes diferencias de costos, además de la versatilidad que tiene el hecho de usar un programa como Visual Basic en cuanto a todo lo que permite realizar, como todos los lenguajes que vienen en el entorno de desarrollo integrado Visual Studio como Visual C, java, etc. Se ha concluido que para este diseño e implementación se utilizará el software Visual Basic.

3.3 REQUERIMIENTOS DEL HMI

El HMI que se va a diseñar e implementar se lo desarrollará como se mencionó utilizando el software Visual Basic 6.0 comunicándolo a la red LonWorks mediante el protocolo DDE utilizando el programa LNS DDE Server versión 2.1 utilizando el LNS Server que viene incluido como herramienta del programa LNS DDE Server.

El HMI en cuestión deberá cumplir satisfactoriamente con los requerimientos del usuario y deberá ser diseñado y ejecutado en un sistema operativo Windows que sea capaz de soportar todos los programas que se requiere que funcionen en la misma máquina para que todo trabaje de manera adecuada, como de la máquina con las aptitudes requeridas por los creadores de los programas a ser utilizados.

3.3.1 REQUERIMIENTOS DEL USUARIO

El usuario pidió un software mediante el cual se pueda visualizar y controlar mediante software todos los elementos que se los puede visualizar y controlar físicamente estos son:

- El momento del día y su luminosidad, tanto en luxes como visualmente, si es de día o de noche y si es un día nublado, además de poder establecer el nivel de luminosidad al que el comportamiento de las luces asumirá que es la noche y funcionarán automáticamente con los sensores de presencia.
- Luces, es decir aquellas que están consideradas dentro del sistema domótico, que son diez. Su encendido/apagado utilizando el software y la visualización de su estado en pantalla.
- Capacidad de programar una secuencia de encendido/apagado de luces a conveniencia del usuario para simular presencia, o acorde a determinadas tareas, se permitirá tres horas de encendido/apagado por luz.
- Visualización de activación/desactivación de Alarmas de Intrusión para dos áreas de la primera planta que cubren los posibles lugares de acceso al domicilio, ubicados en la sala y el comedor, debe visualizarse el lugar en el que hubo intrusión (sensor de presencia de la sala o el comedor), y un sonido que advierta la presencia, además de poder desactivar utilizando el software la alarma y poder modificar el tiempo de activación y el de ingreso.
- Se requiere que exista la posibilidad de activar y desactivar la simulación de presencia mediante software.

- Control y visualización de alarmas de gas y humo con un sonido diferente al de las alarmas de intrusión para poder identificar que se ha producido una detección de fuga de gas o una detección de humo, especificándose gráficamente la alarma respectiva. Debe existir la posibilidad de desactivar la alarma mediante software.

- Dos niveles de acceso con clave, la misma que puede ser cambiada a conveniencia del dueño de casa una vez que ha ingresado la contraseña correcta que funcione actualmente, de modo que solamente quienes tengan el nivel de administrador, una vez ingresada la clave actual en el programa puedan modificarla.

3.3.2 REQUERIMIENTOS DEL SISTEMA OPERATIVO Y EQUIPOS

De acuerdo a la página de soporte de Microsoft los requerimientos para instalar y operar el programa Visual Basic 6.0 son:

- PC con procesador 486DX/66-MHz o superior (procesador Pentium o superior recomendado).

- Microsoft Windows 95 o posterior sistema operativo o del sistema operativo versión 4.0 con Service Pack 3 o posterior (se incluye el Service Pack 3) de Windows NT.

- 16 MB de RAM para Windows 95 (se recomienda 32 MB); 24 MB para Windows NT 4.0 (se recomiendan 32 MB).

- Microsoft Internet Explorer 4.01 Service Pack 1 (incluido).

- Espacio libre en disco para Instalación máxima (94MB) y espacio para instalación del MSDN máximo (493MB).

Para el programa LNS DDE Server versión 2.1 los requerimientos son:

- Microsoft Windows 2000, Windows ME, Windows 98, o Windows NT 4.0 (Service Pack 3 o superior), Windows XP.
- Pentium 133 o superior.
- 20MB libres en el disco duro.
- 64MB de RAM como mínimo.

La máquina donde se desea instalar los programas requeridos posee las siguientes características:

- Intel® Core™2DuoCPU.
- T7500 @2200GHz.
- 2.20GHz 750MB de RAM.
- Windows XP Service Pack 3.
- Más de 10GB de espacio libre en disco.

Por lo tanto el equipo donde se desea instalar los programas mencionados supera ampliamente los requerimientos del sistema, lo que hace completamente factible su instalación.

3.4 PROGRAMACIÓN DEL HMI

Para poder empezar a desarrollar el Software, es necesario primero un interfaz físico utilizando un componente de Hardware con un controlador asociado a la red, que provea la conexión física entre la red LonWorks y la PC al ejecutar el LNS DDE Server. En este caso se utilizará el interfaz de red U10 USB Network Interface-TP/FT-10 Channel que funciona como se mencionó con conexión USB y no requiere instalación previa.

Se requiere previamente haber instalado la red, utilizando una herramienta de instalación como el LonMaker™ Integration Tool, o el Installon. La instalación de una red de control es el proceso de carga de la personalidad única de red de cada dispositivo en sus tablas internas.

Este proceso permite que cada dispositivo se comuniquen con otros dispositivos en el mismo sistema.

La herramienta de instalación configura los dispositivos en la red y almacena una copia de esta información en el LNS Server. Cuando finalice la instalación, el LNS Server contiene una imagen de la configuración de toda la red junto con los nombres de todos los dispositivos y las variables de red, las propiedades de configuración, y objetos LonMark (también llamados bloques funcionales) en los dispositivos. El LNS DDE Server utiliza la información en el LNS Server para cumplir sus Tareas.

De modo que para que el programa realizado en Visual Basic funcione se utilizara el protocolo DDE para comunicarse con los dispositivos, y es necesario que previamente se ejecute el LNS Server y posteriormente el LNS DDE Server y se seleccione la red en la que se va a trabajar de modo que cuando se pida comunicación DDE el programa funcione sin problema alguno.

Para empezar a realizar la programación y enlazar los elementos con el DDE, en primer lugar es necesario instalar la red que se ha creado para poder trabajar en ella, para lo cual es necesario primero conectar la interfaz física (U10 USB Network Interface-TP/FT-10 Channel) que es identificada por el sistema como “LON3”. Posteriormente debe copiarse en la dirección “C:\Lm\Db”, la carpeta que contiene la red que se ha creado utilizando LonMaker, en este caso corresponde a la red llamada “MoyaAlmeida1”. Una vez realizado esto, se la tiene que instalar utilizando una herramienta de instalación que en este caso será el programa InstalLON_V_1_0_2, para lo cual es necesario abrir el programa, presionar el botón “RESTAURACION RED”, elegir la red que está ubicada en la dirección mencionada anteriormente (C:\Lm\Db).



Fig. 62 Ingreso al programa InstalLON_V_1_0_2.



Fig. 63 Ingreso al programa InstallLON_V_1_0_2.

Una vez seleccionada la red es necesario presionar el botón “RESTAURAR_DB”, para que la red sea instalada en el sistema. Esto permite instalar la red en el sistema que estamos trabajando de modo que sea posible operar en dicha red.



Fig. 64 Ingreso al programa InstallLON_V_1_0_2.

Luego de haber instalado la red, es posible salir del programa InstallLON presionando el botón “SALIR” ya que la red ha sido instalada y ahora es

reconocida. Después de esto, es necesario abrir el servidor LNS Server y seleccionar la interfaz física utilizada, en este caso corresponde a “LON3”:

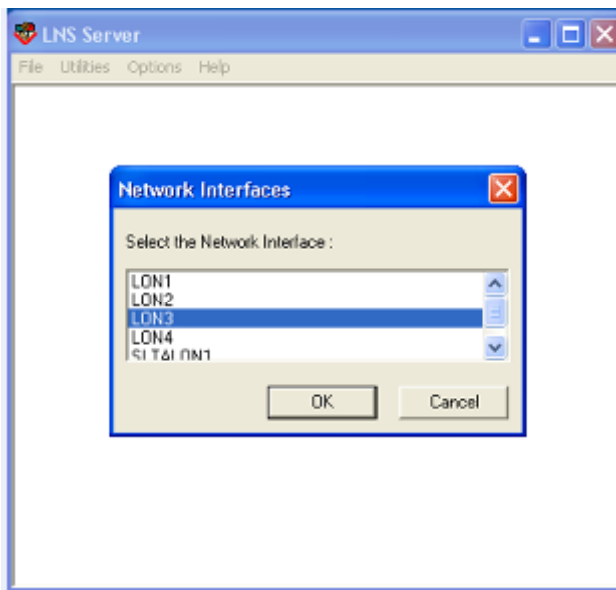


Fig. 65 Selección de la Interfaz física de red.

Una vez abierta la red con el LNS Server es necesario abrir el LNS DDE Server y seleccionar la red que se desea controlar, esto permite además visualizar todas las variables que se puede controlar y sus estados.

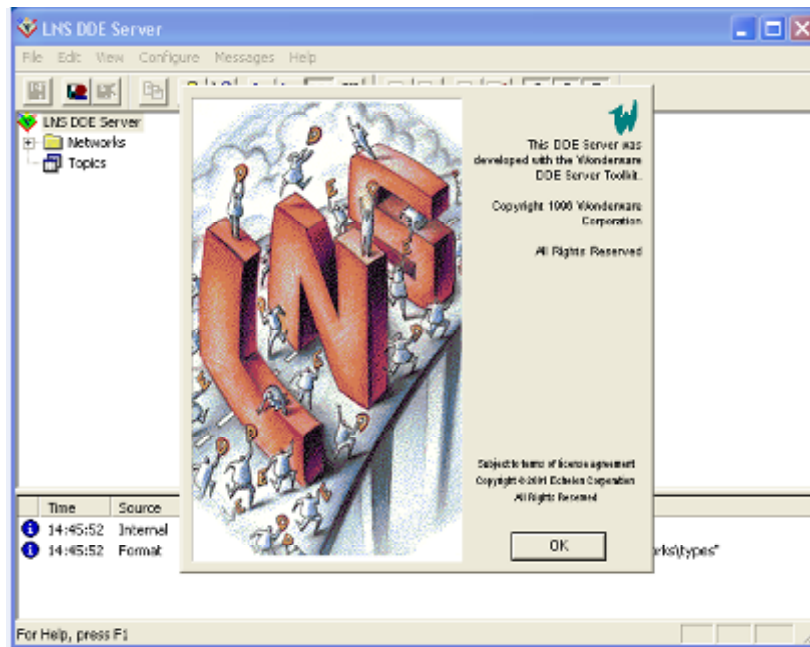


Fig. 66 Ingreso al LNS DDE Server.

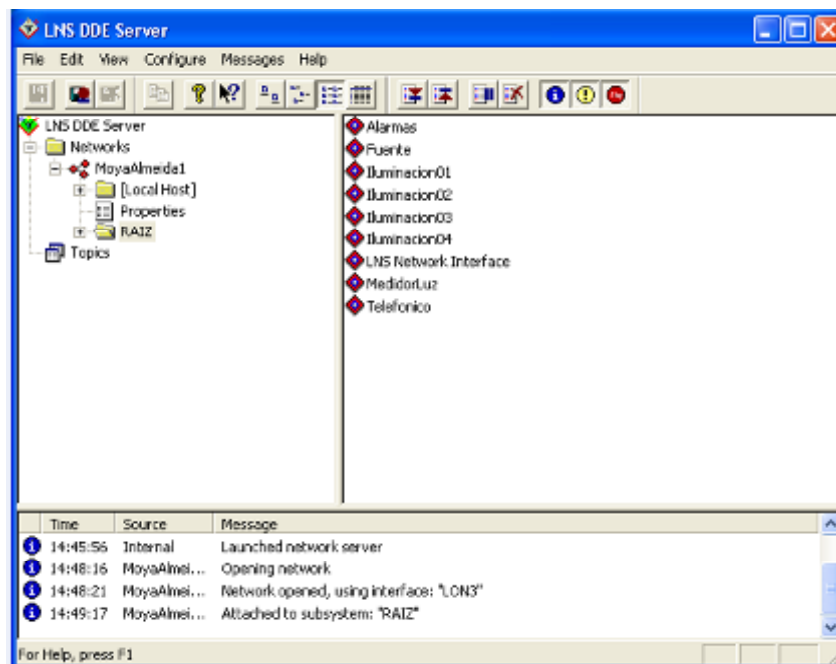
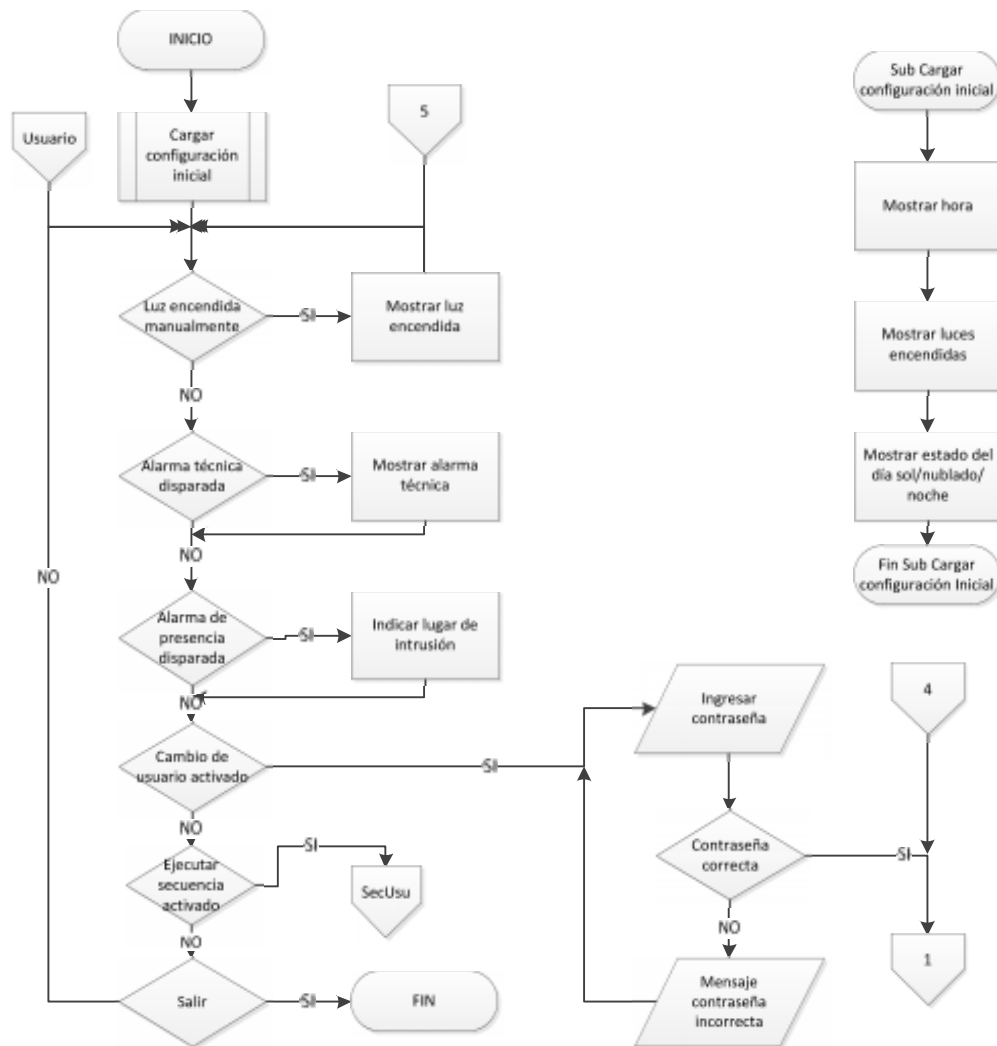
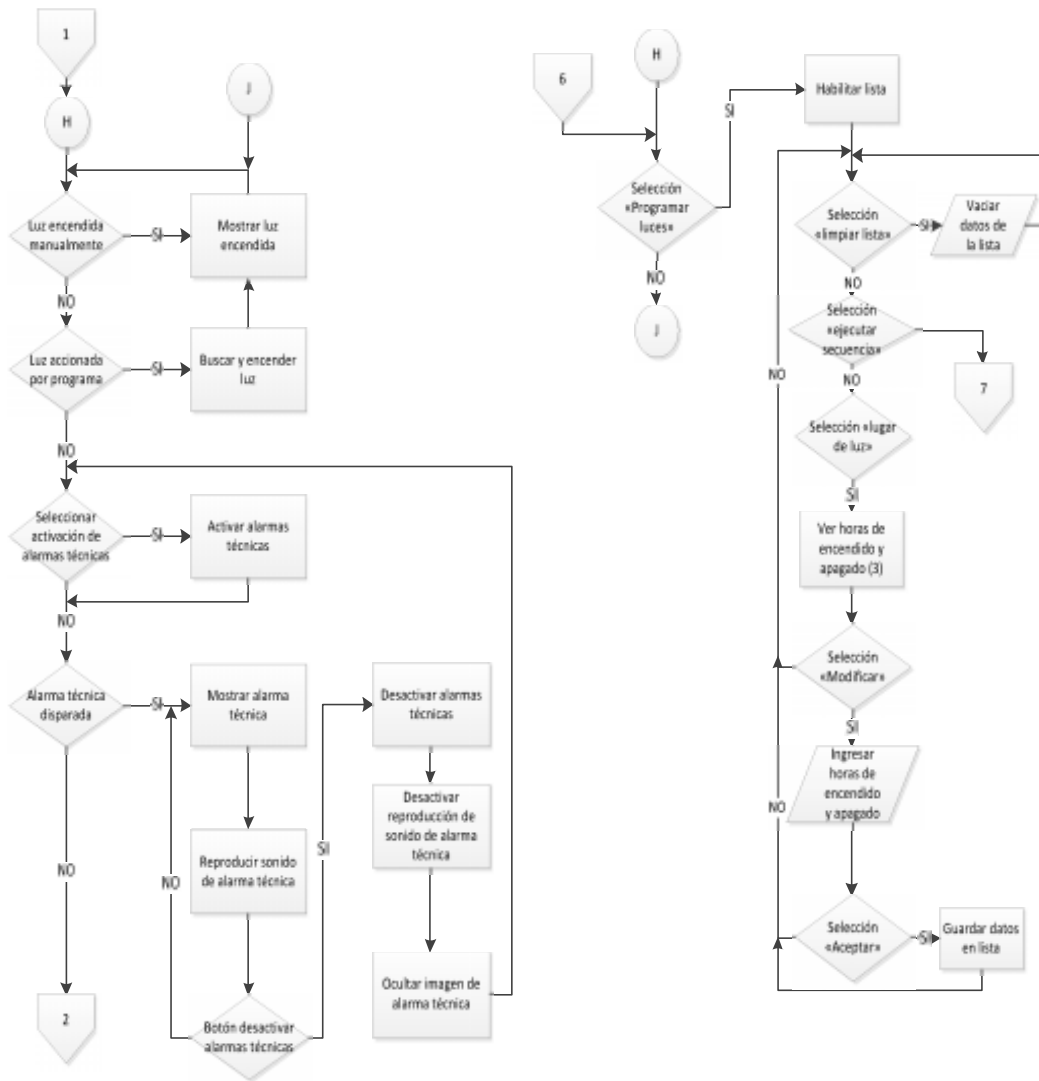


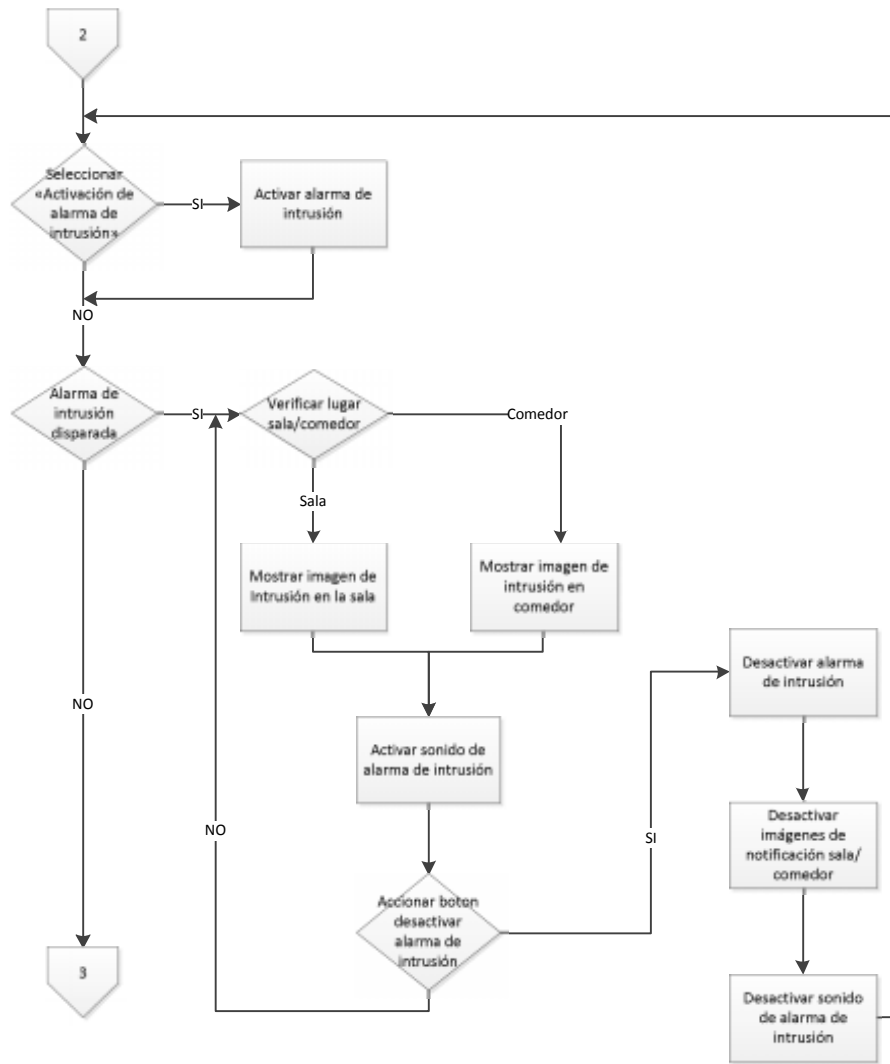
Fig. 67 Selección de la red a controlar.

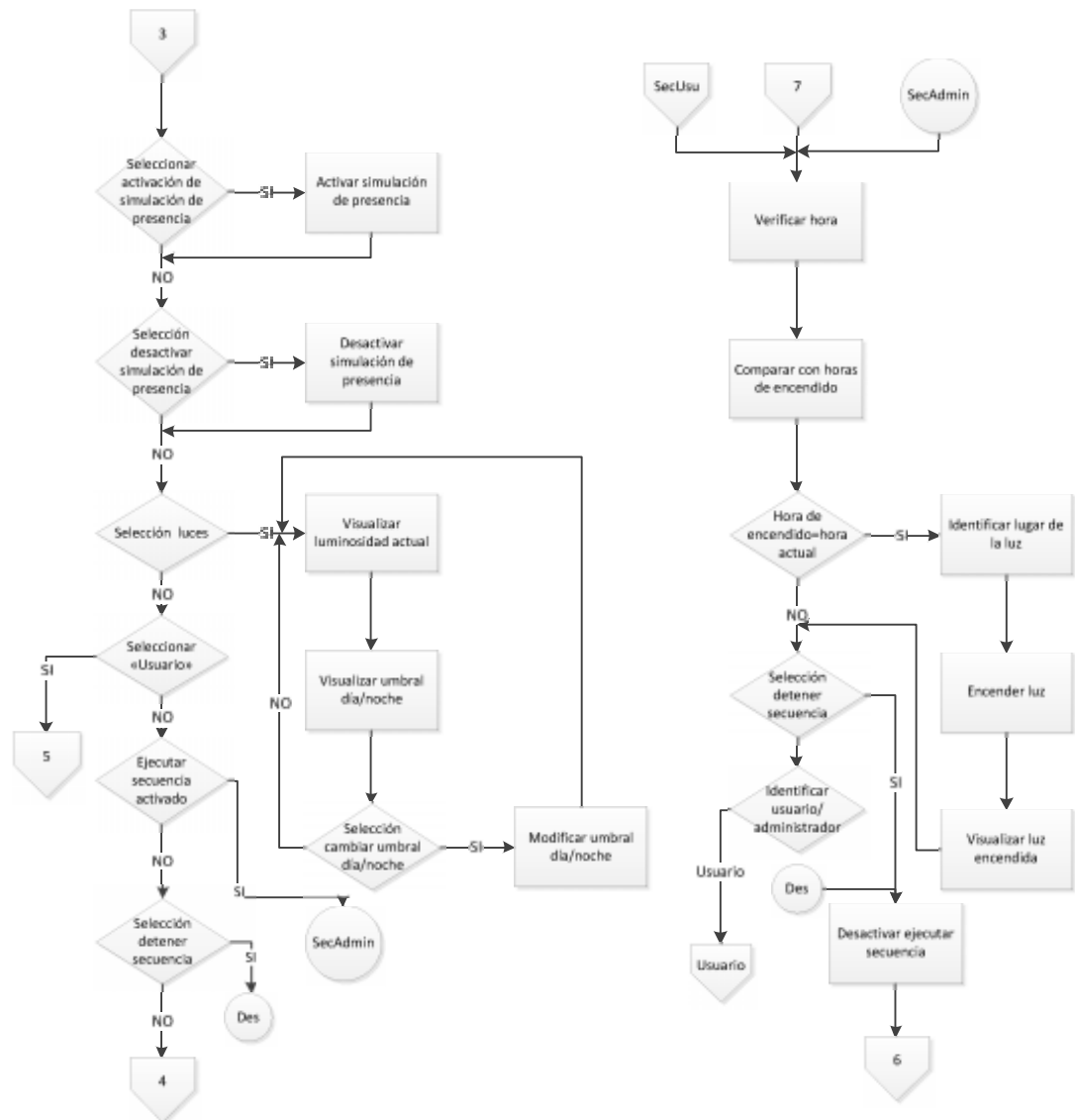
Una vez establecida la comunicación DDE se puede comenzar con la programación del HMI.

3.4.1 DIAGRAMA DE FLUJO









En base a este diagrama de flujo como representación general de lo que necesita el programa se puede comenzar la programación tomando en cuenta todo lo mencionado anteriormente.

3.4.2 IMPLEMENTACIÓN DE NIVELES DE ACCESO DE ACUERDO A LOS REQUERIDOS POR EL DUEÑO DE CASA

De acuerdo a lo requerido por el dueño de casa, considerando que es una vivienda familiar en donde habitan miembros de un hogar conformado por

padre, madre y tres hijos, de los cuales dos son adultos y una niña; siendo que forman parte de un núcleo familiar, las personas que pueden operar el sistema requieren tener el mismo nivel de acceso a todas las funciones del mismo, tanto en el ámbito de monitoreo y visualización, como de control de luces, programación de las mismas y gestión de alarmas. En caso de que alguien que no sea de la familia tenga acceso al programa y lo pueda operar, solamente podrá visualizar algunos campos.

Por lo cual se requieren dos niveles de acceso, los mismos que serán de usuario, y administrador, cada uno con las siguientes características:

El nivel de acceso “Usuario” se refiere a cualquier persona que ingresa a la vivienda, tiene acceso a la PC donde está instalado el programa y al manejo del mismo; pero al no ser miembros del hogar que habita la vivienda, solamente tendrán las siguientes opciones al manejar el programa:

- Visualización del estado del día (sol, nublado, noche).
- Visualización de luces encendidas y apagadas, y de la acción de encendido/apagado físico de las mismas desde la casa, en tiempo real.
- Sonido de alarma de intrusión y presentación visual del lugar donde se suscitó la intrusión, sea en el sensor ubicado en la sala o el del comedor.
- Sonido de alarmas técnicas de humo y gas con representación gráfica para cada una en la ventana principal del programa, con el fin de saber cuál sea específicamente el problema técnico, diferenciando gas, y humo.

El nivel de acceso “Administrador” corresponde a todos los miembros de la familia que el dueño de casa considere facilitar la clave de acceso al nivel, toda persona que posea la clave de acceso podrá utilizar todas las opciones que el programa ofrece en cuanto a monitoreo y control como son:

- Visualización del estado del día (sol, nublado, noche), en la ventana principal y en la pestaña “medidor de luz” correspondiente a la ventana de gestión de alarmas.

- Visualización de luces encendidas y apagadas, y la acción de encendido/apagado mediante software de las en tiempo real.

- Sonido de alarma de intrusión y presentación visual del lugar donde se suscitó la intrusión, sea en el sensor ubicado en la sala o el del comedor, activación y desactivación de las mismas.

- Sonido de alarmas técnicas de humo y gas con representación gráfica para cada una en la ventana principal del programa, con el fin de saber cuál sea específicamente el problema técnico, diferenciando gas, y humo, con su respectiva capacidad de activar y desactivar las alarmas técnicas.

- Programación para encendido automático.

- Cambio de contraseña de administrador.

Para implementar los niveles de acceso del programa, es decir lo que corresponde al ingreso de claves y cambio de usuario, se desarrolló en un formulario o “form” que es una ventana de interfaz de usuario en una aplicación, al que se le ha asignado el nombre “clave_acceso”, en el cual, se encuentran todas las opciones para los niveles de acceso mencionados anteriormente: usuario y administrador:

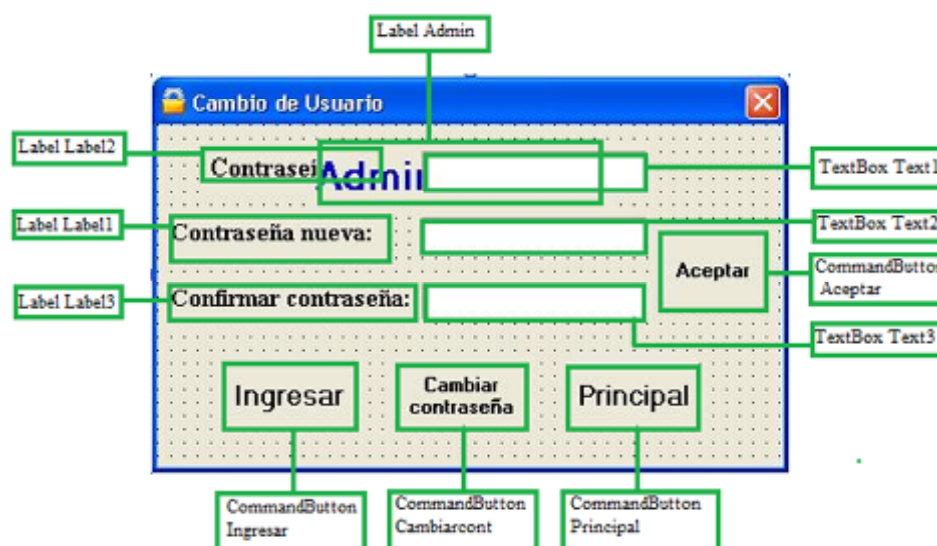


Fig. 68 Formulario clave_acceso.

La imagen presentada, donde se visualiza el formulario “clave_acceso” está compuesta por los siguientes elementos:

- Cinco botones (Command_Button); “Ingresar”, “Usuario”, “Cambiar Contraseña”, “Principal” y “Aceptar”
- Tres cajas de texto (TextBox).
- Cuatro etiquetas (Label)

Los botones de comando o “Command Button” como su nombre lo indica, son utilizados para realizar acciones en base al evento que se esté utilizando.

Las cajas de texto o “TextBox”, son utilizadas para ingresar texto, o para monitorear su variación en tiempo de ejecución del programa.

Las etiquetas o “Label”, son utilizadas mayormente para dar avisos o poner títulos, debido a que pueden variar su tamaño color, apariencia forma, etc.

Las propiedades más trascendentes empleadas fueron:

- **Nombre.** Que identifica en el código determinado objeto.
- **Caption.** Que devuelve o establece el título mostrado en el objeto, es el título que se visualiza al momento de ejecutar el programa.
- **Visible.** Una herramienta que devuelve o establece un valor que determina si un objeto es visible o está oculto.

Se utilizaron los siguientes eventos:

- “Form_load”, al cargarse el formulario, el momento que se accede a este, se realizan determinadas acciones, que permiten ingresar al formulario con cierta apariencia, y establecidos los controles en determinados valores, en este caso el código es el siguiente:

```
“ Private Sub Form_Load()  
  If contusu = 1 Then  
    Label2.Visible = False  
    Text1.Visible = False
```

```
Admin.Visible = True
Ingresar.Visible = False
Usuario.Visible = True
Cambiarcont.Visible = True
End If
If contusu = 2 Then
Label2.Visible = True
Text1.Visible = True
Admin.Visible = False
Ingresar.Visible = True
Usuario.Visible = False
Cambiarcont.Visible = False
End If
End Sub”
```

Esta forma de cargar el formulario permite que una vez accedido a este objeto, se analice la variable global “**contusu**”, la misma que permite que se visualice de dos maneras distintas dependiendo si se trata de un usuario común o si se trata del administrador. Cuando es la primera vez que se ingresa esta variable no tendrá ninguno de los valores que modifican la apariencia del formulario (“1” o “2”), por lo que se verá de la forma que se estableció las propiedades de los objetos inicialmente, y que es similar a la visualización que el formulario ofrecerá en caso de que se reconozca al operador como usuario. En esos casos, el formulario se carga de la siguiente manera:

A screenshot of a Windows-style dialog box titled "Cambio de Usuario". The dialog has a blue title bar with a close button (X) in the top right corner. The main area has a light beige background. It contains a label "Contraseña:" followed by a white text input field. Below the input field, there are two buttons: "Ingresar" on the left and "Principal" on the right.

Fig. 69 Formulario clave_acceso, como usuario.

De manera que para que las opciones de modificación de todos los eventos y dispositivos que el dueño de casa dispuso no puedan ser accedidas por las personas que no posean la clave de acceso que en este caso fue establecida para motivos de diseño como “administrador”. Al ingresar la contraseña solamente se visualizaran los caracteres “*” de modo que no pueda ser vista por quien pueda estar cerca.

Una vez ingresada la contraseña es necesario presionar el botón ingresar que tiene la siguiente codificación en el evento “click” que se produce al presionar el botón:

```

Private Sub Ingresar_Click()
    Dim cadena, ingreso As String
    ingreso = Encriptar(Text1.Text, 1)
    Open "C:\TesisvB>Password.txt" For Input As #1
    Input #1, cadena
    Close #1
    If ingreso = cadena Then
        MsgBox "Bienvenido", vbInformation, "Contraseña Correcta"
        Cambiarcont.Visible = True
        Label2.Visible = False
        Text1.Visible = False
    End If
End Sub

```

```
Admin.Visible = True
Ingresar.Visible = False
Usuario.Visible = True
Cambiarcont.Visible = True
Text1.Text = ""
'habilita la programación y gestión de alarmas
Module1.Habilitar (1)
contusu = 1
End If
If ingreso <> cadena Then
    MsgBox "Ingrese nuevamente la Contraseña", vbExclamation, _
"Contraseña incorrecta"
    Text1.Text = ""
End If
End Sub"
```

Al producirse este evento, el archivo "C:\TesisvB\Password.txt", creado anteriormente, es abierto, leído y descriptado, se compara su contenido con el ingresado en la caja de texto, en caso de ser igual, es presentado un mensaje (MsgBox): "Contraseña Correcta". Además es enviado al evento Habilitar programado en el módulo (Module1), el valor de "1" como dice en la instrucción: "Module1.Habilitar (1)", lo que permitirá realizar la rutina de habilitación de todos los elementos para su visualización y control. Esto permite el programa, tener privilegios de administrador. Adicionalmente se cambia el valor de la variable global "**contusu**" a "1". Una vez ingresada la contraseña de manera correcta la apariencia del formulario cambia ocultándose determinados objetos y visualizándose otros con la propiedad "Visible". El formulario queda de la siguiente forma:

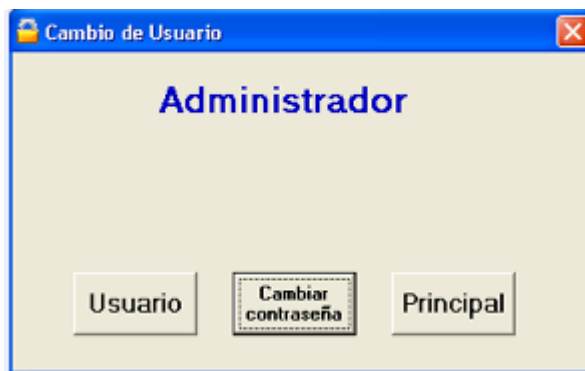


Fig. 70 Formulario clave_acceso, como Administrador.

Entre las opciones que se despliegan esta el botón “Usuario” y el botón Cambiar contraseña. Al seleccionar el botón “Usuario”, se deshabilitan todos los privilegios se administrador, y el formulario vuelve a tener la forma mostrada en la Fig. 64 la codificación de este botón es la siguiente:

```
“Private Sub Usuario_Click()  
    Label2.Visible = True  
    Text1.Visible = True  
    Admin.Visible = False  
    Ingresar.Visible = True  
    Usuario.Visible = False  
    Cambiarcont.Visible = False  
    Text2.Visible = False  
    Text3.Visible = False  
    Label1.Visible = False  
    Label3.Visible = False  
    Aceptar.Visible = False  
    contusu = 2  
    Module1.Habilitar (0)  
End Sub
```


Se puede notar, que entre las instrucciones del código mostrado, se cambia el valor de la variable global “**contusu**” al valor “2”, lo que hace que la apariencia del formulario cambie a la mostrada en la Fig. 64. La instrucción “**Module1.Habilitar (0)**”, tiene por finalidad enviar a la subrutina “**Habilitar**” codificada en el módulo “**Module1**” el valor de “0”, lo que hace que se deshabiliten los privilegios de administrador. La subrutina “**Habilitar**” será explicada de mejor manera en la sección 4.4.2 y 4.4.3.

Por motivos de seguridad el dueño de casa tendrá la opción de cambiar la contraseña para personalizarla de acuerdo a su memoria y su seguridad como él prefiera accediendo al botón “Cambiar Contraseña”, que tiene la siguiente representación en código:

```
“Private Sub Cambiarcont_Click()  
    Cambiarcont.Visible = False  
    Ingresar.Visible = False  
    Label1.Visible = True  
    Label2.Visible = False  
    Label3.Visible = True  
    Text1.Visible = False  
    Text2.Visible = True  
    Aceptar.Visible = True  
    Text3.Visible = True  
End Sub”
```

La interpretación de este código tiene que ver con el cambio de forma del formulario “clave_acceso” a la siguiente:

Fig. 71 Formulario clave_acceso, cambiar contraseña.

Una vez que el formulario tenga la apariencia que se ve en la Fig. 66, es posible cambiar la contraseña ingresada por omisión (administrador), por la que el administrador desee por lo que deberá llenarse las dos cajas de texto con la misma contraseña, una vez que hayan sido llenadas, debe presionarse el botón “Aceptar”, cuyo código se presenta a continuación:

```

“Private Sub Aceptar_Click()
  If Text2.Text = Text3.Text Then
    Open "C:\TesisvB\Password.txt" For Output As #1
    Print #1, Encriptar(Text2.Text, 1)
    Close #1
    Aceptar.Visible = False
    Label2.Visible = False
    Text2.Visible = False
    Text3.Visible = False
    clave_acceso.Show
    Unload Me
    MsgBox "Su Nueva Contraseña ha sido Guardada",_
      vbInformation, "Contraseña Cambiada"
  End If
  If Text2.Text <> Text3.Text Then
    MsgBox "Ingreso Incorrecto", vbExclamation, "Error"
  End If

```

End Sub”

El motivo por el cual se requiere que se ingrese dos veces la contraseña se debe a que pueden existir errores de digitación, una vez guardada la contraseña, esta será cambiada y el administrador podría tener problemas para ingresar de nuevo. Por lo cual, al momento de presionar el botón aceptar, el código escrito anteriormente, representa que en primer lugar se compara la contraseña con la confirmación, en caso de no ser iguales se presenta un mensaje de error y tendrá que ingresarse nuevamente la contraseña. En caso de coincidir se abre el archivo “C:\TesisvB>Password.txt” utilizando la instrucción: **“Open “C:\TesisvB>Password.txt” For Output As #1”**, y a continuación se guarda la nueva contraseña en el archivo mencionado mediante la instrucción: **“Print #1, Encriptar(Text2.Text, 1)”**. La contraseña es guardada utilizando la función encriptar, que tiene por código:

```

“Function Encriptar(UserKey As String, Action As Single) As String
  Dim Temp As Integer
  Dim i As Integer
  Dim j As Integer
  Dim n As Integer
  Dim rtn As String

  n = Len(UserKey)
  ReDim UserKeyASCIIS(n)
  For i = 1 To n
    UserKeyASCIIS(i) = Asc(Mid(UserKey, i, 1))
  Next

  If Action = 1 Then
    For j = 1 To n
      Temp = UserKeyASCIIS(j) + 32
    Next
  End If
End Function

```

```
    If Temp > 255 Then
        Temp = Temp - 255
    End If
    rtn = rtn + Chr(Temp)
Next
ElseIf Action = 2 Then
    For j = 1 To n
        Temp = UserKeyASCIIS(j) - 32
        If Temp < 0 Then
            Temp = Temp + 255
        End If
        rtn = rtn + Chr(Temp)
    Next
End If
    Encriptar = rtn
End Function”
```

El código anteriormente mostrado está escrito en un módulo, que es un conjunto de funciones y procedimientos sin interfaz gráfica de usuario usualmente utilizados para tener código usado para poder usarlo con funciones y eventos públicos y acceder a los mismos de diferentes formularios.

Estas líneas de código antes escritas representan una función que recibe dos valores “**Function Encriptar(UserKey As String, Action As Single)**”. El primero de ellos (**UserKey As String**), es la nueva contraseña ingresada por el usuario, y el segundo (**Action As Single**), la acción que se desea realizar. Cuando el segundo valor pasado es “1” se encripta la cadena de caracteres pasada, separando los caracteres, sumando al código ASCII de cada uno de ellos número treinta y dos, y volviéndolos a unir, para devolver una cadena de caracteres que no sea entendible. De forma análoga, en caso de que el

segundo valor pasado sea “2” se toma la cadena de caracteres (**UserKey As String**), pasada a la función y se realiza un proceso inverso al de encriptación.

Todo esto se lleva a cabo una vez seleccionada la opción aceptar, guardando la nueva contraseña encriptada en un archivo de texto, de tal manera que si alguien pudiera tener acceso a leer el contenido de dicho archivo no podría entender el contenido del mismo, ni acceder a la contraseña.

Una vez guardada la contraseña se presenta un mensaje (MsgBox) que indique que la contraseña ha sido guardada.

En cualquier momento se puede presionar el botón “Principal” que cierra el formulario “Clave_acceso” y va al formulario principal donde se realiza todo el monitoreo y control de los dispositivos domóticos instalados en la vivienda cuya representación en código es:

```
“Private Sub Principal_Click()  
    clave_acceso.Show  
    Unload Me  
End Sub”
```

Instrucción que solamente cierra el formulario. Ha de mencionarse que en caso de que se cambie de usuario sea este; “Usuario” o “Administrador”. Cada vez que se cargue el formulario se revisará la variable “**contusu**” para mostrar la apariencia que sea pertinente.

3.4.3 DISEÑO E IMPLEMENTACIÓN DE LA VENTANA PRINCIPAL DEL SOFTWARE HMI.

Al abrir el software HMI debe presentarse una ventana en la que se pueda monitorear y controlar todos los dispositivos domóticos instalados en la vivienda, la misma que estará en el formulario principal del programa llamado “Plantas”. La apariencia del formulario en el programa de desarrollo **Visual Basic** es la siguiente:

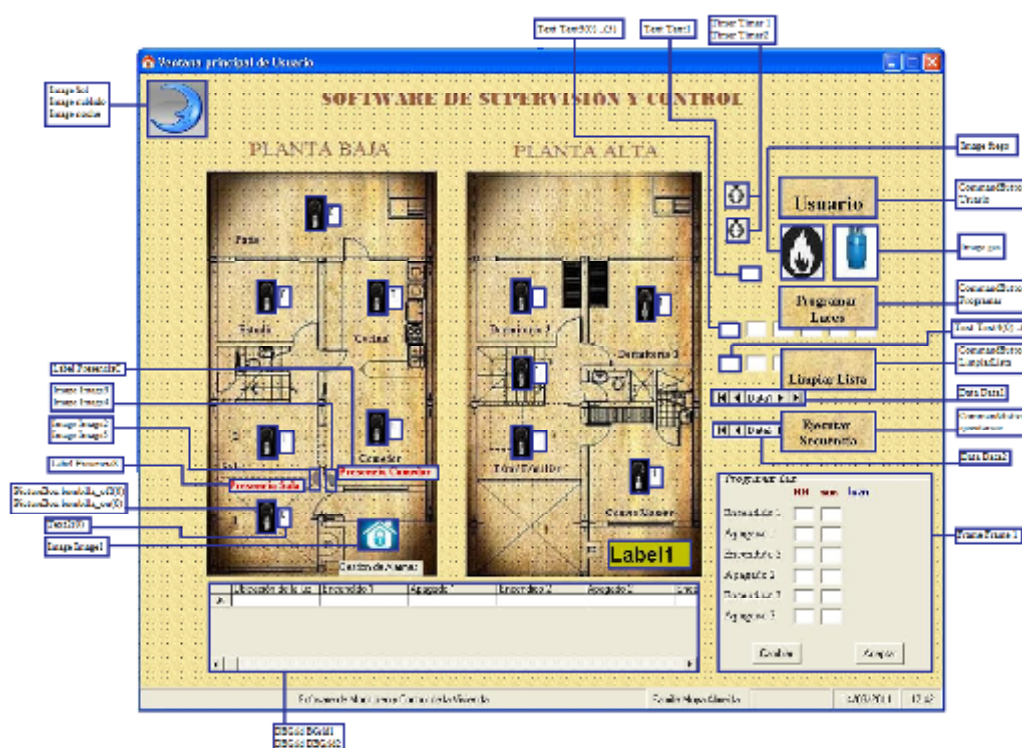


Fig. 72 Formulario “Plantas”.

En el formulario principal están presentes los siguientes objetos:

- Dos arreglos de once cajas de imagen cada uno (PictureBox).
- Un arreglo de once cajas de texto (TextBox)

- Dos arreglos de seis cajas de texto.
- Dos bases de datos.
- Cuatro botones (Command Button).
- Una caja de texto.
- Dos controles “DBGrid”.
- Dos controles “Data”.
- Dos controles “Timer”.
- Tres etiquetas (Label).
- Nueve imágenes (Image).
- Un marco (Frame) que agrupa:
 - + Cuatro arreglos de tres cajas de texto cada uno.
 - + Dos arreglos de tres etiquetas (label) cada uno.
 - + Tres etiquetas.
 - + Dos botones (Command Button)

Los arreglos de controles son empleados en este caso para realizar acciones iguales en diferentes elementos al ocurrir un evento.

Las cajas de imagen, o control **PictureBox**, es una especie de formulario reducido ya que puede contener imágenes u otro tipo de controles.

Las bases de datos es un conjunto de datos pertenecientes a un mismo contexto, almacenados sistemáticamente para su posterior uso. La idea de tener una base de datos es que los datos ingresados permanezcan guardados en el equipo aun cuando el programa en el que se los ingreso sea cerrado, es decir que no solamente serán usados el tiempo de ejecución del programa sino serán almacenados para su posterior uso.

Los controles **DBGrid** son usados para visualizar, acceder y manipular el contenido de una base de datos mediante el enlace con un control “**data**”.

Los controles **Data**, se utiliza como mecanismo de enlace de controles a una base de datos mediante DAO (Data Access Objects).

El control **Timer**, sirve cuando se requiere que una acción suceda con cierta periodicidad. Éste control produce de modo automático un evento cada cierto número de milisegundos.

Las imágenes o control **Image** es un contenedor de gráficos bitmap, íconos, metafile, enhanced metafile, GIF y JPEG. Debido a que tiene una amplia colección de eventos, posee una gran versatilidad, en este caso es utilizado para mostrar alertas y acceder a ciertas funcionalidades del programa.

Un marco o control **frame**, se utiliza para agrupar botones o controles dentro de un marco, que estén relacionados, de esta forma se puede acceder a la propiedad “**Visible**” por ejemplo y solamente utilizarla en el **frame** y no en cada uno de los controles para ocultar todos los controles que contenga un marco.

3.4.3.1 Intercambio dinámico de datos (DDE) en Visual Basic.

El **DDE** es una utilidad de Windows empleada por Visual Basic que permite crear aplicaciones que tomen datos una de la otra. Para que esto suceda se necesitan al menos dos aplicaciones, una que se deje leer, y otra u otras que lean la información presente en la primera. Puede también existir el caso en que una aplicación reciba datos de otra, y los envíe a su vez a una tercera aplicación.

La aplicación que envía información lleva el nombre de **aplicación servidor**, y la que recibe, **aplicación cliente**. Ha de considerarse que la aplicación servidor deberá estar funcionando antes de que la aplicación cliente requiera información, de no ser así ocurrirá un error.

En este caso particular, para el proyecto que se desarrolló, se utilizó el servidor LNS DDE Server, como se mencionó previamente al iniciar la sección 4.4, por lo cual es necesario siempre abrir previamente el LNS Server y el LNS DDE Server, caso contrario se producirá un error en el programa al no poder comunicarse la aplicación cliente con la aplicación servidor.

Los datos que se pasan de una aplicación a otra son textos de un **Label** o en un **TextBox**, o imágenes de un **PictureBox**. Es importante mencionar que el destino de un texto ha de ser necesariamente un **Label** o un **TextBox**, y el destino de un gráfico debe ser un **PictureBox**.

Cuando se requiere que uno de estos controles sea el destino de un intercambio DDE debemos indicarlo en sus propiedades **LinkMode**,

LinkTopic, **LinkItem**, e indicarle el tiempo de espera para un intercambio en la propiedad **LinkTimeout**.

3.4.3.1.1 Propiedad **LinkMode**.

En un control la propiedad **LinkMode** permite iniciar una conversación DDE con una aplicación servidor. Tiene 4 valores posibles:

- **None (“0”).** No existe comunicación DDE con ese control.
- **Automático (“1”)** Los datos se enviarán desde la aplicación servidor a este control de la aplicación cliente cada vez que cambien el dato de la aplicación de origen.
- **Manual (“2”)** Los datos serán pasados cuando lo pida la aplicación de destino, mediante la orden **LinkRequest**.
- **Notify (“3”)** Cuando haya un intercambio de datos de la aplicación origen, se notificará a la aplicación destino que el dato ha cambiado, más el dato nuevo no será enviado. Será generado un evento denominado **LinkNotify** en la aplicación destino donde debe llevarse la información. En este evento es posible escribir el código que se requiera para la aplicación que se esté desarrollando. Para que la información sea contenida, debe ejecutarse la orden **LinkRequest**.

Cuando se esté ejecutando el programa, esta propiedad es de lectura y escritura para un control. Se la puede usar para saber el valor que tiene en un control específico, o para forzar esta propiedad a un valor determinado. Su sintaxis es:

“objeto.LinkMode = [número]”

Siendo **objeto** el nombre del control y “[número]” un valor entero que especifica el tipo de conexión sea “0”, “1”, “2” o “3”, según lo antes mencionado.

3.4.3.1.2 Propiedad LinkTopic

Es una propiedad de lectura y escritura que es utilizada cuando un control destino requiere establecer una comunicación DDE. Para un control destino, **LinkTopic** contiene el nombre de la aplicación y el tema, los mismos que están separados mediante el carácter 124 (“”). La sintaxis es la siguiente:

“Objeto.LinkTopic = [aplicaciónorigen|Tema]”

El nombre de la aplicación origen es el nombre del servidor. El tema es al mismo al que responde, donde se encuentra la información a pasar.

3.4.3.1.3 Propiedad LinkItem

Esta propiedad solamente la posee el destino de la información. En ella deberá expresarse el nombre del control de origen de la información.

La propiedad **LinkItem** es de lectura y escritura, por lo tanto, al utilizarla, es posible saber el nombre del control origen de los datos, y establecerlo en tiempo de ejecución. Su sintaxis es:

“Objeto.LinkItem = [cadena]”

Siendo cadena el nombre del control de origen que tiene los datos.

3.4.3.1.4 Propiedad LinkTimeout

Establece y devuelve la cantidad de tiempo que un control espera la respuesta a un mensaje DDE. Su sintaxis es:

“objeto.LinkTimeout = [número]”

Siendo “número”, una expresión numérica que representa el tiempo que habrá de espera en décimas de segundo. Se la usa de acuerdo a la aplicación, para la que se implementó no fue necesario modificar esta propiedad.

Ha de considerarse que el mayor tiempo de espera posible es 65.355 décimas de segundo (un poco más de una hora y cuarenta y nueve minutos).

3.4.3.1.5 Evento LinkNotify

Se produce cuando el dato definido por el vínculo DDE cambia, siempre y cuando la propiedad **LinkMode** del control destino está establecida en “3”. Su sintaxis es:

“Private Sub objeto_LinkNotify([índice As Integer])

End Sub”

Utilizando este procedimiento, es posible escribir el código necesario para la aplicación que se esté desarrollando, siendo conocido, que este evento se produce cuando cambia la aplicación de origen. El momento que se desee colocar el dato en el control destino, se puede utilizar el método **LinkRequest**, que hace posible obtener el nuevo dato de la fuente.

3.4.3.1.6 Método **LinkRequest**.

Solicita a la aplicación origen de una conversación DDE que actualice el contenido de un control. La sintaxis es:

“Objeto.**LinkRequest**”

El método **LinkRequest**, hace que la aplicación origen envíe los datos actuales al objeto.

En el caso de que la propiedad **LinkMode** del objeto esté definida como “1”, la aplicación origen actualiza el objeto automáticamente y el método **LinkRequest** no es necesario. Cuando es “2”, la aplicación origen actualiza el objeto solo en caso de usarse **LinkRequest**. Si **LinkMode** está definido como “3”, el origen notificará al destino que se han cambiado datos llamando al método **LinkNotify**. El destino tendrá que utilizar el método **LinkRequest** para actualizar los datos.

3.4.3.1.7 Método LinkPoke

Transfiere el contenido de un control a la aplicación de origen de la conversación DDE. La sintaxis que se debe emplear es:

“Objeto.LinkPoke”

Cuando el objeto es un control **Label**, el método **LinkPoke**, el contenido de la propiedad **Caption**, es transmitido al origen. Si se trata de un **TextBox**, **LinkPoke** transfiere el contenido de la propiedad **Text** al origen.

Usualmente, en una conversación DDE, la información se dirige desde el origen hacia el destino. A pesar de aquello, el método **LinkPoke** permite que un objeto destino suministre datos al origen.

Ha de considerarse que no todas las aplicaciones aceptan información de esta manera, en caso de que la aplicación origen no acepte los datos, se produce un error. Para motivos de la implementación realizada en el presente proyecto se ha utilizado repetidas veces el método LinkPoke debido a que al servidor LNS DDE, permite que un objeto de origen suministre datos en las variables de entrada de los dispositivos.

3.4.3.2 Partes principales de la elaboración del formulario.

Una vez dado el panorama general de los objetos y métodos trascendentales, utilizados en el diseño del formulario principal del software HMI que se implementó. Se definirán cada una de las partes principales que contemplan

la elaboración del formulario, y su interacción con los dispositivos respectivos.

3.4.3.2.1 Control de iluminación

Cada bombillo, lámpara o grupo de ellos que se encuentra en un sector de la casa, que esté domotizado, ha sido representado utilizando arreglos de cajas de imagen o **PictureBox**, dos de ellos uno para representar una luz encendida, y otro para representar que se encuentra apagada. Al ser un total de once lugares donde se controla la luz, se tiene dos arreglos de controles **PictureBox**:



Fig. 73 Arreglo de controles PictureBox “bombilla_off”



Fig. 74 Arreglo de controles PictureBox “bombilla_on”

Como se mostró en las figuras anteriores los respectivos arreglos de controles **PictureBox** son `bombilla_off []`, de once elementos (“0”-“10”), y `bombilla_on[]`, también de once elementos (“0” al “10”). Cada elemento del arreglo representa la iluminación de un lugar de

la casa, si está apagada una luz o encendida, en dicho lugar. El lugar al que corresponden los arreglos, respecto a las luces es el siguiente:

Tabla # 16 Detalle de relación de los arreglos `bombilla_on` y `bombilla_off`, respecto al lugar de la luz.

Ubicación de la luz	Elemento <code>bombilla_on</code>	Elemento <code>bombilla_off</code>
Sala (1)	<code>bombilla_on[0]</code>	<code>bombilla_off[0]</code>
Sala (2)	<code>bombilla_on[1]</code>	<code>bombilla_off[1]</code>
Comedor	<code>bombilla_on[2]</code>	<code>bombilla_off[2]</code>
Estudio	<code>bombilla_on[3]</code>	<code>bombilla_off[3]</code>
Cocina	<code>bombilla_on[4]</code>	<code>bombilla_off[4]</code>
Patio	<code>bombilla_on[5]</code>	<code>bombilla_off[5]</code>
Estar Familiar	<code>bombilla_on[6]</code>	<code>bombilla_off[6]</code>
Cuarto Máster	<code>bombilla_on[7]</code>	<code>bombilla_off[7]</code>
Escaleras	<code>bombilla_on[8]</code>	<code>bombilla_off[8]</code>
Dormitorio 3	<code>bombilla_on[9]</code>	<code>bombilla_off[9]</code>
Dormitorio 2	<code>bombilla_on[10]</code>	<code>bombilla_off[10]</code>

De esta manera hay una representación de encendido y apagado de cada una de las luces controladas. Adicionalmente para una mejor comprensión del lugar de ubicación de las luces, de modo que sea fácilmente entendible para el usuario, se puso como fondo del formulario principal una imagen que representa los planos caricaturizados para una apariencia de envejecidos, utilizando la propiedad `Picture` del formulario principal llamado “**Plantas**”. Nótese en la figura que se presenta a continuación la apariencia de los planos de la vivienda con cada una de las representaciones de las luces:

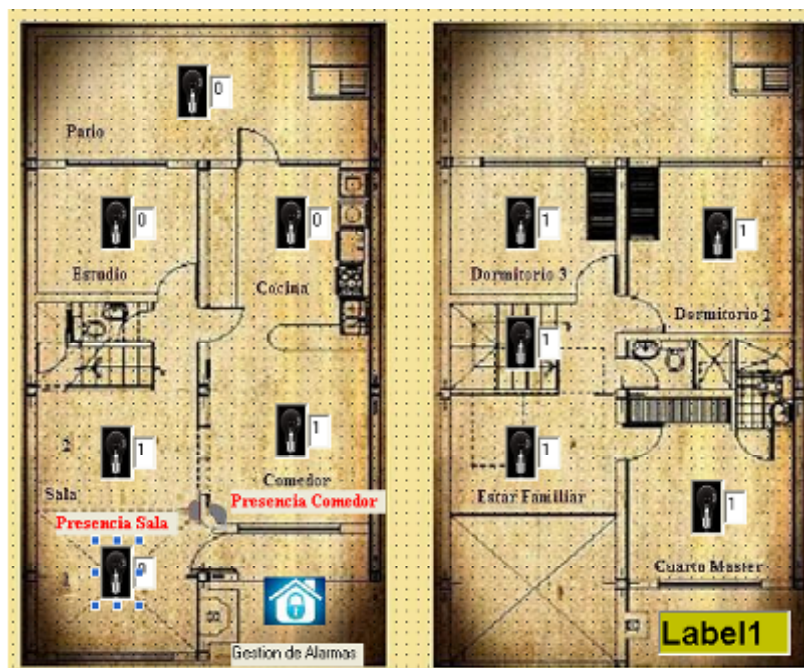


Fig. 75 Planos de la vivienda con representaciones de las luces.

Adicionalmente a la representación gráfica de los lugares de ubicación de las respectivas luces en el plano, para dar una representación intuitiva, se ha escrito el nombre de cada lugar, de modo que sea fácil ubicar el lugar respectivo donde se encuentra la luz controlada. Los elementos del arreglo **bombilla_off** están sobrepuestos a los del arreglo **bombilla_on**, pero los últimos están establecidos su propiedad “Visible” como “false”.

Cuando se enciende una luz en la vivienda, se puede visualizar cual fue la que se encendió mediante el respectivo elemento del arreglo **bombilla_on**. Para este motivo se ha usado una matriz de controles **TextBox**, llamado “Text2[]”, de once elementos (“0” al “10”), cada uno de los cuales responde a una variable de red, que pertenece a un objeto **lonmark**, en cuanto, a la configuración de cada elemento del arreglo de cajas de texto empleado para poder visualizar el estado de las luces, se ha utilizado las propiedades **LinkTopic**, y **LinkItem**. La

propiedad **LinkTopic**, como ya se indicó anteriormente tiene la sintaxis:

“Objeto.LinkTopic = [aplicaciónorigen|Tema]”

Además, debe considerarse que ésta propiedad depende de la aplicación origen en el servidor LNS DDE Server, la sintaxis del **“Topic” (Tema)**, que requiere el servidor LNS DDE Server deberá ser de la forma:

“Tema = Nombre_Red.Ruta_Subistema.Tipo_Tema”

Donde **“Nombre_Red”**, es el nombre, con el que se ha definido a la red en la base de datos LNS. **“Ruta_Subistema”**, corresponde a la ruta completa al subsistema al subsistema que contiene el dispositivo a ser accedido. **“Tipo_Tema”**, especifica el tipo de información requerida. Para este proyecto el tipo de tema utilizado, es **“DevNV”**. Que es una variable de red en un dispositivo, para manejar las entradas, salidas y configuraciones de los nodos domóticos implementados.

Finalmente además de la sintaxis dada para la propiedad **LinkTopic**, que puede ser usada en el código del programa, se puede también establecerla directamente en la ventana de propiedades del control respectivo accediendo a **LinkTopic**, esta propiedad ha sido establecida en todos los controles que forman parte de la comunicación DDE establecida de la siguiente forma:

**“Objeto.LinkTopic=LNSDDE|MoyaAlmeida1.RAIZ.Dev
NV”**

Todos los elementos del arreglo de controles **TextBox**, **Text2**, tienen la misma configuración de la propiedad **LinkTopic**.

La propiedad **LinkItem**, como ya se vio anteriormente debe tener la sintaxis:

“Objeto.LinkItem = [cadena]”

Donde **“cadena”** representa la variable de red **ítem**. La cual identifica un dispositivo y una variable de red o campo de una variable de red en un dispositivo. Su sintaxis es la siguiente:

“Nombre_Dispositivo.Nombre_variableRed.Nombre_Campo”

De modo que para el primer elemento del arreglo de controles **Text2**, (**Text2[0]**), la propiedad **LinkItem** tendrá una sintaxis de la forma:

“LinkItem=Iluminacion03.nvoEstCir3.state”

Que significa que el ítem al que se hace referencia se encuentra en el dispositivo **“Iluminacion03”**, la variable de red a la que se quiere acceder es **“nvoEstCir3”**, y el campo que se requiere monitorear es el campo **“state”**.

Es importante mencionar, que la variable de red **nvoEstCir3**, es una variable de salida, por lo tanto no puede ser cambiada sino solamente

monitoreada, se la ha utilizado para poder verificar si una luz se encuentra encendida o apagada, en caso de estar apagada, el campo **state**, permanecerá en “0”, al ser encendida, este campo cambiará su valor a “1”, siendo estos valores, los únicos posibles para ésta variable.

A continuación se puede observar cada uno de los elementos del arreglo **Text2**, con el respectivo valor de la propiedad **LinkItem** que se ha asignado a cada uno de acuerdo al dispositivo, variable de red, y campo que monitorean:

Tabla # 17 Detalle de relación de los arreglos Text2, respecto al lugar de la luz y su propiedad LinkItem.

Ubicación de la luz	Elemento de arreglo Text2	LinkItem
Sala (1)	Text2[0]	Iluminacion03.nvoEstCir3 .state
Sala (2)	Text2[1]	Iluminacion03.nvoEstCir1 .state
Comedor	Text2[2]	Iluminacion03.nvoEstCir2 .state
Estudio	Text2[3]	Iluminacion01.nvoEstCir1 .state
Cocina	Text2[4]	Iluminacion03.nvoEstCir4 .state
Patio	Text2[5]	Iluminacion01.nvoEstCir2 .state
Estar Familiar	Text2[6]	Iluminacion02.nvoEstCir1

		.state
Cuarto Máster	Text2[7]	Iluminacion02.nvoEstCir2 .state
Escaleras	Text2[8]	Iluminacion04.nvoEstCir3 .state
Dormitorio 3	Text2[9]	Iluminacion04.nvoEstCir1 .state
Dormitorio 2	Text2[10]	Iluminacion04.nvoEstCir2 .state

Una vez establecidos, estos campos (LinkTopic y LinkItem), de todos los elementos del arreglo **Text2**, se han ubicado los elementos del arreglo al lado derecho de su homólogo en el arreglo **bombilla_off**, como se ve en la **Fig.70**, pero la propiedad “**Visible**”, de cada uno de los elementos del arreglo **Text2**, ha sido establecida en “**false**”, de modo que al ejecutar el programa, no se pueda ver ninguno de ellos, y sea fácil identificar el lugar donde está la luz que controla para el programador.

Cuando inicia el programa en el evento de carga del formulario (**form_Load**), se establece la propiedad **LinkMode**, de todos los elementos del arreglo **Text2**, en “1” (automático), de modo que pueda presentar automáticamente los cambios del objeto de origen. El código en Visual Basic es:

```

Private Sub Form_Load()
    verificarhora
    Habilitar (0)
    For Index = 0 To 10

```

```
Text2(Index).LinkMode = 1
If Text2(Index).Text = "1" Then
    verluzon (Index)
End If
If Text2(Index).Text = "0" Then
    verluzoff (Index)
End If
Next
Load cuadrocontrol
End Sub”
```

Adicionalmente a lo antes mencionado, se verifica la hora para saber el estado del día (sol, nublado, noche), Se habilita el nivel de ingreso “Usuario”, sin privilegios de administrador. Adicionalmente carga el formulario cuadro de control. Cada uno de estas acciones serán explicadas posteriormente.

El momento que una luz, es encendida físicamente en la vivienda, se produce un cambio en el contenido del respectivo elemento del arreglo **Text2**, en la propiedad “Text”, por lo cual, se produce un evento “**Text2_change**”, en el que es posible indicar la acción respectiva, el evento está programado con el siguiente código:

```
“Private Sub Text2_Change(Index As Integer)
    If Text2(Index).Text = "1" Then
        verluzon (Index)
    End If
    If Text2(Index).Text = "0" Then
        verluzoff (Index)
    End If
End Sub”
```

Las líneas de código mostradas anteriormente representan que cada vez que se ejecute, este evento se tomará en cuenta si el cambio en el elemento del arreglo es de encendido (“1”) o apagado (“0”).

En caso de ser un valor de “1” el que quede en la propiedad “Text” del elemento del arreglo el programa va a ejecutar un sub proceso llevando el número de elemento del arreglo donde se produjo el cambio, el subproceso que al que va es:

```
“Private Sub verluzon(cont As Integer)
    bombilla_off(cont).Visible = False
    bombilla_on(cont).Visible = True
End Sub”
```

Cuya función es únicamente mostrar visualmente la luz que se ha encendido, esto se logra únicamente cambiando la propiedad “Visible” el elemento homólogo del arreglo **bombilla_off**, para que se oculte y se pueda ver el elemento respectivo del arreglo **bombilla_on**, de este modo se logra dar una apariencia de encendido.

Cuando el cambio del elemento del arreglo **Text2**, provoca que la propiedad “Text” del elemento respectivo tenga un valor “0”, se accede al subproceso:

```
“Private Sub verluzoff(cont As Integer)
    bombilla_on(cont).Visible = False
    bombilla_off(cont).Visible = True
End Sub”
```

Lo que en código significa sobreponer el elemento homólogo de **bombilla_off** al de **bombilla_on** respectivo, de modo que se pueda visualizar la imagen de una luz apagada.

Una vez que se ha ingresado en el programa y se ha explicado el funcionamiento de las luces para su visualización en caso de activación física, es menester mencionar, que mientras el operador no ingrese como “Administrador” no tendrá los privilegios de administrador, esto implica que solamente puede monitorear lo que ocurre en la red, es decir que tiene inhabilitadas todas las otras funciones de control de los dispositivos.

Como parte de la supervisión visual y auditiva que ofrece el HMI implementado, se encuentra, la visualización de la claridad del día, la hora actual, aviso de intrusión, aviso de humo y aviso de fuga de gas.

- **Estado de claridad del día:**

Se podrán visualizar tres estados de claridad del día mediante una imagen en la esquina superior izquierda del formulario principal, las posibilidades son; día claro, día nublado, noche. Cada uno de los cuales está representado por la imagen correspondiente:



Fig. 76 Representación día claro.



Fig. 77 Representación día nublado.



Fig. 78 Representación noche.

- **Hora del día:**

En la parte inferior derecha de la representación del plano de la casa, se encuentra un objeto **Label**, que está asociado al controlador **Timer1**, que tiene el siguiente código en su evento **Timer**:

```
“Private Sub Timer1_Timer()  
    Label1.Caption = Time  
End Sub”
```



Fig. 79 Hora del día.

3.4.3.2.2 Sistema de seguridad

- **Aviso de intrusión:**

En caso de que la alarma de intrusión haya sido activada, cuando uno de los sensores de presencia de sala y/o comedor, sea activado, se podrá visualizar en el programa que ha ocurrido una intrusión y el lugar donde ha sido:



Fig. 80 Intrusión Comedor.



Fig. 81 Intrusión Sala y Comedor.

Se anunciará de acuerdo a las imágenes anteriormente mostradas, dependiendo del sensor que lo haya detectado, y adicionalmente el ordenador emitirá un sonido de alarma.

- **Aviso de humo:**

En caso de estar activada la alarma de humo, al detectarse este elemento en el interior de la casa, se emitirá un sonido de alarma técnica y adicionalmente, en el formulario principal se dará un aviso gráfico de que se ha producido una detección de humo en el interior de la vivienda.



Fig. 82 Detección de humo.

- **Aviso de gas:**

En caso de estar activada la alarma de gas, al detectarse este elemento en el interior de la casa, se emitirá un sonido de alarma técnica y adicionalmente, en el formulario principal se dará un aviso gráfico de que se ha producido una detección de gas en el interior de la vivienda.



Fig. 83 Detección de gas.

3.4.3.2.3 Niveles de acceso en función del tipo de usuario

Una vez que el usuario presione el botón “Usuario”, se podrá acceder al formulario **clave_acceso**, en caso de ingresar correctamente como “Administrador”, será habilitado el control de todos los elementos disponibles en el HMI, la ventana mostrada una vez ingresada correctamente la contraseña y cerrado el formulario **clave_acceso** es:



Fig. 84 Usuario “Administrador”.

Entre los controles a los que se puede acceder es el encendido y apagado de luces vía software. El programa ha sido diseñado de tal forma que al hacer un click, sobre las representaciones de las luces, se encienda o se apague la luz respectiva, el código que se utilizó fue:

En caso de que la luz esté apagada:

```
“Private Sub bombilla_off_Click(Index As Integer)
    encenderluz (Index)
    verluzon (Index)
End Sub”
```

En el evento **click** del elemento del arreglo **bombilla_off** respectivo, se produce una llamada a la subrutina **encenderluz**, el valor que se lleva (**Index**) representa el número del elemento del arreglo para identificar que luz es la que se desea encender. El código de la subrutina **encenderluz** es:

```
“Private Sub encenderluz(cont As Integer)
    Text1.LinkTopic = "LNSDDE|MoyaAlmeida1.RAIZ.DevNV"
    If cont = 0 Then
        Text1.LinkItem = "Iluminacion03.nviEstLuz3.state"
        Text1.LinkMode = 3
        Text1.Text = "1"
        Text1.LinkPoke
    End If
    If cont = 1 Then
        Text1.LinkItem = "Iluminacion03.nviEstLuz1.state"
        Text1.LinkMode = 3
        Text1.Text = "1"
        Text1.LinkPoke
    End If
    If cont = 2 Then
        Text1.LinkItem = "Iluminacion03.nviEstLuz2.state"
        Text1.LinkMode = 3
        Text1.Text = "1"
        Text1.LinkPoke
    End If
End Sub”
```

```
End If
If cont = 3 Then
    Text1.LinkItem = "Iluminacion01.nviEstLuz1.state"
    Text1.LinkMode = 3
    Text1.Text = "1"
    Text1.LinkPoke
End If
If cont = 4 Then
    Text1.LinkItem = "Iluminacion03.nviEstLuz4.state"
    Text1.LinkMode = 3
    Text1.Text = "1"
    Text1.LinkPoke
End If
If cont = 5 Then
    Text1.LinkItem = "Iluminacion01.nviEstLuz2.state"
    Text1.LinkMode = 3
    Text1.Text = "1"
    Text1.LinkPoke
End If
If cont = 6 Then
    Text1.LinkItem = "Iluminacion02.nviEstLuz1.state"
    Text1.LinkMode = 3
    Text1.Text = "1"
    Text1.LinkPoke
End If
If cont = 7 Then
    Text1.LinkItem = "Iluminacion02.nviEstLuz2.state"
    Text1.LinkMode = 3
    Text1.Text = "1"
    Text1.LinkPoke
End If
If cont = 8 Then
    Text1.LinkItem = "Iluminacion04.nviEstLuz3.state"
    Text1.LinkMode = 3
```

```
Text1.Text = "1"
Text1.LinkPoke
End If
If cont = 9 Then
Text1.LinkItem = "Iluminacion04.nviEstLuz1.state"
Text1.LinkMode = 3
Text1.Text = "1"
Text1.LinkPoke
End If
If cont = 10 Then
Text1.LinkItem = "Iluminacion04.nviEstLuz2.state"
Text1.LinkMode = 3
Text1.Text = "1"
Text1.LinkPoke
End If
End Sub
'Ubica la lampara que se desea apagar y la apaga
Private Sub apagarluz(cont As Integer)
Text1.LinkTopic = "LNSDDE|MoyaAlmeida1.RAIZ.DevNV"
If cont = 0 Then
Text1.LinkItem = "Iluminacion03.nviEstLuz3.state"
Text1.LinkMode = 3
Text1.Text = "0"
Text1.LinkPoke
End If
If cont = 1 Then
Text1.LinkItem = "Iluminacion03.nviEstLuz1.state"
Text1.LinkMode = 3
Text1.Text = "0"
Text1.LinkPoke
End If
If cont = 2 Then
Text1.LinkItem = "Iluminacion03.nviEstLuz2.state"
Text1.LinkMode = 3
```

```
Text1.Text = "0"  
Text1.LinkPoke  
End If  
If cont = 3 Then  
Text1.LinkItem = "Iluminacion01.nviEstLuz1.state"  
Text1.LinkMode = 3  
Text1.Text = "0"  
Text1.LinkPoke  
End If  
If cont = 4 Then  
Text1.LinkItem = "Iluminacion03.nviEstLuz4.state"  
Text1.LinkMode = 3  
Text1.Text = "0"  
Text1.LinkPoke  
End If  
If cont = 5 Then  
Text1.LinkItem = "Iluminacion01.nviEstLuz2.state"  
Text1.LinkMode = 3  
Text1.Text = "0"  
Text1.LinkPoke  
End If  
If cont = 6 Then  
Text1.LinkItem = "Iluminacion02.nviEstLuz1.state"  
Text1.LinkMode = 3  
Text1.Text = "0"  
Text1.LinkPoke  
End If  
If cont = 7 Then  
Text1.LinkItem = "Iluminacion02.nviEstLuz2.state"  
Text1.LinkMode = 3  
Text1.Text = "0"  
Text1.LinkPoke  
End If  
If cont = 8 Then
```



```
Text1.LinkItem = "Iluminacion04.nviEstLuz3.state"  
Text1.LinkMode = 3  
Text1.Text = "0"  
Text1.LinkPoke  
End If  
If cont = 9 Then  
Text1.LinkItem = "Iluminacion04.nviEstLuz1.state"  
Text1.LinkMode = 3  
Text1.Text = "0"  
Text1.LinkPoke  
End If  
If cont = 10 Then  
Text1.LinkItem = "Iluminacion04.nviEstLuz2.state"  
Text1.LinkMode = 3  
Text1.Text = "0"  
Text1.LinkPoke  
End If  
End Sub”
```

En este código mostrado anteriormente se utiliza un control **TextBox**, llamado **Text1**, para lograr encender la luz que se requiera para lo que mediante programa se definen sus propiedades de acuerdo a lo anteriormente indicado, tal es así que se establece la propiedad **LinkTopic** de dicho control como la de los elementos del arreglo **Text2** ("LNSDDE|MoyaAlmeida1.RAIZ.DevNV"), la sentencia **If**, es utilizada para verificar que elemento específicamente es, en caso de identificarlo, la propiedad **LinkItem** es cambiada de acuerdo al dispositivo que se necesita acceder, a la variable que se requiere, en este caso son variables de entrada (**nvi**) lo que permite modificarlas, y se accede al campo **state** de cada una de ellas. La propiedad **LinkMode** es puesta en 3, es decir que el valor será cambiado cuando el programa destino (VB) lo requiera, finalmente el valor ingresado es un "1" que representa un estado de encendido,

y se utiliza el método **LinkPoke** para que el dato sea enviado. Las variables de red **nviEstLuz**, representan variables de entrada del estado de la luz.

En caso de que la luz se encuentre encendida, el código escrito del programa es:

```
“Private Sub bombilla_on_Click(Index As Integer)
    apagarluz (Index)
    verluzoff (Index)
End Sub”
```

En el evento **click** del elemento del arreglo **bombilla_on** respectivo, se produce una llamada a la subrutina **apagarluz**, el valor que se lleva (**Index**) representa el número del elemento del arreglo para identificar que luz es la que se desea encender. El código de la subrutina **apagarluz** es:

```
“Private Sub apagarluz(cont As Integer)
    Text1.LinkTopic = "LNSDDE|MoyaAlmeida1.RAIZ.DevNV"
    If cont = 0 Then
        Text1.LinkItem = "Iluminacion03.nviEstLuz3.state"
        Text1.LinkMode = 3
        Text1.Text = "0"
        Text1.LinkPoke
    End If
    If cont = 1 Then
        Text1.LinkItem = "Iluminacion03.nviEstLuz1.state"
        Text1.LinkMode = 3
        Text1.Text = "0"
        Text1.LinkPoke
    End If
```

```
If cont = 2 Then
  Text1.LinkItem = "Iluminacion03.nviEstLuz2.state"
  Text1.LinkMode = 3
  Text1.Text = "0"
  Text1.LinkPoke
End If
If cont = 3 Then
  Text1.LinkItem = "Iluminacion01.nviEstLuz1.state"
  Text1.LinkMode = 3
  Text1.Text = "0"
  Text1.LinkPoke
End If
If cont = 4 Then
  Text1.LinkItem = "Iluminacion03.nviEstLuz4.state"
  Text1.LinkMode = 3
  Text1.Text = "0"
  Text1.LinkPoke
End If
If cont = 5 Then
  Text1.LinkItem = "Iluminacion01.nviEstLuz2.state"
  Text1.LinkMode = 3
  Text1.Text = "0"
  Text1.LinkPoke
End If
If cont = 6 Then
  Text1.LinkItem = "Iluminacion02.nviEstLuz1.state"
  Text1.LinkMode = 3
  Text1.Text = "0"
  Text1.LinkPoke
End If
If cont = 7 Then
  Text1.LinkItem = "Iluminacion02.nviEstLuz2.state"
  Text1.LinkMode = 3
  Text1.Text = "0"
```

```
Text1.LinkPoke
End If
If cont = 8 Then
Text1.LinkItem = "Iluminacion04.nviEstLuz3.state"
Text1.LinkMode = 3
Text1.Text = "0"
Text1.LinkPoke
End If
If cont = 9 Then
Text1.LinkItem = "Iluminacion04.nviEstLuz1.state"
Text1.LinkMode = 3
Text1.Text = "0"
Text1.LinkPoke
End If
If cont = 10 Then
Text1.LinkItem = "Iluminacion04.nviEstLuz2.state"
Text1.LinkMode = 3
Text1.Text = "0"
Text1.LinkPoke
End If
End Sub”
```

La explicación de estas líneas de código es similar a la dada para el encendido, pero con la diferencia de que el valor que es enviado al campo **state**, de la variable respectiva es “0” para que pueda ser apagada la luz que se requiera.

Para la visualización del encendido y apagado, se utiliza la misma función anteriormente explicada, que se usa para ver que luz ha sido encendida manualmente.

3.4.3.3 Programación de luces

Cuando se haya accedido correctamente como administrador al programa, es posible realizar una secuencia determinada por el usuario que permite el encendido y apagado de hasta tres veces cada luz, de manera automática. Se ingresa mediante el botón “Programar Luces”, al hacerlo la apariencia del formulario cambiará a la siguiente:



Fig. 85 Programar luces.

Los botones que aparecen como visibles son; “Salir de programación”, “Limpiar Lista” y “Ejecutar Secuencia”. Para dar un panorama más claro de la manera en la que funciona la programación de las luces, es necesario explicar primero algunos recursos utilizados.

3.4.3.3.1 Base de datos del control de programación de luces

Las bases de datos, se utilizan para guardar información de forma organizada. Información que puede ser reutilizada. Usualmente todo lo que se realiza en Visual Basic con los controles normales tiene funcionamiento y permanece guardado durante el tiempo de ejecución del programa, pero una vez cerrado dicho programa, los datos se pierden, cuando se requiere que los datos permanezcan en el equipo, una vez que el programa se cierre, se requiere almacenarlos en el sistema, para esto se utiliza herramientas que permitan acceder a una base de datos, modificarla, guardarla y utilizarla.

Uno de los programas utilizados para crear y manejar base de datos es el programa **Microsoft Access**, pero se requiere que la base de datos se comunique con el HMI para lo cual, existe una herramienta dentro del programa Visual Basic, que permite administrar bases de datos con todo lo que eso implica, es decir, crearlas, llenarlas y modificarlas. Esta herramienta es llamada “**Administrador Visual de Datos**”. Se accede a ella seleccionando el menú “**Complementos**”-“**Administrador Visual de Datos**”.

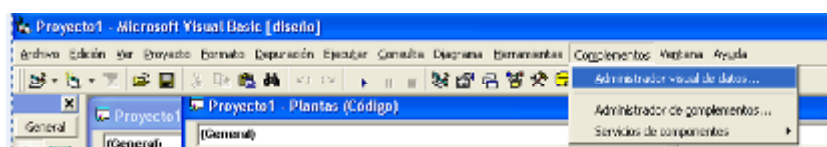


Fig. 86 Acceso al administrador visual de datos.

Una vez que se ha accedido a este complemento, al momento de crear se seleccionó la ruta “**Nuevo - Microsoft Access - MBD Versión 7.0**”.

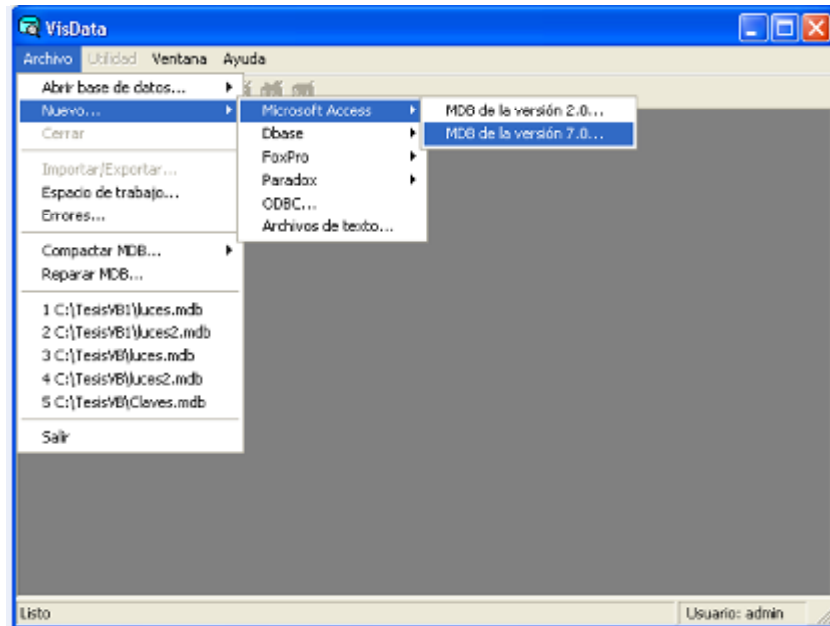


Fig. 87 VisData.

De esta forma se puede crear una base de datos, en Access utilizando esta herramienta de Visual Basic. Las bases de datos que fueron creadas para este proyecto fueron luces y luces2, llenadas con los mismos campos, el lugar donde cada una de ellas está guardada es; “C:\TesisVB1\luces.mdb” y “C:\TesisVB1\luces2.mdb”, respectivamente con siete campos, una vez llenados son:

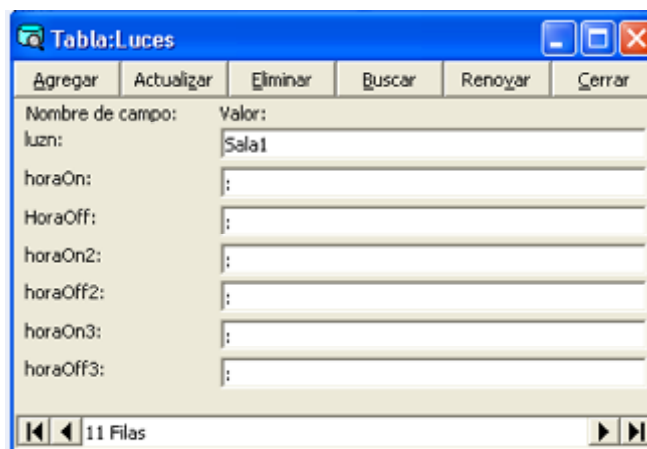


Fig. 88 Campos de bases de datos.

Cada campo tiene once elementos, el primer campo posee los lugares donde se encuentra determinada luz, los cuales están en el mismo orden que los elementos de la primera fila de la **Tabla#17**, los demás campos contienen valores de horas. Una vez creadas las bases de datos, deben ser accedidas por el programa, para visualizar todos los elementos de una base de datos en una tabla, se utilizó el control “**DBGrid**”. Este control no está disponible, así que debe agregarse seleccionando el menú “**Proyecto - Componentes**”, una vez que se despliegue la ventana “**Componentes**”, es necesario seleccionar el control “**Microsoft Data Bound Grid Control (SP5)**”.

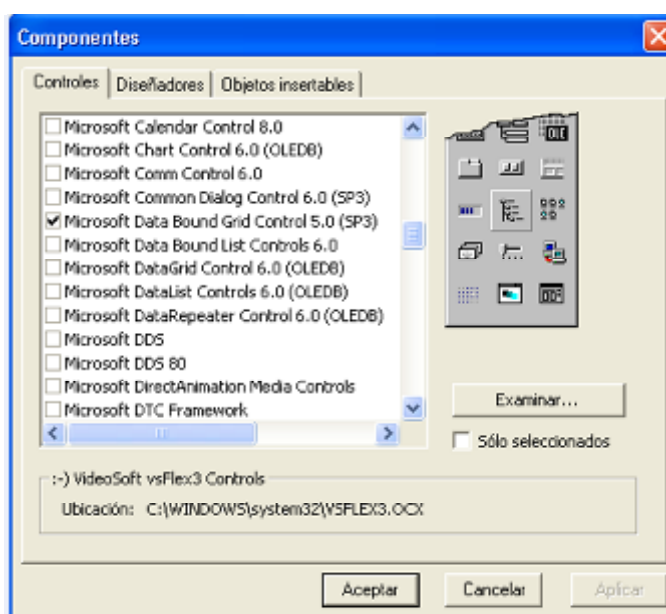


Fig. 89 Selección de control DBGrid.

Una vez seleccionado el control aparecerá un ícono en el cuadro de herramientas.



Fig. 90 Ícono DBGrid

Una vez con este control en el cuadro de herramientas se puede poner un control **DBGrid** en el formulario principal, se utilizó dos controles **DBGrid**, los mismos que se encuentran en la parte inferior del formulario debajo de los planos de la casa.

	Ubicación de la luz	Encendido 1	Apagado 1	Encendido 2	Apagado 2	Encendido 3
*						

Fig. 91 DBGrid.

Para asociar un control **DBGrid** a una Base de datos, es necesario utilizar el control **Data**, utilizando la propiedad **DataSource**.

Para el proyecto realizado se emplearon dos controles **DBGrid** (**DBGrid1** y **DBGrid2**), y dos controles **Data**, (**Data1** y **Data2**), asociados en el mismo orden.

Los controles **Data**, permiten acceder a distintos tipos de bases de datos, las propiedades que se utilizan para enlazarlos a las bases de datos son:

Connect. Indica el tipo de base de datos con la que se va a trabajar, en este caso se usó Access.

DataBaseName. Especifica el lugar y el nombre de la base de datos a la que se quiere acceder, en este caso para **Data1** y **Data2** son respectivamente “C:\TesisVB1\luces.mdb” y “C:\TesisVB1\luces2.mdb”.

Todos los elementos de los controles **DBGrid** han sido bloqueados para la escritura directa, de modo que para modificar las horas de encendido y apagado que están guardadas en la base de datos, se utilizó controles aparte. Que están en un control **frame** (marco), ubicado al lado derecho de los **DBGrid1** y **2**.

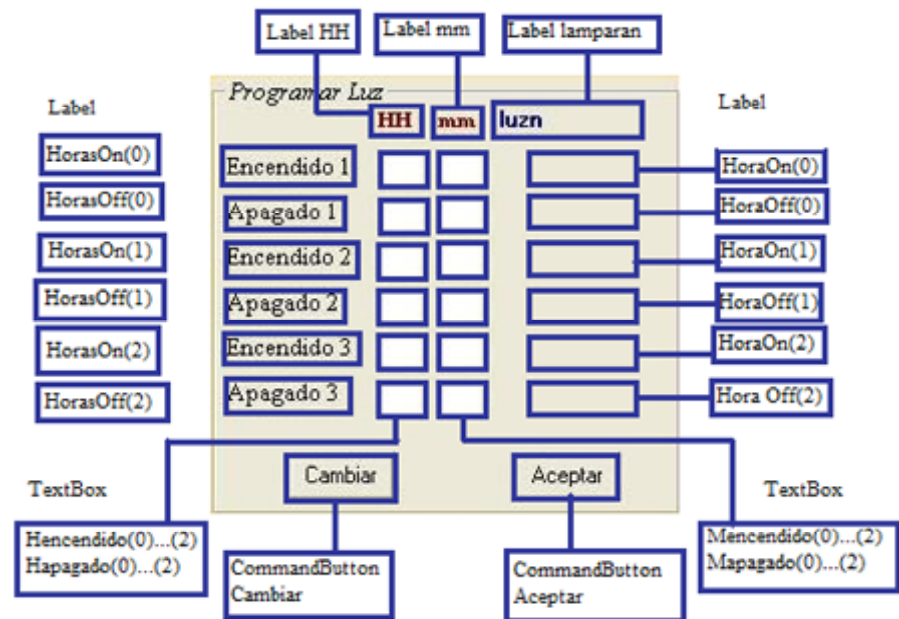


Fig. 92 Framel.

Algunos de los controles presentes en el **Frame1** (marco) de la figura anterior está asociados a elementos de las bases de datos mediante los controles **Data** y **DBGrid**, de modo que, cuando se señale un elemento de la tabla del control **DBGrid**, en el control **Frame** se pueda ver los valores que existen en el elemento específico de la base de datos al que esté relacionado cada control, tal es así que el **label** de nombre **lamparan**, representado en la (Fig. 87) con letras azules como **luzn**, tiene relación con el primer campo de la base de datos “C:\TesisVB1\luces.mdb”. La configuración de las propiedades de dicho **label (lamparan)**, para que sea enlazado a un campo de la base de datos, es necesario utilizar las propiedades:

- **DataSource.** Establece un valor que permite especificar el control **Data**, mediante el cual se enlaza un control, a una base de datos.
- **DataField.** Devuelve o establece un valor que enlaza a un control, con un campo del registro al que se refiera.

Para todos los **Label** presentes en este marco (**Frame1**), la propiedad **DataSource** es **Data1**, y la propiedad **DataField**, tiene diferentes campos, de la forma:

Tabla # 18 Detalle de propiedad DataField de los controles Label del Frame1.

Control Label	DataField
Lamparan	Luzn
HoraOn(0)	horaOn
HoraOff(0)	horaOff
HoraOn(1)	horaOn2
HoraOff(1)	horaOff2
HoraOn(2)	horaOn3
HoraOff(2)	horaOff3

De modo que cuando se seleccione un elemento del control **DBGrid**, cada **Label** muestre el contenido que tiene el campo respectivo de la fila donde se encuentra el elemento que se seleccionó.



Fig. 93 Selección de un elemento del DBGrid.

En la figura anterior se ha seleccionado el primer elemento del **DBGrid**, y se puede ver que los **Label**, muestran el elemento del campo al que están enlazados, por ejemplo el campo “**luzn**”, muestra su primer elemento, en este caso “**Sala1**” y así sucesivamente. Esto permite que en los **Label**, donde es visualizado el elemento respectivo del campo del registro al que hace referencia, se pueda hacer cambios que automáticamente son ingresados en la base de datos, y visualizados en el **DBGrid**.

Cuando se requiera cambiar las horas a las que está programado que se encienda o apague una determinada luz, es necesario presionar el botón “Cambiar” que se encuentra en el interior del **Frame**, en la parte inferior izquierda dicho **Frame**. Cuando se realice esta acción. Aparecerán a la vista cuatro arreglos de tres controles **TextBox** cada uno, que permitirán ingresar la hora del día a la que se encenderá y apagará esa luz.

	HH	mm	Salida
Encendido 1	<input type="text"/>	<input type="text"/>	0:00
Apagado 1	<input type="text"/>	<input type="text"/>	1:00
Encendido 2	<input type="text"/>	<input type="text"/>	2:00
Apagado 2	<input type="text"/>	<input type="text"/>	3:00
Encendido 3	<input type="text"/>	<input type="text"/>	4:00
Apagado 3	<input type="text"/>	<input type="text"/>	5:00

Fig. 94 Cambiar hora de encendido de luces.

El formato de ingreso de la hora debe ser de “0:00” a “23:59”, la lógica que se sigue, es que la caja de texto de la izquierda es para el ingreso de la hora, y la de la derecha, para los minutos. El código escrito para este efecto es:

En el control **Hencendido**, (Hora de encendido) en el evento **Change**:

```

“Private Sub Hencendido_Change(Index As Integer)
    Call Validarhora(Hencendido(Index).Text, 1, Index)
End Sub”

```

Se llama a la subrutina **Validarhora** y se lleva el texto ingresado, el número uno, y el número que identifica el elemento de la lista. La función validar hora se encarga de verificar que se ingrese correctamente la hora, de acuerdo al formato que es reconocido por el programa Visual Basic para utilizar posteriormente la hora ingresada.

```

“Private Sub Validarhora(hora As String, op As Integer, Indice_
As Integer)

```

```

    num = Val(hora)

```

```
If hora <> num And hora <> "" Then
    Call Herronea(op, Indice)
End If
If num > 23 Or num < 0 Then
    Call Herronea(op, Indice)
End If
End Sub”
```

Si la hora no es ingresada de acuerdo al formato, el programa ejecuta una subrutina que da un mensaje de alerta, de que se ha ingresado correctamente y de acuerdo al índice pasado entre la información, borra el contenido ingresado erróneamente, de modo que deba ser llenado de nuevo, el segmento de código respectivo es:

```
“Private Sub Herronea(opcion As Integer, Indice As Integer)
    respuesta = MsgBox("Ingrese una hora valida", 0, "Hora_
inexistente")
    If opcion = 1 Then
        Hencendido(Indice) = ""
    End If
    If opcion = 2 Then
        Mencendido(Indice) = ""
    End If
    If opcion = 3 Then
        Hapagado(Indice) = ""
    End If
    If opcion = 4 Then
        Mapagado(Indice) = ""
    End If
End Sub“
```

Las opciones son de acuerdo al número que ha sido pasado dependiendo del **TextBox** que se esté validando.

En el caso del arreglo **Hapagado** (hora apagado) el segmento de código es:

```
“Private Sub Hapagado_Change(Index As Integer)
    Call Validarhora(Hapagado(Index).Text, 3, Index)
End Sub”
```

El arreglo **Mencendido** (minuto encendido) corresponde al segmento de código:

```
“Private Sub Mencendido_Change(Index As Integer)
    Dim cadena As String
    num = Val(Mencendido(Index))
    If Mencendido(Index).Text <> num And
Mencendido(Index).Text_ <> "" Then
        If 0 <= num <= 9 Then
            cadena = num
            cadena = "0" + cadena
            If Mencendido(Index).Text <> cadena Then
                Call Herronea(2, Index)
            End If
        End If
    End If
    If num > 59 Or num < 0 Then
        Call Herronea(2, Index)
    End If
End Sub”
```

Se valida el ingreso de los minutos de acuerdo al formato requerido (“00” a “59”) en caso de no estar escritos de la forma adecuada, se llama a la subrutina **Herronea**, con el número “2” que indica que casilla será vaciada para el ingreso correcto del formato de minutos.

El arreglo **Mapagado** (minuto apagado) tiene el segmento de código:

```
“Private Sub Mapagado_Change(Index As Integer)
  Dim cadena As String
  num = Val(Mapagado(Index))
  If Mapagado(Index).Text <> num And Mapagado(Index).Text_
<> "" Then
    If 0 <= num <= 9 Then
      cadena = num
      cadena = "0" + cadena
      If Mapagado(Index).Text <> cadena Then
        Call Herronea(4, Index)
      End If
    End If
  End If
  If num > 59 Or num < 0 Then
    Call Herronea(4, Index)
  End If
End Sub”
```

Que tiene una función similar a la de la subrutina descrita para **Mencendido**, con la diferencia que el valor pasado para la función **Herronea** es “4”.

Una vez que todos los valores de la hora respectiva a la que se desee que se encienda y apague una luz respectivamente (hasta tres veces), se confirma presionando el botón “Aceptar”, que hace que los valores ingresados, entren a la base de datos, y se los pueda visualizar en los controles **Label** contiguos respectivos, y posteriormente en el **DBGrid**, adicionalmente los arreglos de controles **TextBox**, son ocultados para que sea visible que los nuevos valores se han ingresado, de este modo la apariencia del control **Frame**.

En cualquier momento de la programación de las luces se puede presionar el botón “Limpiar Lista”, cuya función es vaciar completamente la Lista llenando en su lugar el carácter “:”, cuando se presiona este botón se produce el evento **click** respectivo, que tiene el segmento de código:

```
“Private Sub LimpiarLista_Click()  
    Data1.Recordset.MoveFirst  
    For i = 0 To 5  
        While Not (Data1.Recordset.EOF = True)  
            Text3(i).Text = ":"  
            Data1.Recordset.Edit  
            Data1.Recordset.Update  
            Data1.Recordset.MoveNext  
        Wend  
        Data1.Recordset.MoveFirst  
    Next  
    Data1.Recordset.MoveFirst  
End Sub”
```

Para realizar esta rutina de vaciado de datos se utiliza el objeto **Recordset**, que contiene todos los datos que se pueden leer de una

base de datos, o enviar a ella. Un **Recordset** puede tener varias filas y columnas de datos. Cada fila es un registro, y cada columna es un campo del registro. Sólo se podrá acceder a una fila, la denominada fila actual. Por este motivo en el código mostrado anteriormente se ubica el objeto **Recordset** en el primer elemento de la fila con la instrucción “**MoveFirst**” se recorre una a una las filas del campo y se pone el carácter “.” en cada campo, luego se cambia de fila, de modo que pueda ocurrir eso con todos los campos de las filas. Para que esto ocurra se ha utilizado un arreglo de seis controles **TextBox**, (**Text3(0)-Text3(5)**), cada uno de los cuales está enlazado a un campo de la base de datos.

Tabla # 19 Detalle de propiedad DataField del arreglo Text3.

Elemento de Text3	DataFiled
Text3(0)	horaOn
Text3(1)	horaOff
Text3(2)	horaOn2
Text3(3)	horaOff2
Text3(4)	horaOn3
Text3(5)	horaOff3

De este modo, cada vez que se recorre un campo se pone el carácter “.” En cada fila, utilizando las propiedades **Edit** y **Update**, de modo que el valor sea guardado en la base de datos. Por lo tanto en caso de presionar el botón “Limpiar Lista”, se guardará en todas las horas programables el valor “.”, de manera que se puedan ingresar todos los valores de nuevo.

Una vez que el operador con privilegios de administrador desee que la secuencia programada se inicie simplemente debe presionar el

botón “Ejecutar Secuencia”. Una vez que lo haga se lleva a cabo lo descrito en el código siguiente:

```
“Private Sub ejecutarsec_Click()  
    ejecutarsecuencia  
    Data1.Recordset.MoveFirst  
    Data2.Recordset.MoveFirst  
    For i = 0 To 5  
        While Not (Data2.Recordset.EOF = True)  
            Text4(i).Text = Data1.Recordset(i + 1)  
            Data2.Recordset.Edit  
            Data2.Recordset.Update  
            Data1.Recordset.MoveNext  
            Data2.Recordset.MoveNext  
        Wend  
        Data2.Recordset.MoveFirst  
        Data1.Recordset.MoveFirst  
    Next  
    Data1.Recordset.MoveFirst  
    Data2.Recordset.MoveFirst  
    Timer2.Enabled = True  
End Sub”
```

Que en primer lugar llama a la subrutina **ejecutarsecuencia** (programada en el módulo **Module1**), que tiene el siguiente segmento de código:

```
“Public Sub ejecutarsecuencia()  
    Plantas.ejecutarsec.Visible = False  
    Plantas.detenersec.Visible = True  
    Plantas.LimpiarLista.Visible = False  
    Plantas.DBGrid1.Visible = False  
    Plantas.DBGrid2.Visible = True
```

```

Plantas.Programar.Visible = False
Plantas.Frame1.Visible = False
End Sub”

```

Que oculta a los botones “Ejecutar Secuencia”, “Programar” y “Limpiar Lista”, debido a que estas acciones no pueden hacerse mientras se ejecuta el programa. Además oculta el **DBGrid1**, debido a que en dicho control se está ejecutando la secuencia, y esto no permite que el usuario pueda visualizar todos los elementos el control **DBGrid1** libremente. De manera que muestra un **DBgrid** en el que están todos los elementos presentes en el **DBGrid1** pero puedan sus elementos ser visualizados por el usuario.

A continuación, una vez ejecutada la subrutina “**ejecutarsecuencia**”, se ubican los objetos **Recordset** del control **Data1** y **Data2** en el primer elemento del campo, y de manera similar a la que se explicó al realizar la rutina para limpiar la lista, se copian los valores de la base de datos que se está utilizando, en una base de datos con los mismos campos, para que sea visualizado de igual forma. Se utiliza un arreglo de controles **TextBox (Text4)** para enlazar los campos de la segunda base de datos. La propiedad **DataSource** tiene el valor **Data2 (base de datos “C:\TesisVB1\luces2.mdb”)**, debido a que es la fuente a donde están enlazados sus campos.

Tabla # 20 Detalle de propiedad **DataField** del arreglo **Text4**.

Elemento de	DataFiled
Text4	
Text4(0)	horaOn
Text4(1)	HoraOff
Text4(2)	horaOn2

Text4(3)	horaOff2
Text4(4)	horaOn3
Text4(5)	horaOff3

Los campos llevan nombres iguales a los de la base de datos “**Luces**”, (“C:\TesisVB1\luces.mdb”).

Finalmente se habilita el control **Timer2**, que en su evento “**Time**” el siguiente segmento de código:

```
“Private Sub Timer2_Timer()  
    hora = Format(Now, "h:nn")  
    a = 1  
    While a <= 6  
        n = 0  
        While Not (Data1.Recordset.EOF = True)  
            If Data1.Recordset(a) = hora Then  
                encenderluz (n)  
                verluzon (n)  
            End If  
            If Data1.Recordset(a + 1) = hora Then  
                apagarluz (n)  
                verluzoff (n)  
            End If  
            n = n + 1  
            Data1.Recordset.MoveNext  
        Wend  
        Data1.Recordset.MoveFirst  
        a = a + 2  
    Wend  
End Sub”
```

La propiedad **Interval** del control **Timer2** tiene el valor de “1000”, lo que significa que todas las líneas de código dentro de esta propiedad se ejecutarán cada segundo.

Las líneas de código significan que se recorrerá todas las filas y todos los campos que poseen información de la hora a la que se programó que una luz se encienda o apague y en caso de llegar a coincidir la hora actual, con la programada, se buscará esa luz y se la encenderá o apagará, utilizando las mismas subrutinas anteriormente explicadas (**encenderluz** y **apagarluz**). Además podrá visualizarse en el programa la luz respectiva que ha sido encendida o apagada por el programa.

En cualquier momento el operador puede presionar el botón “Detener Secuencia” que tiene el segmento de código siguiente:

```
“Private Sub detenersec_Click()  
    detenersecuencia  
    Programar.Caption = "Programar Luces"  
End Sub”
```

Se ejecuta la sub rutina detener secuencia programada en el módulo **Module1**:

```
“Public Sub detenersecuencia()  
    Plantas.detenersec.Visible = False  
    Plantas.ejecutarsec.Visible = True  
    Plantas.Timer2.Enabled = False  
    Plantas.DBGrid2.Visible = False  
    Plantas.DBGrid1.Visible = True
```

```
Plantas.Programar.Visible = True  
End Sub”
```

Que deshabilita el control **Timer2** deteniendo de esta forma la rutina de comparación y encendido/apagado de luces. Además, muestra nuevamente el control **DBGrid1**, para que se pueda trabajar en la base de datos a utilizar para la secuencia de luces (“C:\TesisVB1\luces.mdb”).

3.4.4 DISEÑO E IMPLEMENTACIÓN DE LA VENTANA GESTIÓN DE ALARMAS.

Cuando el operador que está utilizando el software HMI, ha ingresado como administrador, tiene acceso al formulario “**cuadrodecontrol**”, que es de donde se puede controlar las alarmas de intrusión y presencia, el nombre visible de este formulario es “Controles”, el modo de acceder a este formulario es por medio de una imagen (evento **click**).



Fig. 95 Gestión de alarmas.

Una vez que se ha accedido el formulario fue llenado utilizando el control **TabStrip**, que gráficamente brinda la comodidad de una visualización de los elementos como en un separador de cuaderno. Está conformado por cinco etiquetas.

3.4.4.1 Acerca de HMI



Fig. 96 Acerca de HMI.

Al seleccionar esta etiqueta es desplegada la información referente al programa, en el sentido de autoría.

3.4.4.2 Información de controles

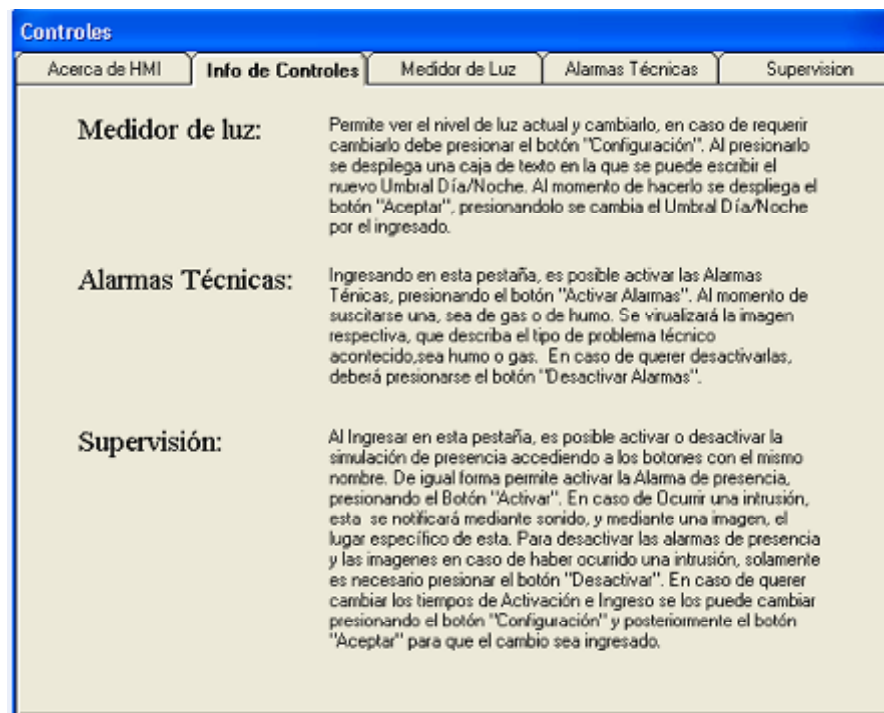


Fig. 97 Info de controles.

Una vez que esta etiqueta ha sido seleccionada, puede verse las acciones que realizan los controles, para facilitar al operador el manejo de los controles.

3.4.4.3 Medidor de luz

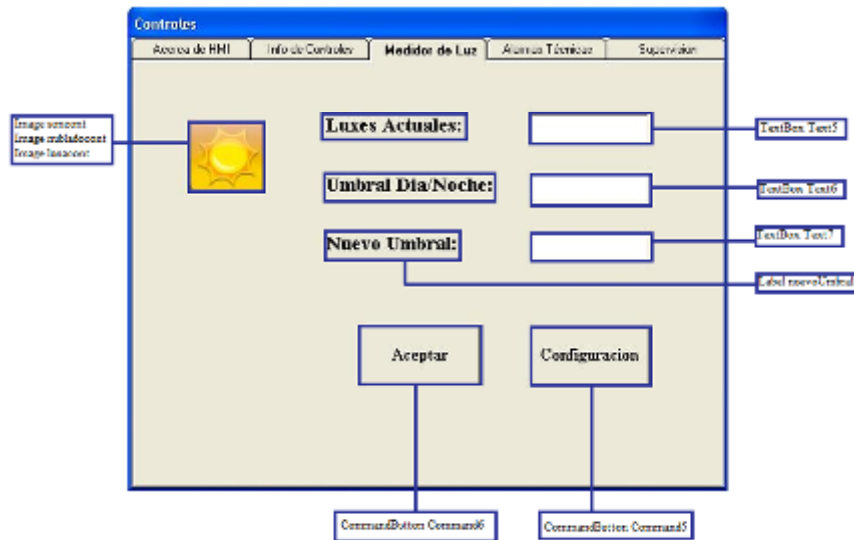


Fig. 98 Medidor de Luz.

Los controles **TextBox**, que hacen referencia a “Luxes Actuales” y “Umbral Día/Noche”, (**Text5** y **text6** respectivamente). Solamente son de lectura, mientras que el control **TextBox** que se encuentra frente al **Label** “**nuevoUmbral**”, es de lectura y escritura, es decir que ingresando aquí el valor de luminosidad al que se desee que empiecen a funcionar las luces de la casa con detección de presencia. La propiedad **LinkTopic** de todos los controles **TextBox** mencionados es “**LNSDDE|MoyaAlmeida1.RAIZ.DevNV**”, la propiedad **LinkItem** de cada uno es de acuerdo a la siguiente tabla:

Tabla # 21 Propiedad **LinkItem** de los controles **TextBox** “Medidor de Luz”.

Control TextBox	Propiedad LinkItem
Text5	MedidorLuz.nvoNivelLux
Text6	MedidorLuz.nvoUmbralN1
Text7	MedidorLuz.cfgUmbralN1

Cuando se cambie el valor del “Umbral día/Noche”, es necesario seleccionar el botón “Aceptar”, caso contrario no se efectuará el cambio.

3.4.4.4 Alarmas técnicas

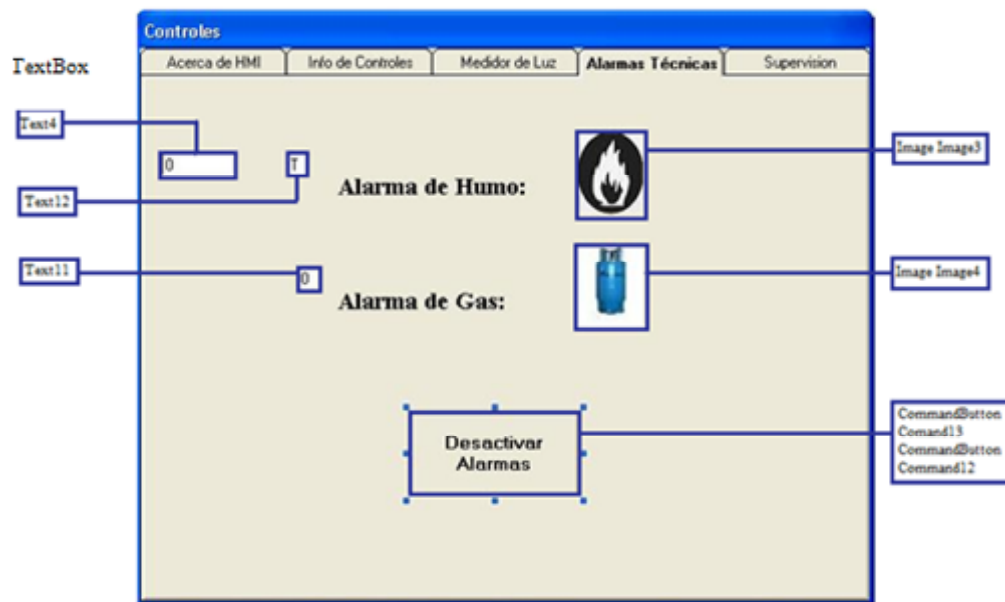


Fig. 99 Alarmas Técnicas.

Se utilizan tres controles **TextBox** que corresponden a las variables de la red que controlan la activación de las alarmas técnicas y el disparo de ellas, se las ha enlazado mediante la propiedad **LinkItem**, correspondiente, tomando en cuenta que la propiedad **LinkTopic** es la misma que se usó para todos los controles del software.

Tabla # 22 Propiedad **LinkItem** de los controles **TextBox** de “Alarmas Técnicas”.

Control TextBox	Propiedad LinkItem
Text4	Alarmas.nvoAlarTecn
Text11	Alarmas.nviActVigilaGas.state
Text12	Alarmas.nviActVigilaHumo.state

Cuando se selecciona el botón “Activar” en tiempo de ejecución se ejecuta el siguiente segmento de código:

```
“Private Sub Command12_Click()
    Text12.Text = "1"
    Text12.LinkPoke
Text11.Text = "1"
    Text11.LinkPoke
    Command12.Visible = False
    Command13.Visible = True
End Sub”
```

Cuando se produce el evento **Click** se envía el valor “1” a los campos **state** de las variables respectivas activando las alarmas de gas y humo. Una vez que están activadas las alarmas, en caso de que se produzca una detección de fuego o de humo, se producirá un cambio en la variable **nvoAlarTecn**, esto implica un cambio en el control **Text4**, produciéndose un evento **Change**, dentro del cual se tiene un segmento de código:

```
“Private Sub Text4_Change()
    If Text4.Text = "5" Then
        Image3.Visible = True
        Plantas.fuego.Visible = True
    Timer2.Enabled = True
    Timer2.Interval = 500
    End If
    If Text4.Text = "ST_HIGH" Then
        Image4.Visible = True
        Plantas.gas.Visible = True
        Timer2.Enabled = True
        Timer2.Interval = 500
    End If
End Sub”
```

De acuerdo al valor que quede en el control **Text4**, se verificará que tipo de alarma técnica se ha producido y se activará un sonido, que está relacionado a un control **Timer**.

```
“Private Sub Timer2_Timer()  
    mciExecute "play " + "C:\SonidosTesis\AlarTecnica.mp3"  
End Sub”
```

Cuando el control Timer2 es activado se ejecuta el archivo de sonido guardado en la ruta indicada en el código. Esto se logra mediante una librería:

```
“Private Declare Function mciExecute _  
Lib "winmm.dll" (_  
ByVal lpstrCommand As String) As Long”
```

Al utilizarla, es posible reproducir archivos de audio desde Visual Basic. Para desactivar las alarmas es necesario solamente presionar el botón “Desactivar Alarmas” y desaparecerá la alerta gráfica como el sonido que se reproduce. De igual forma desaparecerán las alertas presentes en el formulario principal.

3.4.4.5 Supervisión.

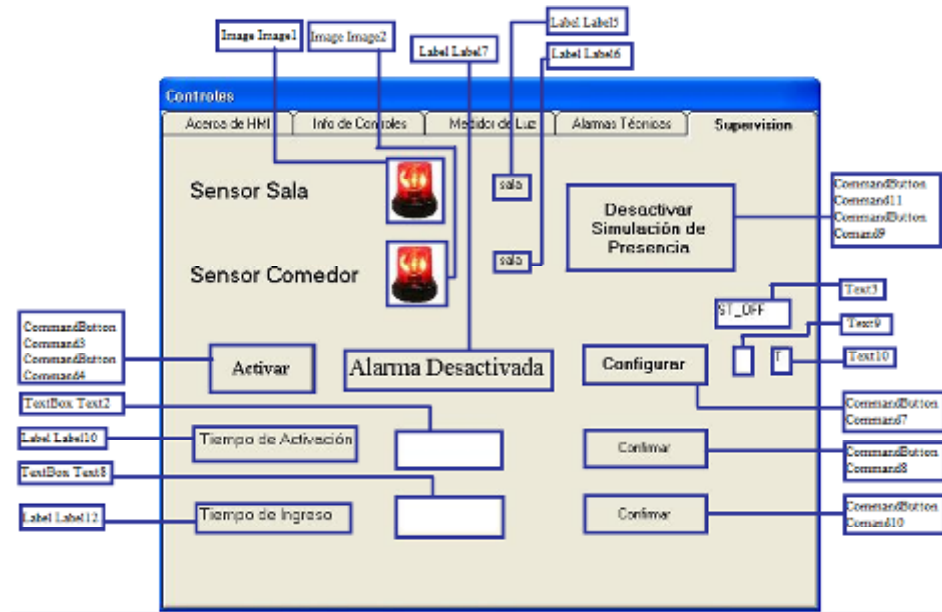


Fig. 100 Supervisión.

En esta etiqueta se han utilizado cinco controles **TextBox**, los cuales tienen su propiedad **LinkItem**, de acuerdo a la siguiente tabla:

Tabla # 23 Propiedad **LinkItem** de los controles **TextBox** de “Supervisión”.

Control TextBox	Propiedad LinkItem
Text2	Alarmas.cfgTimeSalida
Text3	Alarmas.nviActAlarma
Text8	Alarmas.cfgTimeEntrada1
Text9	Alarmas.nvoVigilanciaZ1.state
Text10	Alarmas.nviActSimulPres.state

En caso de que el operador desee activar la simulación de presencia que la red ofrece, al seleccionar el botón “Activar Simulación de presencia”, será enviado el valor “1” al campo **state** de la variable **nviActSimulPres**, lo que producirá la activación de dicha función.

```

Private Sub Command9_Click()
Text10.Text = 1
Text10.LinkPoke
Command9.Visible = False
    
```

```
Command11.Visible = True  
End Sub”
```

Para desactivarla se realizará el proceso inverso, al seleccionar el botón “Desactivar Simulación de Presencia”.

Al seleccionar el botón “Activar”, se ejecutará las instrucciones presentes en el código:

```
“Private Sub Command3_Click()  
Text3.LinkMode = 1  
Text3.Text = "ST_ON"  
Text3.LinkPoke  
Command3.Visible = False  
Command4.Visible = True  
Label7.Caption = "Activando Alarma"  
Label7.ForeColor = &HC00000  
End Sub”
```

Que consiste en enviar el valor “ST_ON” a la variable **nviActAlarma**, produciendo que se active la alarma.

Existe un tiempo de salida, que permite al habitante de una vivienda salir de ella sin que el sistema de alarma detecte una intrusión, la alarma no es activada hasta que este tiempo ha pasado, mientras eso ocurre, se torna visible al mensaje “Activando Alarma”, de modo que se alerte al usuario que se está produciendo una activación.

Una vez que la Alarma esté activada se producirá un cambio en la variable **nvoVigilanciaZ1**, de modo que se producirá un evento **Change** en el control **Text9**, cuyo código es de la siguiente forma:

```
“Private Sub Text9_Change()  
If Text9.Text = 1 Then  
    Command4.Visible = True  
    Command3.Visible = False  
    Label5.LinkMode = 1  
    Label6.LinkMode = 1  
    Label7.Caption = "Alarma Activada"  
    Label7.ForeColor = &HC000&  
End If  
If Text9.Text = 0 Then  
    Command4.Visible = False  
    Command3.Visible = True  
    Label7.Caption = "Alarma Desactivada"  
    Label7.ForeColor = &H0&  
    Image1.Visible = False  
    Image2.Visible = False  
End If  
End Sub”
```

De este modo puede mostrarse visualmente si la alarma está activada o desactivada.

En caso de que la alarma esté activada y se produzca una intrusión, se notificará mediante la aparición de la imagen de una baliza (Fig. 100) en el lugar correspondiente, y su representación respectiva en la página principal (Fig. 80 y Fig. 81), de acuerdo al lugar donde se haya producido (comedor o sala). Adicionalmente se activa un sonido que alerte de la intrusión. Esto es

activado, cuando cambia el estado del respectivo sensor estando activada la alarma, su representación en código es:

```
“Private Sub Label5_Change()  
  If Text9.Text = 1 And Label5.Caption = 0 Then  
    Image1.Visible = True  
    Plantas.Image5.Visible = True  
    Plantas.Image2.Visible = False  
    Plantas.presenciaS.Visible = True  
    Timer1.Enabled = True  
    Timer1.Interval = 500  
  End If  
End Sub
```

```
Private Sub Label6_Change()  
  If Text9.Text = 1 And Label6.Caption = 0 Then  
    Image2.Visible = True  
    Plantas.Image4.Visible = True  
    Plantas.Image3.Visible = False  
    Plantas.presenciaC.Visible = True  
    Timer1.Enabled = True  
    Timer1.Interval = 500  
  End If  
End Sub”
```

El sonido es activado cuando se habilita el controlador **Timer1**.

```
“Private Sub Timer1_Timer()  
  mciExecute "play " + "C:\SonidosTesis\AlarInt.mp3"  
End Sub”
```

Para desactivar la alarma al presionar el botón “Desactivar” se detienen todas las alertas y se desactiva la alarma de presencia.

Es posible configurar el tiempo de salida, que como ya se mencionó es el tiempo que se demora la alarma en ser activada, y el tiempo de ingreso, que es el tiempo que se demora la alarma desde ser detectada una intrusión hasta el momento de sonido de la sirena de alerta.

Para configurar estos tiempos seleccionando el botón “Configurar” se despliegan los controles **TextBox** respectivos que permitirán cambiar el tiempo que se desee, una vez cambiado, al presionar el botón “Confirmar”, el valor es guardado en la variable correspondiente. El segmento de código que realiza esta acción es:

```
“Private Sub Command8_Click()
    Text2.LinkPoke
    Text2.Visible = False
    Command8.Visible = False
    Label10.Visible = False
End Sub
Private Sub Command8_Click()
    Text2.LinkPoke
    Text2.Visible = False
    Command8.Visible = False
    Label10.Visible = False
End Sub
Private Sub Command10_Click()
    Text8.LinkPoke
    Text8.Visible = False
    Command10.Visible = False
    Label12.Visible = False
End Sub”
```

3.5 INTEGRACIÓN DEL HMI EN EL SISTEMA DOMÓTICO

La integración del HMI en el sistema domótico se fue probando, mientras se realizaba el HMI. El funcionamiento del software de supervisión y control tiene un comportamiento como se detalla en el ANEXO # 11.

El software de supervisión y control (HMI), puede estar o no funcionando integrado al sistema domótico, sin que esto afecte al funcionamiento del mismo, todos los dispositivos responderán a la programación que tenga la red.

En caso de que haya una desconexión súbita del software HMI, esto no afectará de ninguna manera a la red.

Es necesario que el interfaz físico USB, sea conectado y reconocido por la PC donde se ejecuta el software HMI, caso contrario pueden existir problemas de conexión, debidos a alguna problema físico de la red, o de comunicaciones.

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

- Comparando el consumo de energía eléctrica producido por la comparación del consumo eléctrico en dos lapsos de tiempo iguales y de la misma época, de años diferentes demuestra que se puede alcanzar una disminución de alrededor del 22,3% de electricidad. Este ahorro se le atribuye a la no existencia de pérdidas debido a luces encendidas innecesariamente por descuido humano.
- En tema de seguridad se ha alcanzado importantes avances, ya que el poder supervisar el bien inmueble contra intrusión, fuego, y gas natural, más la simulación de presencia y los avisos telefónicos, cada vez que se lo requiera disminuye significativamente la probabilidad de daños y/o robos, al punto que podemos considerar al hogar como un bien inmueble más seguro.
- Debido a los bajos costos de la energía eléctrica, a pesar del ahorro energético conseguido no podemos considerar que este es un proyecto económicamente recuperable al corto y mediano plazo, cuando lo comparamos con el alto costo del diseño e implementación de la red domótica. Sin embargo el valor agregado obtenido se reflejará en confort, seguridad, y un importante incremento en la plusvalía del bien inmueble.

- Es importante considerar que LonWorks es un protocolo domótico abierto sumamente estable y confiable, en el cual mediante la consola del LonMaker se observaron muy pocas fallas y pérdidas de paquetes, y cuando ocasionalmente existieron el sistema respondió inmediatamente corrigiéndolas de manera que en ningún momento la red dejó de funcionar.
- Poder tener un HMI añade comodidad al usuario del sistema, debido a que puede tener acceso a todos los controles que requiere desde un solo punto, de una manera fácil e interactiva, sin tener que moverse del lugar en donde tiene instalado dicho HMI, esto es facilitado gracias a la tecnología de comunicación DDE que permite la interacción de la red LonWorks, con Windows.
- El hecho de utilizar un sistema en donde el control no tenga dependencia directa del HMI, hace que el sistema sea robusto. En caso de desconectar súbitamente el interfaz físico de la PC donde se encuentra funcionando el software HMI, el funcionamiento de la red no se ve afectado en manera alguna.
- La utilización de software Visual Basic, para el desarrollo del HMI, da una amplia gama de opciones al momento de programar, lo que permite que se pueda diseñar totalmente la interfaz gráfica del programa, provee de varios controles que pueden ser utilizados para realizar acciones de control y monitoreo del sistema domótico. Sin embargo, requieren de un conocimiento básico de programación al momento de implementar el software que tiene repercusión en el tiempo que toma el desarrollo del HMI, lo que representa un costo de horas hombre alto, dependiendo de la magnitud de la aplicación.
- La implementación de un software a medida, y para una red específica es aplicable únicamente en el lugar para el que fue diseñado el HMI, lo que resta versatilidad al

programa, en caso de realizarse cambios en la red, el software deberá ser corregido para adaptarse a las nuevas necesidades.

- Al utilizar la tecnología de comunicación DDE con una red LonWorks, se requiere de un procedimiento previo que debe realizar el usuario para que el software HMI pueda establecer una conversación DDE, sin este proceso previo, la red no puede ser controlada ni monitoreada por el HMI. Este procedimiento, a pesar de estar presente en el manual de usuario, puede resultar tedioso para el operador, o complicar el acceso al HMI al usuario que lo requiera.
- El costo de la elaboración del HMI con los programas requeridos, considerando que es necesario que la PC que contenga el HMI debe tener el servidor **LNS DDE Server** instalado para que sea posible la comunicación DDE, y las horas-hombre, que representa desarrollar un software tomando en cuenta que el programador debe tener conocimiento de la tecnología de comunicación DDE, para una vivienda resultaría costoso (2800USD aproximadamente), e implicaría un tiempo de elaboración posiblemente extenso, dependiendo de la complejidad del HMI.

RECOMENDACIONES

- La lectura de los datos de consumo en el medidor de energía eléctrica deben hacerse en periodos regulares de tiempo, ya que los datos proporcionados por la Empresa Eléctrica Quito se realizan en un lapso de entre 20 y 30 días con respecto a la lectura anterior, lo cual provoca un error difícilmente corregible. Esta desviación provoca que la medición del mes de junio del 2009, por ejemplo, sea el día 19, y la de junio del 210 el día 25; arrojando así datos que pueden dar a entender un aumento o disminución del consumo de energía eléctrica cuando no fue así necesariamente.

-
- Se debe tener un especial cuidado con las normas técnicas que rigen en el lugar de manufactura de los nodos domóticos (España, en este caso), ya que tendrán características especiales como en el caso de las vinchas que sostienen los cables eléctricos, de sensores y los de comunicaciones y poder, de la cual no se pudo conseguir (en Ecuador) la herramienta específica para la correcta manipulación de las vinchas, obligándonos a utilizar herramientas alternativas que provocaron incomodidad y consecuentemente pérdida de tiempo.
 - Por todo el tiempo y costo hora-hombre que implica el diseño y elaboración del HMI sería ideal poderlo realizar para aplicaciones como el caso de conjuntos habitacionales con diseños similares de vivienda, donde pueda utilizarse el mismo HMI en distintas casas, o con cambios mínimos en la programación, lo que restaría el costo del diseño, y haría más versátil la aplicación.
 - Para poder utilizar el HMI de diferentes puntos de una vivienda, en caso de ser una vivienda grande especialmente, deberían existir varios puntos donde pueda conectarse la PC para ejecutar el HMI, de modo que el usuario tenga mayor comodidad para controlar y monitorear los dispositivos que desee, sin necesidad de movilizarse demasiado, esto además da mayor seguridad al usuario, debido a que no requiere acceder a los controles manuales del sistema.
 - En caso de requerir un HMI para aplicaciones más complejas, donde se requiere que la comunicación se realice con mayor velocidad, se recomienda utilizar tecnologías de comunicación más modernas como OLE, debido a que DDE es una tecnología de comunicación antigua. Por lo tanto las tecnologías más modernas ofrecen mejoras en la velocidad y calidad de comunicaciones.
 - Tomando en cuenta el costo total del diseño de un HMI utilizando el servidor LNS DDE Server, en un software como el Visual Basic, podría considerarse la

utilización de HMI en pantallas táctiles que vienen con un software especializado para diseñar el HMI de manera más rápida, ya que estarían cumpliendo siempre la función de HMI, sin necesidad de un proceso conexión al momento de utilizar, debido a su permanencia en la red. Debido a que el costo es similar al que se produce al diseñar un software utilizando el servidor LNS DDE Server, con el beneficio de tener un HMI funcionando permanentemente en el sistema, sin necesidad de establecer la conversación DDE.

BIBLIOGRAFÍA

- <http://es.wikipedia.org/wiki/Dom%C3%B3tica>, Domótica.
- http://buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=dom%C3%B3tica, Término de domótica.
- <http://www.nova.es/~mromero/domotica/caracter.htm>, Características de la domótica.
- <http://www.domotica-maiher.com/ventajas/>, Ventajas de la domótica.
- ECHELON CORPORATION, **LonMaker User's Guide**, Estados Unidos de América, 2006, 406 págs.
- ISDE, **INM-020 Manual de Usuario**, Versión 1.0, España, 23 págs.
- ISDE, **INM-020_X831200000313_V11**, Versión 1.1, España, 22 págs.
- ISDE, **IFA-200_X9A0700000102_V10**, Versión 1.0, España, 21 págs.
- ISDE, **INM-011_X9B1202000503_V10**, Versión 1.0, España, 26 págs.
- ISDE, **INM-030_X850700000402_V11**, Versión 1.1, España, 17 págs.
- ISDE, **INS-231_X020700000610_V10**, Versión 1.0, España, 25 págs.
- ISDE, **INS-231_X020700000713_V10**, Versión 1.0, España, 27 págs.
- ISDE, **INS-451_X030700000203_V11**, Versión 1.1, España, 29 págs.
- <http://www.elprisma.com/apuntes/curso.asp?id=11616>, 31/05/2010, 14h54.
- <http://www.casadomo.com/noticiasDetalle.aspx?c=14>, 31/05/2010, 15h34.
- www.jmpingenieros.es/domotica/DOMOTICA_2.pdf, 01/06/2010, 9h26.
- <http://www.proyectosdomotica.com/articulos-domotica.php?hogar-digital=64>, 01/06/2010, 15h18.
- <http://topologiadered-kelvin.blogspot.com/>, 01/06/2010, 15h51.
- http://www.lonmark.es/lonworks_tecnologia.asp, 03/06/2010, 10h36.

- [http://casafutura.diatel.upm.es/rrssmd/trabajos/2004/powerpoint/01.-%20Tecnolog%C3%ADa%20LonWorks%20\(J.A.%20Izquierdo\).pdf](http://casafutura.diatel.upm.es/rrssmd/trabajos/2004/powerpoint/01.-%20Tecnolog%C3%ADa%20LonWorks%20(J.A.%20Izquierdo).pdf), 03/06/2010, 12h03.
- <http://www.unizar.es/aeipro/finder/INFORMATICA,TELECOMUNICACIONES/FF10.htm>, 03/06/2010, 15h49.
- http://www.repsol.com/es_es/casa_y_hogar/sin_salir_de_casa/reportajes/domotica/domotica.aspx, 26/07/2010, 15h15.
- http://www.lonmark.es/lonworks_tecnologia.asp, 28/02/2011, 01h46.
- <http://www.zonadeprogramacion.com.ar/vb61.htm>, 04/10/2010, 18h00.
- <http://odisea.ii.uam.es/esp/recursos/Lonwork.htm>, 28/02/2011, 01h50.
- <http://www.instrumentacionycontrol.net/es/curso-supervision-procesos-por-computadora/238-intercambio-de-datos-entre-sistemas-de-automatizacion-y-aplicaciones-software-parte1.html>, 28/02/2011, 03h55.
- [http://interfacemindbraincomputer.wetpaint.com/page/2.A.1.-+Definicion+de+Interaccion+Humano+M%C3%A1quina+\(Interaction-Interface\)](http://interfacemindbraincomputer.wetpaint.com/page/2.A.1.-+Definicion+de+Interaccion+Humano+M%C3%A1quina+(Interaction-Interface)) 28/02/2011, 15h43.
- [http://interfacemindbraincomputer.wetpaint.com/page/2.A.1.-+Definicion+de+Interaccion+Humano+M%C3%A1quina+\(Interaction-Interface\)](http://interfacemindbraincomputer.wetpaint.com/page/2.A.1.-+Definicion+de+Interaccion+Humano+M%C3%A1quina+(Interaction-Interface)), 28/02/2011, 15h36.
- http://148.202.148.5/cursos/cc321/fundamentos/unidad4/tema4_7.html, 28/02/2011, 16h00.
- http://www.echelon.com/products/networktools/lms_dde/default.htm, 04/03/2011, 03h50.

ANEXOS

ANEXO # 1

PLANO ARQUITECTÓNICO

ANEXO # 2

PLANO ELÉCTRICO

ANEXO # 3

NODOS DOMÓTICOS

ANEXO # 4

FUNTIONAL BLOCKS

ANEXO # 5

DETALLE DE FIRMWARES

ANEXO # 6

PLANO DE PREINSTALACIÓN

ANEXO # 7

PLANO DOMÓTICO

ANEXO # 8

PLANOS UNIFILARES

ANEXO # 9

DATASHEETS DE SENSORES, ACTUADORES Y OTROS DISPOSITIVOS

ANEXO # 10

HISTORIA DE FACTURACIONES

ANEXO # 11

MANUAL DE USUARIO DEL HMI

ANEXO # 12

CÓDIGO DEL HMI

ÍNDICE DE CUADROS Y FIGURAS

Fig. 1 Elementos de un hogar digital	2
Fig. 2 Urban Plaza	3
Fig. 3 Ahorro energético en Industrias	8
Fig. 4 Ahorro energético en Edificios	8
Fig. 5 Ahorro energético en Residencias	9
Fig. 6 Ahorro energético	10
Fig. 7 Gestión de la seguridad	10
Fig. 8 Sistemas domóticos centralizados	19
Fig. 9 Sistemas domóticos distribuidos.....	20
Fig. 10 Topología de estrella.....	28
Fig. 11 Topología de bus.....	28
Fig. 12 Topología libre o mixta.....	29
Fig. 13 Estructura de una red LonWorks	36
Fig. 14 Diagrama de bloques interno de un Neuron Chip	39
Fig. 15 Neuron ID.....	40
Fig. 16 Plaza de toros de Navalcanero.....	42
Fig. 17 Villa Olímpica de Beijing.....	43
Fig. 18 Posición del inmueble con respecto al resto de la ciudadela Terracota F.....	58
Fig. 19 Plano arquitectónico. Planta baja.....	59
Fig. 20 Plano arquitectónico. Planta alta.....	60
Fig. 21 Plano eléctrico. Planta baja.....	61
Fig. 22 Plano eléctrico. Planta alta.	62
Fig. 23: Sala y comedor	63
Fig. 24 Estudio.....	63
Fig. 25 Cuarto.....	64
Fig. 26 Sala de estar.....	64
Fig. 27 Cuarto Máster	65
Fig. 28 Planta baja. Plano eléctrico.....	68
Fig. 29 Planta alta. Plano eléctrico.	69
Fig. 30 Panel frontal nodo de supervisión.....	72
Fig. 31 Detalle de los grupos del panel frontal del nodo de supervisión.	72
Fig. 32 Panel frontal nodo telefónico.....	73
Fig. 33 Ventana principal del LonMaker.....	80
Fig. 34 Créditos LonMaker.....	80
Fig. 35 Network Wizard. Nombre de la red y su ubicación en disco duro.	81
Fig. 36 Network Wizard. Network interface.....	82
Fig. 37 Network Wizard. Management Mode.....	83
Fig. 38 Network Wizard. Plug-in Registration.....	84

Fig. 39 Microsoft Visio. Pantalla principal.....	84
Fig. 40 Microsoft Visio. Principales herramientas del LonMaker.....	85
Fig. 41 Botón Device.....	85
Fig. 42 Device Wizard. Pantalla principal.....	86
Fig. 43 Device Wizard. Selección del firmware.....	87
Fig. 44 Device Wizard. Información y Ping Interval.....	88
Fig. 45 Device Wizard. Pantalla informativa final.....	88
Fig. 46 Nodos domóticos agregados.....	89
Fig. 47 Botón Funtional Block.....	90
Fig. 48 Funcional Block Wizard. Pantalla principal.....	91
Fig. 49 Botones para agregar variables de red y conexiones.....	91
Fig. 50 Funtional Block del nodo Fuente.....	92
Fig. 51 Funtional Blocks de todo el sistema domótico.....	92
Fig. 52 Plano de preinstalación domótica. Planta baja.....	111
Fig. 53 Plano de preinstalación domótica. Planta alta.....	112
Fig. 54 Plano domótico. Planta baja.....	113
Fig. 55 Plano domótico. planta alta.....	114
Fig. 56 Plano unifilar.....	115
Fig. 57 Agujero de la Caja principal.....	117
Fig. 58 Agujero para la Caja de registro de la estancia <i>Escaleras</i>	117
Fig. 59 Agujero para el aprovechamiento de las tuberías instaladas.....	118
Fig. 60 Histórico del consumo energético. Comparación entre meses correspondientes al mismo periodo.....	122
Fig. 61 Funcionamiento de LNS Server.....	126
Fig. 62 Ingreso al programa InstalLON_V_1_0_2.....	134
Fig. 63 Ingreso al programa InstalLON_V_1_0_2.....	135
Fig. 64 Ingreso al programa InstalLON_V_1_0_2.....	135
Fig. 65 Selección de la Interfaz física de red.....	136
Fig. 66 Ingreso al LNS DDE Server.....	137
Fig. 67 Selección de la red a controlar.....	137
Fig. 68 Formulario clave_acceso.....	144
Fig. 69 Formulario clave_acceso, como usuario.....	147
Fig. 70 Formulario clave_acceso, como Administrador.....	149
Fig. 71 Formulario clave_acceso, cambiar contraseña.....	151
Fig. 72 Formulario “Plantas”.....	155
Fig. 73 Arreglo de controles PictureBox “bombilla_off”.....	164
Fig. 74 Arreglo de controles PictureBox “bombilla_on”.....	164
Fig. 75 Planos de la vivienda con representaciones de las luces.....	166
Fig. 76 Representación día claro.....	173
Fig. 77 Representación día nublado.....	174
Fig. 78 Representación noche.....	174
Fig. 79 Hora del día.....	174
Fig. 80 Intrusión Comedor.....	175
Fig. 81 Intrusión Sala y Comedor.....	175
Fig. 82 Detección de humo.....	176
Fig. 83 Detección de gas.....	176
Fig. 84 Usuario “Administrador”.....	177
Fig. 85 Programar luces.....	186
Fig. 86 Acceso al administrador visual de datos.....	187
Fig. 87 VisData.....	188

Fig. 88 Campos de bases de datos.	188
Fig. 89 Selección de control DBGrid.	189
Fig. 90 Ícono DBGrid.	190
Fig. 91 DBGrid.	190
Fig. 92 Framel.	191
Fig. 93 Selección de un elemento del DBGrid.	193
Fig. 94 Cambiar hora de encendido de luces.	194
Fig. 95 Gestión de alarmas.	204
Fig. 96 Acerca de HMI.	205
Fig. 97 Info de controles.	206
Fig. 98 Medidor de Luz.	207
Fig. 99 Alarmas Técnicas.	208
Fig. 100 Supervisión.	211

ÍNDICE DE TABLAS

Tabla # 1: Niveles de automatización.....	15
Tabla # 2: Espectro de frecuencias	26
Tabla # 3: Principales características de las tecnologías de redes domóticas	32
Tabla # 4: Nodos a emplearse en el sistema domótico	66
Tabla # 5 Sistemas presentes en el presente proyecto	70
Tabla # 6 Funciones del nodo TELEFÓNICO	75
Tabla # 7 Nombres de los archivos XIF	86
Tabla # 8 Neuron IDs.....	89
Tabla # 9 Conexiones de variables de red de entrada y salida en cada nodo.....	93
Tabla # 10 Variables de configuración de cada nodo.....	96
Tabla # 11 Distribución de nodos en las distintas estancias.....	119
Tabla # 12 Distribución de los sensores, actuadores y demás dispositivos en los nodos. .	119
Tabla # 13 Ahorro energético por mes.....	121
Tabla # 14 Detalle de materiales utilizados en el sistema domótico.....	122
Tabla # 15 Detalle de la mano de obra utilizada en el sistema domótico.....	123
Tabla # 16 Detalle de relación de los arreglos bombilla_on y bombilla_off, respecto al lugar de la luz.....	165
Tabla # 17 Detalle de relación de los arreglos Text2, respecto al lugar de la luz y su propiedad LinkItem.....	169
Tabla # 18 Detalle de propiedad DataField de los controles Label del Frame1.....	192
Tabla # 19 Detalle de propiedad DataField del arreglo Text3.....	199
Tabla # 20 Detalle de propiedad DataField del arreglo Text4.....	201
Tabla # 21 Propiedad LinkItem de los controles TextBox “Medidor de Luz”.....	207
Tabla # 22 Propiedad LinkItem de los controles TextBox de “Alarmas Técnicas”.....	208
Tabla # 23 Propiedad LinkItem de los controles TextBox de “Supervisión”.....	211

GLOSARIO

DOMÓTICA. Se entiende por domótica al conjunto de sistemas capaces de automatizar una vivienda, aportando servicios de gestión energética, seguridad, bienestar y comunicación, y que pueden estar integrados por medio de redes interiores y exteriores de comunicación, cableadas o inalámbricas, y cuyo control goza de cierta ubicuidad, desde dentro y fuera del hogar.

HMI. Medio con que el usuario puede comunicarse con una máquina, un equipo o una computadora, y comprende todos los puntos de contacto entre el usuario y el equipo, normalmente suelen ser fáciles de entender y fáciles de accionar.

INTEROPERABILIDAD. Es la capacidad que tiene un producto o un sistema, cuyas interfaces son totalmente conocidas, para funcionar con otros productos o sistemas existentes o futuros y eso sin restricción de acceso o de implementación.

LNS. Sistema operativo de red para la gestión de la red LonWorks. LNS incluye un servidor de red y una interfaz de programación de aplicaciones (API).

LON. Local Operating Network. Consiste en un conjunto de dispositivos inteligentes, o nodos, que se conectan mediante uno o más medios físicos y que se comunican utilizando un protocolo común.

LONTALK. Ha sido creado dentro del marco del control industrial por lo que se enfoca a funciones de monitorización y control de dispositivos.

LONWORKS. Plataforma de control creada por la compañía norteamericana Echelon.

NEURON CHIP. Es el corazón de la tecnología LonWorks. Contiene toda implementación del protocolo LonTalk. Cada CI *Neuron* tiene tres procesadores de 8-bit, dos dedicados al protocolo y un tercero a la aplicación del nodo.

NODO. Objeto que responde a varias entradas y que produce unas salidas.

TRANSCEIVERS. Estos dispositivos sirven de interface entre el Neuron chip y el medio físico. Dependiendo del medio físico la velocidad de transmisión y topología es distinta

VARIABLES DE RED. Son los datos que intercambian los diferentes nodos para el funcionamiento del sistema. Cada nodo tiene variables de entrada y de salida, que son definidas por el desarrollador.