

UNIVERSIDAD TECNOLÓGICA ISRAEL

FACULTAD DE SISTEMAS INFORMÁTICOS

**CREACIÓN DE CÓDIGOS Y CONTRA CÓDIGOS PARA LA
ACTIVACIÓN DE PROGRAMAS, UTILIZANDO ELEMENTOS DE
INTELIGENCIA ARTIFICIAL.**

**Estudiante
Diego Ricardo Orellana Olivo**

**Tutor
Ing. Vladimir Bonilla**

**Quito Ecuador
Mayo 2010**

UNIVERSIDAD TECNOLÓGICA ISRAEL

FACULTAD DE SISTEMAS INFORMÁTICOS

CERTIFICADO DE RESPONSABILIDAD

Yo, Ing. Vladimir Bonilla, certifico que el Señor Diego Ricardo Orellana Olivo con C.C. No. 1715273742, realizó la presente tesis con título “**Creación de códigos y contra códigos para la activación de programas, utilizando elementos de inteligencia artificial**”, y que es autor intelectual del mismo, que es original, autentica y personal

Ing. Vladimir Bonilla

UNIVERSIDAD TECNOLÓGICA ISRAEL

FACULTAD DE SISTEMAS INFORMÁTICOS

CERTIFICADO DE AUTORÍA

El documento de tesis con título “**Creación de códigos y contra códigos para la activación de programas, utilizando elementos de inteligencia artificial**”, ha sido desarrollado por Diego Ricardo Orellana Olivo con C.C. No. 1715273742, persona que posee los derechos de autoría y responsabilidad, restringiéndose la copia o utilización de cada uno de los productos de esta tesis sin previa autorización.

Diego Ricardo Orellana Olivo

DEDICATORIA

El presente proyecto de grado, va dedicado a Dios y a mis padres, por enseñarme todos los valores que necesita una persona para orientarse por el camino del bien a través del amor, la responsabilidad, la paciencia, la honradez y la perseverancia que me inculcaron, y con todo esto cumplir una más de las metas que tengo propuestas.

También va dedicado con todo mi amor a mis hijos Madeline y Nicolas, quienes son los seres que amo con todo mi corazón y que cada día me inspiran ser mejor, quienes son la razón de mí existir en esta vida y que algún día les sirva de ejemplo que una persona puede llegar lejos cuando se trazan metas y objetivos.

AGRADECIMIENTOS

En general una gran retribución a todas las personas entre familiares y amigos que aportaron de alguna u otra manera para que este proyecto se logre, no quiero poner nombres para no olvidarme de nadie, pero mi memoria los tendrá siempre muy presentes y los llevaré muy guardados en mi corazón.

Un agradecimiento especial a todos los docentes, quienes muchas veces no negaron sus conocimientos y supieron brindarme todo su saber, dentro y fuera de las aulas, sin necesidad de recibir algo a cambio.

A la Universidad Israel que me abrió sus puertas desde el colegio y que gracias a su enseñanza, soy quien soy profesionalmente hoy en día.

RESUMEN

En la actualidad, el desarrollo de software ha crecido notablemente, hasta el punto de que en todas las organizaciones se trabaja al menos con un programa y al hablar de programas, hablamos de información, la cual es muy vital para el desenvolvimiento de cualquier negocio, por lo que a su vez, la piratería está en constante aumento y sucede a nivel mundial, generando grandes pérdidas para toda organización que se dedica al desarrollo de software. Como medio para tratar de impedir estos plagios, se utiliza la Seguridad Informática, que no es más que la protección de Información y de Sistemas de Información, ya sea para evitar su acceso, uso, divulgación, interrupción o destrucción. En el presente proyecto, como Solución, se aplicarán métodos de encriptación, usando Redes Neuronales Artificiales con el método de aprendizaje Hopfield, para proteger de mejor manera el acceso no deseado a programas que se activan por medio de un código o serial, tomado de los periféricos de un computador.

SUMMARY

Currently, the software development has grown significantly, to the point that in all organizations working with at least one program and talk about programs, we talk about information, which is very vital for the development of any business, so turn, piracy is growing steadily and is happening worldwide, generating big losses for any organization that is dedicated to software development. As a means to try to prevent these abductions, use the Computer Security, which is nothing more than the protection of information and information systems, either to prevent access, use, disclosure, disruption or destruction. In this project, such as solution, apply encryption methods using artificial neural networks with the Hopfield learning method, to better protect unwanted access to programs that are activated by a code or serial, taken from peripherals of a computer.

Contenido

1.	INTRODUCCIÓN	5
1.1	Antecedentes	5
1.2	Planteamiento del Problema	7
1.3	Sistematización	8
1.3.1	Diagnóstico.....	8
1.3.2	Pronóstico	10
1.3.3	Control del pronóstico	11
1.4	Objetivos	13
1.4.1	Objetivo General	13
1.4.2	Objetivos Específicos	13
1.5	Justificación.....	14
1.5.1	Teórica	14
1.5.2	Metodológica.....	14
1.5.3	Práctica	14
1.6	Alcance	15
1.7	Estudio de factibilidad	16
1.7.1	Técnica.....	16
1.7.2	Económica.....	19
1.7.3	Operacional	21
2.	MARCO DE REFERENCIA	22
2.1	Marco Teórico	22
2.2	Marco Conceptual	24
2.2.1	Ingeniería de Software	24
2.2.2	Metodología Orientada a Objetos.....	26
2.2.3	Microsoft Solutions Framework (MSF)	27
2.2.4	Unified Modeling Language (UML).....	29
2.2.5	Inteligencia Artificial.....	38
2.2.6	Redes Neuronales Artificiales (RNA)	39
2.2.7	Perceptrón.....	40
2.2.8	Hopfield	41

2.2.9	BackPropagation.....	42
2.2.10	Seguridad Informática.....	42
2.3	Marco Legal.....	44
2.4	Marco Témporo / Espacial.....	45
3.	PROCESO METODOLÓGICO.....	46
3.1	Metodología de Investigación.....	46
3.1.1	Unidad de Análisis.....	46
3.1.2	Tipo de investigación.....	46
3.1.3	Métodos de investigación.....	46
3.1.4	Técnicas y herramientas de investigación.....	47
3.2	Metodología Informática.....	47
3.2.1	Programación Orientada a Objetos.....	47
3.2.2	Microsoft Solutions Framework.....	49
4.	PROCESO DE DESARROLLO.....	51
4.1	Microsoft Solutions Framework.....	51
4.1.1	Visión.....	51
4.1.2	Planeación.....	64
4.1.3	Desarrollo.....	76
4.1.4	Estabilización.....	81
4.1.5	Implantación.....	82
5.	CONCLUSIONES Y RECOMENDACIONES.....	83
5.1	CONCLUSIONES.....	83
5.2	RECOMENDACIONES.....	84
	BIBLIOGRAFÍA.....	85
	ANEXOS.....	86

LISTA DE GRÁFICOS

Ilustración 1. Pantalla de Ingreso a ECUAKARAOKE	8
Ilustración 2. MSF (Microsoft Solutions Framework v3.1).....	27
Ilustración 3. Ejemplo Diagrama de clases	29
Ilustración 4. Ejemplo Diagrama de Componentes.....	30
Ilustración 5. Ejemplo Diagrama de Objetos	31
Ilustración 6. Ejemplo Diagrama de Despliegue.....	32
Ilustración 7. Ejemplo Diagrama de Paquetes.....	33
Ilustración 8. Ejemplo Diagrama de Actividades	34
Ilustración 9. Ejemplo Diagrama de Casos de Uso	35
Ilustración 10. Ejemplo Diagrama de Estados.....	36
Ilustración 11. Ejemplo Diagrama de Secuencia.....	37
Ilustración 12. Ejemplo Diagrama de Colaboración.....	38
Ilustración 13. Red Neuronal Perceptrón	40
Ilustración 14. Función Red Neuronal Perceptrón.....	40
Ilustración 15. Función Tipo Escalón Red Neuronal Hopfield.....	41
Ilustración 16. Red Neuronal de Hopfield Aplicada al Proyecto.....	41
Ilustración 17. Red Neuronal BackPropagation.....	42
Ilustración 18. Caso de Uso ECUAKARAOKE	67
Ilustración 19. Caso de Uso Iniciar Sesión.....	67
Ilustración 20. Caso de Uso Pantalla Principal.....	68
Ilustración 21. Caso de Uso Pantalla Cliente	68
Ilustración 22. Diagrama General de Casos de Uso.....	69
Ilustración 23 Diagrama de Secuencia Iniciar Sesión.....	74
Ilustración 24 Diagrama de Secuencia Pantalla Principal.....	74
Ilustración 25 Diagrama de Secuencia Cliente / Distribuidor	75
Ilustración 26 Diagrama de Secuencia ECUAKARAOKE	75
Ilustración 27. Diagrama de Clase Método de Activación (MDA)	77
Ilustración 28. Diagrama de Clase Capa Entidad de Negocio Control de Clientes	78
Ilustración 29. Diagrama de Clases Capa Lógica de Negocio Control de Clientes	79
Ilustración 30. Diagrama de Clases Capa Presentación Control de Clientes.....	80
Ilustración 31. Reporte de Pruebas con NUnit 2.0	81
Ilustración 32. Nueva Pantalla de Ingreso a ECUAKARAOKE	87
Ilustración 33. Pantalla de Ingreso Prototipo de Control.....	87
Ilustración 34. Pantalla Principal	88
Ilustración 35. Pantalla Cliente.....	88
Ilustración 36. Pantalla Generar Códigos.....	89
Ilustración 37. Pantalla Historial de Códigos	89
Ilustración 38. Pantalla Nuevo Cliente.....	90
Ilustración 39. Pantalla Nuevo Teléfono.....	90
Ilustración 40. Pantalla Nuevo Negocio.....	91

LISTA DE TABLAS

Tabla 1. Porcentaje de copias ilegales en el programa ECUAKARAOKE.....	7
Tabla 2. Direcciones MAC idénticas en los computadores.....	9
Tabla 3. Matriz de Ponderación Redes Neuronales	16
Tabla 4. Matriz de Ponderación Sistema Operativo	17
Tabla 5. Matriz de Ponderación Base de Datos	17
Tabla 6. Matriz de Ponderación Lenguaje de Programación.....	18
Tabla 7. Requerimientos de Software	19
Tabla 8. Requerimiento de Personal	19
Tabla 9. Flujo de Caja	20
Tabla 10. Roles, Tareas y Entregables en la Fase de Visión.....	55
Tabla 11. Roles, Tareas y Entregables en la Fase de Planificación.....	56
Tabla 12. Roles, Tareas y Entregables en la Fase de Desarrollo	57
Tabla 13. Roles, Tareas y Entregables en la Fase de Estabilización.....	58
Tabla 14. Roles, Tareas y Entregables en la Fase de Implantación	59
Tabla 15. Equipo de Trabajo	59
Tabla 16. Matriz de Riesgos.....	62
Tabla 17. Plan de Desarrollo.....	64
Tabla 18. Actores y Descripción que intervienen en el proyecto.....	66
Tabla 19. Especificación Caso de Uso Ingresar a ECUAKARAOKE.....	70
Tabla 20. Especificación Caso de Uso Iniciar Sesión	71
Tabla 21. Especificación Caso de Uso Pantalla Principal	72
Tabla 22. Especificación Caso de Uso Cliente / Distribuidor.....	73
Tabla 23. Diagrama de Base de Datos.....	76

1. INTRODUCCIÓN

1.1 Antecedentes

En la actualidad, el desarrollo de software ha crecido notablemente, hasta el punto de que en todas las organizaciones se trabaja al menos con un programa y al hablar de programas, hablamos de información, la cual es muy vital para el desenvolvimiento de cualquier negocio, por todo esto hay que dar mucha importancia a las siguientes preguntas:

¿Qué importancia tiene esa información?

¿Cómo proteger la información?

¿Quién está autorizado a mirar esa información?

Es un dilema que toda empresa desarrolladora tiene, y pocas se preocupan en hacer algo para proteger sus programas, siendo fácilmente vulnerados de alguna forma, ya sea con cracks¹ o keygens².

Actualmente la piratería está en constante aumento y sucede a nivel mundial, generando grandes pérdidas para toda organización que se dedica al desarrollo de software, existiendo la decepción de encontrar su obra vendiéndose en cualquier lugar a precios sumamente bajos, cuando la Propiedad Intelectual fue fruto de mucho trabajo, esfuerzo y dedicación.

Como medio para tratar de impedir estos plagios, se utiliza la Seguridad Informática, que no es más que la protección de Información y de Sistemas de Información, ya sea para evitar su acceso, uso, divulgación, interrupción o

¹ Modificar el comportamiento o ampliar la funcionalidad de un programa.

² Programa generalmente ilegal, usado para generar códigos que sirven para activar programas pagados.

destrucción; teniendo como principio básico la CIA (Confidentiality, Integrity, Availability) Confidencialidad³, Integridad⁴ y Disponibilidad⁵, utilizando técnicas como son: Codificar la Información, Tecnologías repelentes o protectoras, Detección y prevención de intrusos, entre otras.

Como Solución, se aplicarán métodos de encriptación y desencriptación, usando Redes Neuronales Artificiales (RNA)⁶, para proteger de mejor manera el acceso no deseado a programas que se activan por medio de un código o serial.

³ Prevenir la dispersión de información a personas no autorizadas.

⁴ Proteger que los datos sean modificados sin autorización.

⁵ Cuando la información está disponible para quienes deben acceder a ella.

⁶ Abreviación de Red Neuronal Artificial.

1.2 Planteamiento del Problema

Actualmente existen varias copias ilegales del programa ECUAKARAOKE y esto está en constante aumento, generando grandes pérdidas para la organización, existiendo la decepción al encontrar su desarrollo vendiéndose en cualquier lugar a precios sumamente bajos, cuando la Propiedad Intelectual fue fruto de mucho trabajo, esfuerzo y dedicación.

Tabla 1. Porcentaje de copias ilegales en el programa ECUAKARAOKE

CUADRO DE PROGRAMAS ECUAKARAOKE				
CANTIDAD	LEGAL	ILEGAL	% ILEGAL	SONDEO
2000	500	1500	75,00%	Ecuador

Fuente: el Autor. Datos aproximados obtenidos a través de la Empresa ECUAKARAOKE, los cuales se obtuvieron, a través de visitas a los establecimientos existentes en el Ecuador.

¿Cuáles son los elementos de la Inteligencia Artificial que permiten proteger la seguridad de un software, mediante la encriptación?

¿Cuáles elementos físicos de un computador se podrían tomar como referencia para generar un código de activación único para cada equipo?

1.3 Sistematización

1.3.1 Diagnóstico

El programa ECUAKARAOKE hoy en día tiene un método de activación que consiste en obtener del computador la dirección MAC⁷ de alguna tarjeta de red disponible o a su vez del Id del procesador⁸ (si no existiera tarjeta de red), estos identificadores están en notación hexadecimal, se los convierte a notación decimal, se realizan cálculos básicos como suma, resta, multiplicación y división, generando un número único de identificación para cada computador, con dicho serial se procede a realizar otros cálculos como los anteriores, obteniendo un código de activación único para el uso del programa.



Ilustración 1. Pantalla de Ingreso a ECUAKARAOKE

⁷ Media Access Control Address: Es un identificador de 48 bits único, proveniente de una tarjeta o interfaz de red.

⁸ Id del procesador: Número de serie o identificador de un procesador.

En este método de activación se han encontrado algunas falencias que permiten el uso del programa en muchas computadoras sin la debida autorización legal, a continuación se detallan los errores hallados:

Existen computadores que no tienen tarjeta de red o en algunos casos las desactivan ya sea desde el BIOS⁹ o desde el Sistema Operativo, al ocurrir esto, con el método de activación actual de ECUAKARAOKE se crea el Serial por medio de una de las tres opciones siguientes:

- Varios computadores poseen dos puertos con direcciones MAC idénticas:

Tabla 2. Direcciones MAC idénticas en los computadores

NOMBRE	DIRECCIÓN MAC
Minipuerto WAN (PPTP)	50:50:54:50:30:30
Minipuerto WAN (PPPOE)	33:50:6F:45:30:30

- Al no encontrar tarjeta de red en el computador, se crea un número de serie a través del Id del procesador.
- Existen programas que permiten cambiar las direcciones MAC de las tarjetas de red existentes en un computador, con esto se puede clonar la dirección **MAC de un equipo que ya tiene activado el programa ECUAKARAOKE.**

En conclusión, ya sea por no existir una tarjeta de red, estar desactivada, que el procesador sea el mismo o que se clone la dirección MAC de otro

⁹ BIOS: (Basic Input-Output System) Sistema Básico de Entrada/Salida, software muy básico instalado en la placa base que permite que ésta cumpla su cometido.

computador, el método de activación actual de ECUAKARAOKE crea un serial idéntico en los computadores con esas características, activándose de igual forma con el mismo código, produciéndose con esto que el programa funcione sin la debida autorización.

Al momento, ECUAKARAOKE no cuenta con un programa de control códigos que permita acceder de manera rápida a los respectivos seriales y códigos de cada cliente; éstos son guardados en una hoja de cálculo Excel.

También podría ocurrir que los usuarios al adquirir software pirata, en muchos de los casos pueden contener virus o a su vez no funcionar correctamente, y por no haberlo obtenido de una manera legal, pierde derechos sobre información, soporte, actualización, parches o ampliaciones en el producto.

Según la IDC¹⁰ (International Data Corporation) y BSA¹¹ (Business Software Alliance), el 67% de software usado en Ecuador es ilegal, lo que generó aproximadamente 35 millones de dólares en pérdidas en el 2009.

1.3.2 Pronóstico

Al existir errores en el método de activación actual de ECUAKARAOKE y generarse los mismos seriales en diferentes computadores, nos lleva a que el programa se active con un mismo código, aumentándose las copias ilegales de ECUAKARAOKE, con esto, las amenazas de piratería seguirán latentes y por ende aumentará cada vez más las pérdidas en la empresa.

¹⁰ Líder global de inteligencia de mercados, firma de consultoría y eventos en las industrias de Tecnologías de la Información (TI), Telecomunicaciones y Tecnología para Consumidores.

¹¹ Voz de la industria del software comercial y sus socios de la industria del hardware ante los gobiernos y en el mercado internacional.

Por no contar con una herramienta automatizada para el control de códigos y seriales de los clientes, se hace difícil encontrar los datos respectivos al momento que el usuario lo necesita.

El usuario al seguir usando software ilegal se mantendrá en constantes problemas con su computador, ya sea con virus o malfuncionamiento, incluso podría llegar a pensar que el problema es del software, en este caso la mala reputación se lleva la empresa que lo creo y no la persona que modificó el programa para su uso ilegal.

Si no se busca alguna manera para reducir el índice de software ilegal en el Ecuador, lo único que se generará es más piratería y más pérdidas de dinero.

1.3.3 Control del pronóstico

Para proteger el uso no autorizado y copias ilegales del programa ECUAKARAOKE, se creará un método de encriptación y desencriptación de datos, aplicando Redes neuronales Artificiales con el algoritmo de aprendizaje Hopfield¹², el cual va a generar seriales y códigos de activación para programas que se activan mediante este mecanismo, los cuales serán guardados en la misma red y en una base de datos como respaldo para ayudar al cliente en el caso de inconvenientes. Los seriales serán un código único para cada PC, que se lo creará tomando una dirección MAC del computador, filtrando las direcciones MAC que son iguales en diferentes computadores, si ésta no existiera o estuviese desactivada, se tomarán datos como el Id del procesador, Id de Tarjeta madre y el Id del

¹² Algoritmo de aprendizaje no supervisado usado para entrenar redes neuronales artificiales.

Disco Duro, al ser tres las combinaciones para crear el serial se reducirá la duplicación de los mismos.

En el caso que se llegue a clonar una dirección MAC, se añadirá un método que borrará las direcciones MAC clonadas, cambiándolas a las que vienen originalmente de fábrica.

También se realizará el desarrollo de un prototipo para el control de los seriales y códigos obtenidos por el módulo de activación, éste brindará la información necesaria de manera ágil el momento que los clientes lo necesiten.

El módulo de generación de códigos a través de redes neuronales, podría ser usado por empresas desarrolladoras para lograr mayor seguridad en sus programas.

1.4 Objetivos

1.4.1 Objetivo General

Desarrollar e implementar un módulo que permita crear seriales y códigos de activación para el programa ECUAKARAOKE a través de una Red Neuronal.

1.4.2 Objetivos Específicos

Examinar los datos de los periféricos del computador que se piensan usar.

Analizar el método de aprendizaje que se aplicará en la red neuronal, para la encriptación de datos.

Desarrollo e implementación del módulo para crear códigos y seriales de activación, para mejorar la seguridad del programa ECUAKARAOKE.

Diseñar e implementar el prototipo de control de seriales y códigos de activación, para agilizar la atención al usuario.

1.5 Justificación

1.5.1 Teórica

Mediante la utilización de lenguajes de programación, Redes Neuronales Artificiales y Seguridad Informática, encontrar alternativas para desarrollar una clase para la seguridad en la activación de programas, aplicándolos en conjunto para encontrar un mecanismo seguro de protección de software.

1.5.2 Metodológica

Se realizarán encuestas a las empresas desarrolladoras, se aplicará metodologías de investigación y de desarrollo existentes, que permita alcanzar el Objetivo propuesto.

1.5.3 Práctica

Cumpliendo los objetivos mencionados en la investigación, se llegará a desarrollar e implementar un método de activación por códigos en los programas usando Redes Neuronales Artificiales, para controlar el uso y copias ilegales de programas, aumentando rentabilidad en las empresas de desarrollo.

1.6 Alcance

Se creará un prototipo de encriptación utilizando Redes Neuronales Artificiales con aprendizaje Hopfield, en donde se entrenará a la red neuronal para crear seriales y códigos para activar programas, además se creará un prototipo para el Control de Códigos y Seriales.

Los códigos y seriales serán únicos para cada computador

Se enfocará con perfiles de usuario para el ingreso.

El prototipo tendrá una pantalla principal, que contendrá el acceso a las Interfaces de Cliente / Distribuidor y Salir.

La Interfaz Cliente Distribuidor, tendrá dos pestañas con el acceso a las interfaces de Generar Códigos e Historial de Códigos; esta Interfaz de Cliente también llamará a una Interfaz Nuevo Cliente.

Desde la Interfaz Nuevo Cliente se tendrá el acceso a las interfaces de Nuevo Teléfono y Nuevo Negocio.

La Interfaz Nuevo Cliente, permitirá guardar los datos del Cliente (Cédula, Nombre, Dirección, Correo, Teléfonos, Fecha de nacimiento, País, Provincia, Cantón, Negocios, Vendedor)

En la Interfaz Generar Códigos se podrá generar y guardar el código y el serial de activación de acuerdo a la versión escogida, tipo de Actualización, que negocio y un campo para observaciones.

El prototipo de control de códigos y seriales, solo permitirá guardar los datos del cliente, los códigos y seriales generados y el historial de información; no se permitirá modificar ni eliminar ningún dato.

El módulo de activación y el prototipo de control de códigos y seriales, será creado solo para pruebas con el programa ECUAKARAOKE.

El módulo y el prototipo a pesar de estar diseñados por capas, no trabajarán en ambiente distribuido, está destinado solo para el uso en un computador.

1.7 Estudio de factibilidad

1.7.1 Técnica

Se debe analizar los requerimientos de software necesarios para poner en marcha el proyecto propuesto:

Requerimientos de Redes Neuronales

Tabla 3. Matriz de Ponderación Redes Neuronales

Red Neuronal	Ponderación						
		HOPFIELD		BACKPROPAGATION		PERCEPTRON SIMPLE	
		Valor	Total	Valor	Total	Valor	Total
Aprendizaje Supervisado	14,29%	1	0,14	5	0,71	5	0,71
Aprendizaje NO Supervisado	25,00%	5	1,25	1	0,25	1	0,25
Monocapa	25,00%	5	1,25	2	0,50	5	1,25
Multicapa	14,29%	1	0,14	5	0,71	1	0,14
Capacidad de Almacenamiento	21,43%	3	0,64	4	0,86	2	0,43
Valores	100,01%		3,43		3,04		2,79
Excelente	5						
Muy bueno	4						
Bueno	3						
Regular	2						
Malo	1						

La Red Neuronal que se acopla para éste proyecto es Hopfield, por ser de aprendizaje no supervisado y monocapa, que es lo que se necesita para cifrar el código de activación.

Requerimientos de Software

Sistema Operativo

Tabla 4. Matriz de Ponderación Sistema Operativo

Sistema Operativo	Ponderación						
		WIN XP		WIN VISTA		LINUX	
		Valor	Total	Valor	Total	Valor	Total
Multitarea	15,38%	4	0,62	4	0,62	4	0,62
Multiusuario	15,38%	4	0,62	4	0,62	4	0,62
Experiencia del Usuario	23,08%	5	1,15	4	0,92	3	0,69
Seguridad	23,08%	3	0,69	3	0,69	4	0,92
Eficiencia	23,08%	4	0,92	3	0,69	4	0,92
Valores	100,00%		4,00		3,54		3,77
Excelente	5						
Muy bueno	4						
Bueno	3						
Regular	2						
Malo	1						

El Sistema Operativo a usarse para el desarrollo del proyecto será Windows XP.

Base de datos

Tabla 5. Matriz de Ponderación Base de Datos

Base de Datos	Ponderación						
		SQL 2005 EXPRESS		ORACLE 10G		MYSQL 5	
		Valor	Total	Valor	Total	Valor	Total
Escalabilidad y Rendimiento	20,52%	3	0,62	4	0,82	2	0,41
Administración	12,82%	4	0,51	3	0,38	4	0,51
Seguridad	20,51%	4	0,82	4	0,82	3	0,62
Programabilidad	12,82%	4	0,51	4	0,51	3	0,38
Integración e interoperabilidad	12,82%	4	0,51	4	0,51	2	0,26
Costos	20,51%	3	0,62	2	0,41	5	1,03
Valores	100,00%		3,59		3,46		3,21
Excelente	5						
Muy bueno	4						
Bueno	3						
Regular	2						
Malo	1						

De acuerdo a la matriz, SQL Server Express es la Base de Datos que mejor se ajusta al proyecto.

Lenguaje de programación

Tabla 6. Matriz de Ponderación Lenguaje de Programación

Lenguaje de Programación	Ponderación						
		C#		JAVA		PHP	
Características		Valor	Total	Valor	Total	Valor	Total
Portabilidad	17,50%	4	0,70	4	0,70	4	0,70
Independencia	17,50%	3	0,53	4	0,70	4	0,70
Experiencia	12,50%	4	0,50	2	0,25	3	0,38
Utilidad	15,00%	4	0,60	4	0,60	3	0,45
Rendimiento	22,50%	4	0,90	4	0,90	4	0,90
Flexibilidad	15,00%	3	0,45	3	0,45	3	0,45
Valores	100,00%		3,68		3,60		3,58
Excelente	5						
Muy bueno	4						
Bueno	3						
Regular	2						
Malo	1						

El Lenguaje escogido para el desarrollo es Microsoft Visual C#, el cual es orientado a objetos y también escogido por conocimiento y experiencia.

1.7.2 Económica

ECUAKARAOKE cuenta con el hardware necesario para el funcionamiento del proyecto, la inversión a realizarse será por licencias de los programas a utilizarse y el personal para el desarrollo.

Requerimientos de Software

Tabla 7. Requerimientos de Software

PRODUCTO	PRECIO
Microsoft Windows XP SP2	175
Microsoft Visual Studio Standard Edition 2005	70
Microsoft SQL Server Express	0
TOTAL	245

Requerimiento de Personal

Tabla 8. Requerimiento de Personal

PERSONAL	PRECIO
Análisis y Diseño	1500
Programación	500
TOTAL	2000

1.7.3 Operacional

El impacto que pueda ocurrir a los usuarios que usan el programa ECUAKARAOKE es transparente, debido a que el cliente se centra en la funcionalidad del Karaoke, más no en la seguridad que tiene el programa.

La empresa ECUAKARAOKE está dispuesta a utilizar los nuevos métodos de seguridad, porque su principal preocupación es la copia ilegal de su producto.

El desarrollo del módulo de seguridad va orientado a empresas desarrolladoras de software, como apoyo para la protección de sus productos, lo cual tiene la ventaja de que va a ser usado solo por quien lo necesite, garantizándose la acogida y uso del mecanismo de protección.

2. MARCO DE REFERENCIA

Para el progreso del proyecto, existen varios requisitos, como son los conocimientos adquiridos a través de la carrera, conceptos y teorías que servirán de guía para un buen desenvolvimiento en el desarrollo del producto.

2.1 Marco Teórico

Para el desarrollo de este proyecto, a través de la **Ingeniería de Software** se aplicará la **Metodología Orientada a objetos**, el Proceso de Desarrollo Microsoft Solutions Framework (**MSF**) y utilizando **UML** como Herramienta de Modelado.

Con la ayuda de la Ingeniería de software, se realizará paso a paso y de manera ordenada el desarrollo de la aplicación, con esto llegar al objetivo que es la construcción del programa.

La Metodología Orientada a Objetos nos ayudará a que la programación del sistema sea más fácil, debido a que sus características se acercan mucho a la realidad.

Al realizar un programa que trabajará bajo Windows y por conocimiento, se usará Microsoft Solutions Framework (MSF) para cubrir todos los requerimientos del software y realizarlo de la mejor manera posible.

El modelado del sistema se realizará con Lenguaje Unificado de Modelado (UML siglas en inglés) la cual es una herramienta fuerte al momento de diseñar y se acopla perfectamente con la programación Orientada a objetos,

obteniendo así todas las vistas del sistema para un buen desarrollo del programa.

Se añadirá conocimientos de Inteligencia artificial para fortalecer la seguridad en los programas, por medio de la **Red Neuronal** con aprendizaje **Hopfield**, la cual nos ayudará a que la información ingresada en la red sea cifrada, adquiriendo con esto dificultad al momento de querer descifrar dicha información.

2.2 Marco Conceptual

2.2.1 Ingeniería de Software

La Ingeniería de Software, no es más que un conjunto de pasos realizados ordenadamente, para el cumplimiento de un objetivo, como desarrollar una aplicación; utilizando métodos, técnicas y herramientas.

La Ingeniería de Software, consta de las siguientes etapas:

Análisis de requisitos

Se obtienen los requisitos del proyecto a realizarse con la participación del cliente y/o las personas que van a utilizar el sistema.

Especificación

Describe el comportamiento esperado en el software una vez desarrollado, El éxito de un proyecto de software radicará en la identificación de las necesidades del negocio (definidas por la alta dirección), así como la interacción con los usuarios funcionales para la recolección, clasificación, identificación, priorización y especificación de los requerimientos del software.

Arquitectura

Consiste en el diseño de componentes de una aplicación (entidades del negocio), generalmente utilizando patrones de arquitectura. El diseño arquitectónico debe permitir visualizar la interacción entre las entidades del

negocio y además poder ser validado, por ejemplo por medio de diagramas de secuencia.

Un diseño arquitectónico describe en general el cómo se construirá una aplicación de software. Para ello se documenta utilizando diagramas, por ejemplo:

Diagramas de clases

Diagramas de base de datos

Diagramas de despliegue

Diagramas de secuencia

Diagramas de infraestructura física

Programación

Reducir un diseño a código puede ser la parte más obvia del trabajo de ingeniería de software, pero no necesariamente es la que demanda mayor trabajo y ni la más complicada. La complejidad y la duración de esta etapa está íntimamente relacionada al o a los lenguajes de programación utilizados, así como al diseño previamente realizado.

Prueba

Consiste en asegurarse que el software realice correctamente las tareas indicadas en la especificación del problema, utilizando varias técnicas, como probar por separado cada módulo del software.

Documentación

Toda la documentación propia del desarrollo de software y de la gestión del proyecto, pasando por modelaciones (UML), diagramas, pruebas, manuales de usuario, manuales técnicos, con el propósito de eventuales correcciones, usabilidad, mantenimiento futuro y ampliaciones al sistema.

Mantenimiento

Mantener y mejorar el software para enfrentar errores descubiertos o nuevos requisitos. La mayor parte consiste en extender el sistema para hacer nuevas cosas.

2.2.2 Metodología Orientada a Objetos

Conjunto de objetos que colaboran entre ellos o interactúan entre sí, al realizar tareas que permitan escribir, mantener y reutilizar. Es decir, la programación Orientada a Objetos, es una forma especial de programar, más cercana a como expresaríamos las cosas en la vida real.

Se utilizará las características más importantes de la Orientación a objetos, como es: abstracción, encapsulación, modularidad.

2.2.3 Microsoft Solutions Framework (MSF)

MSF, propone una secuencia generalizada de actividades para la construcción de soluciones empresariales. Este proceso es flexible y se puede adaptar al diseño y desarrollo de proyectos de cualquier empresa.

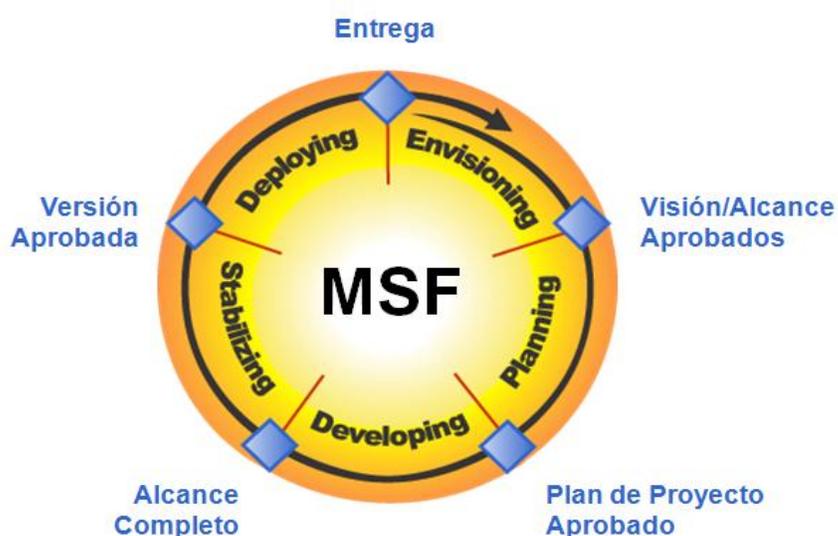


Ilustración 2. MSF (Microsoft Solutions Framework v3.1)

Consiste en una combinación de un modelo interactivo incremental que en la actualidad genera un mayor rendimiento, desarrollo y resultados óptimos; combina los mejores principios del modelo en cascada y del modelo en espiral. Consta de 5 fases, las cuales se detallan a continuación:

Visión

Sirve para dar una visión general del proyecto, en donde se identifica las tareas y los entregables que permitirán al grupo de trabajo cumplir con los objetivos del proyecto.

Planificación

En esta fase se realiza la preparación de la especificación funcional, diseño de la solución, planes de trabajo, costos estimados y calendarios para los entregables. Implica la recogida y el análisis de los requerimientos de negocio, de usuario, operacionales y de sistema.

Desarrollo

La meta de la fase de desarrollo es la construcción de los elementos y entregables de la solución, incluidos los códigos de los componentes, infraestructura (software, hardware, red) y la documentación para el uso de las operaciones.

Estabilización

La solución se pasa un entorno real de exploración, determinar que la solución cumple los criterios de calidad necesarios para pasar a la fase de despliegue.

Reproducción de condiciones reales y el equipo se concentra en detectar y priorizar errores, preparando la solución para su despliegue.

Implantación

Es la fase final, en la cual se instalan los componentes, se estabiliza el proyecto y se obtiene la aprobación por parte del cliente.

2.2.4 Unified Modeling Language (UML)

Lenguaje gráfico para construir, documentar, visualizar y especificar un sistema a través de diagramas.

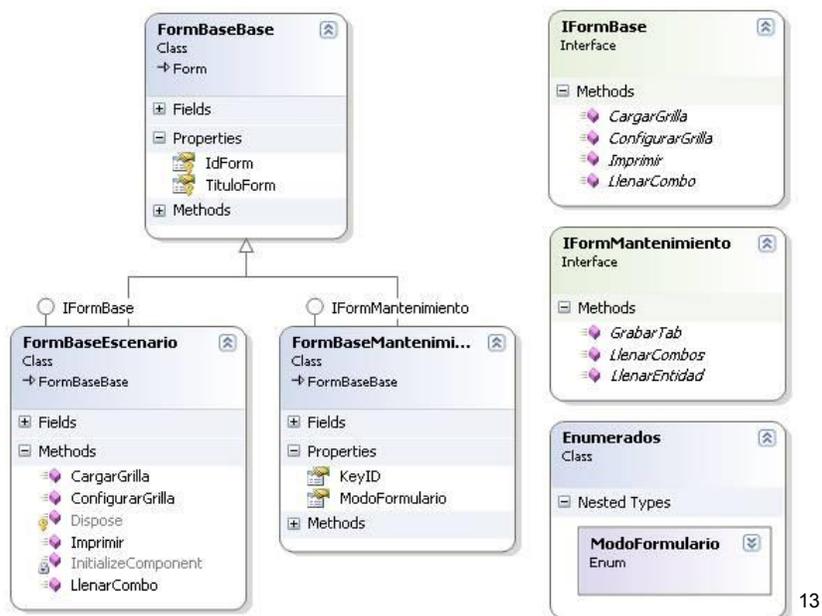
Existen varios diagramas que se dividen en tres grupos:

Diagramas de Estructura

Detalla los elementos que deben existir en el proyecto a realizarse y son los siguientes:

Diagrama de clases

Es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos.



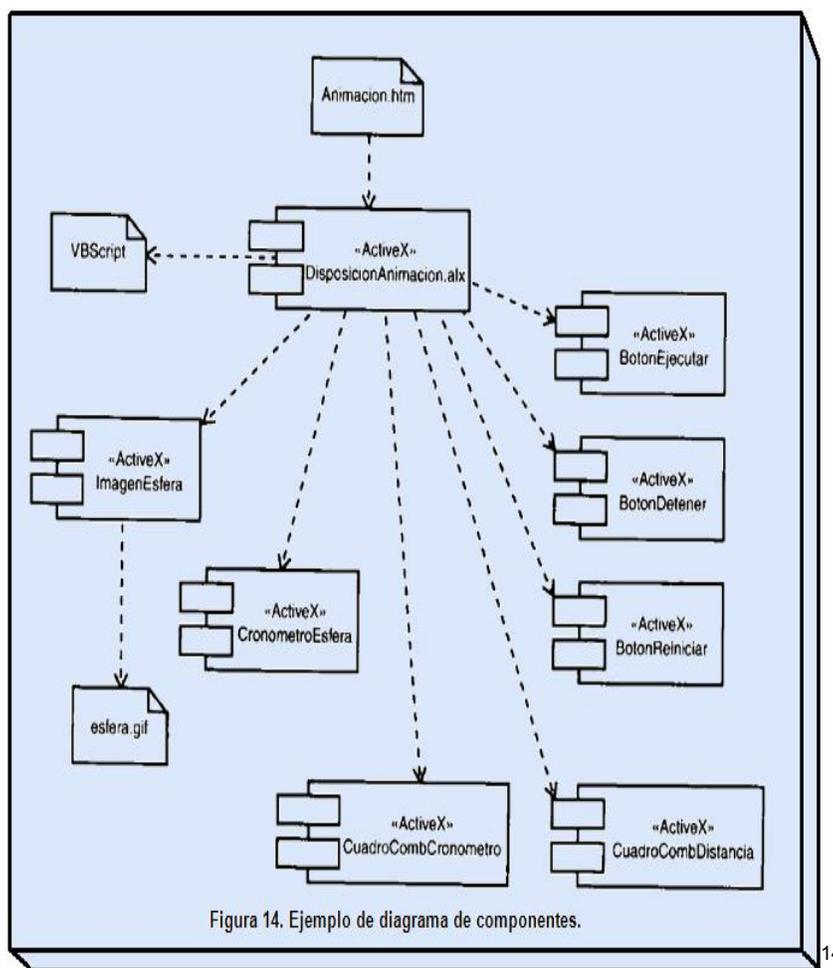
13

Ilustración 3. Ejemplo Diagrama de clases

¹³ <http://www.3devnet.com/images/image002.jpg>

Diagrama de componentes

Un diagrama de componentes representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre estos componentes. Los componentes físicos incluyen archivos, cabeceras, librerías compartidas, módulos, ejecutables, o paquetes.



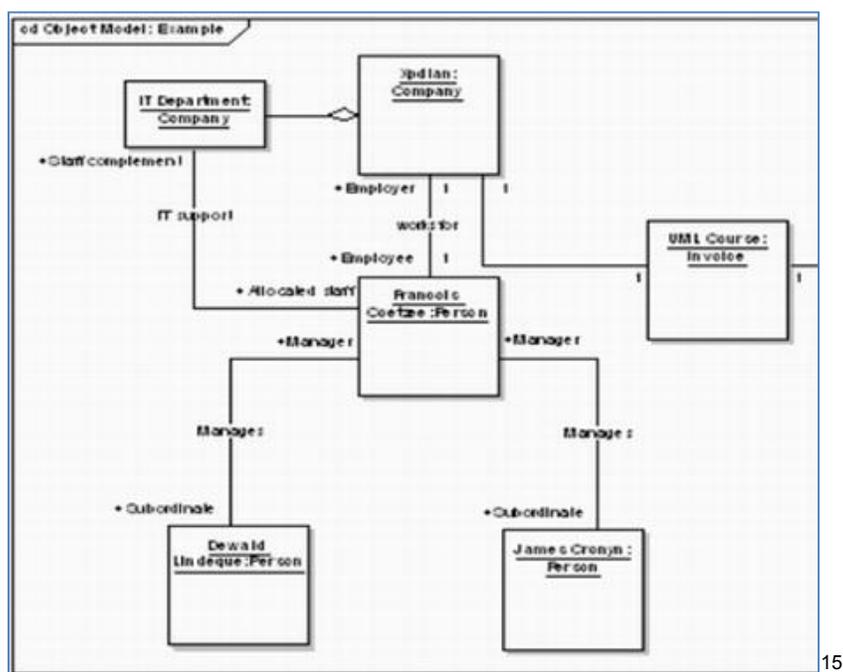
14

Ilustración 4. Ejemplo Diagrama de Componentes

¹⁴ http://jaimemontoya.com/informaticprojects/uml_archivos/image014.png

Diagrama de objetos

Se puede considerar un caso especial de un diagrama de clases en el que se muestran instancias específicas de clases (objetos) en un momento particular del sistema. Los diagramas de objetos utilizan un subconjunto de los elementos de un diagrama de clase. Los diagramas de objetos no muestran la multiplicidad ni los roles, aunque su notación es similar a los diagramas de clase.



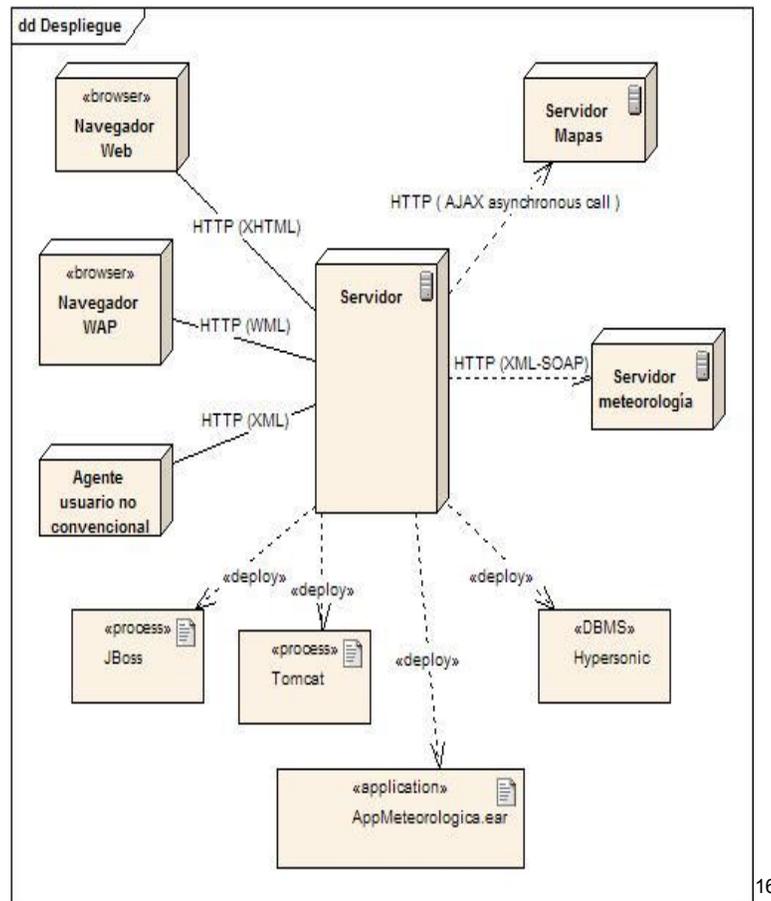
15

Ilustración 5. Ejemplo Diagrama de Objetos

¹⁵ <http://www.monografias.com/trabajos34/ingenieria-software/Image595.gif>

Diagrama de despliegue

Se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes.



16

Ilustración 6. Ejemplo Diagrama de Despliegue

¹⁶ <http://petra.euitio.uniovi.es/~i1650878/diagramas/dv01.JPG>

Diagrama de paquetes

Muestra como un sistema está dividido en agrupaciones lógicas mostrando las dependencias entre esas agrupaciones. Dado que normalmente un paquete está pensado como un directorio, los diagramas de paquetes suministran una descomposición de la jerarquía lógica de un sistema.

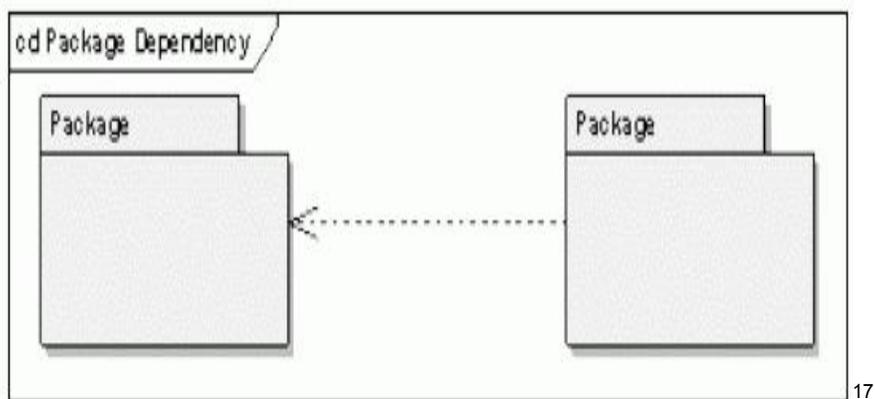


Ilustración 7. Ejemplo Diagrama de Paquetes

¹⁷ <http://www.monografias.com/trabajos34/ingenieria-software/ing7.gif>

Diagramas de Comportamiento

Destacan lo que debe suceder en el proyecto a desarrollarse, y se detallan a continuación:

Diagrama de actividades

Representa los flujos de trabajo paso a paso de negocio y operacionales de los componentes en un sistema. Un Diagrama de Actividades muestra el flujo de control general.

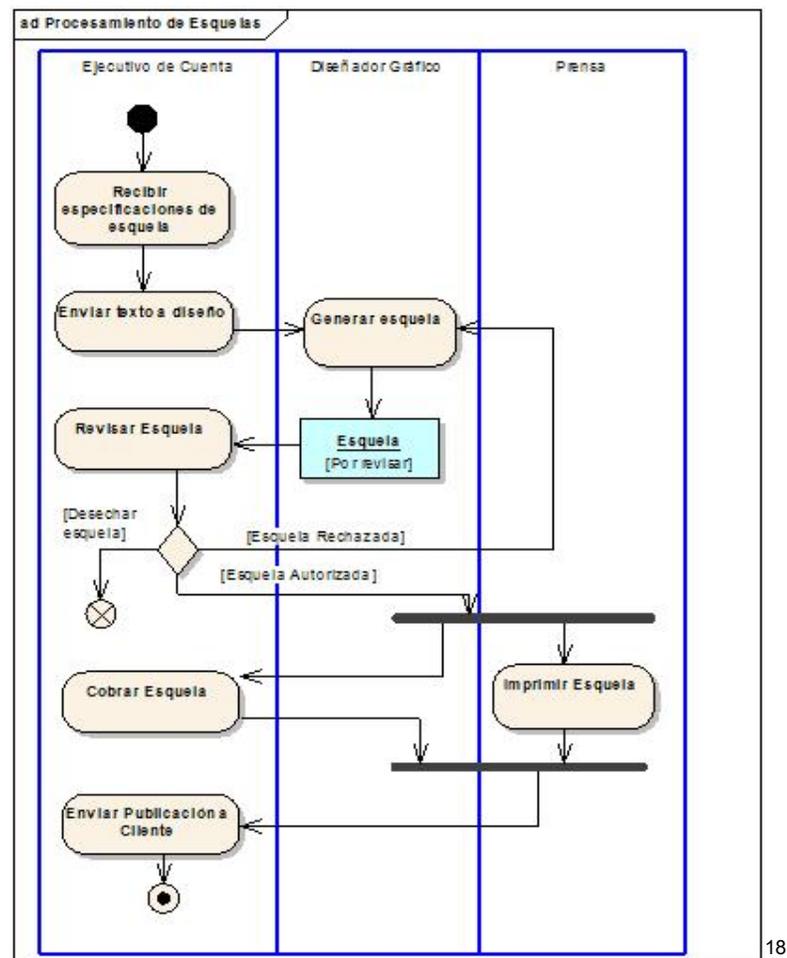
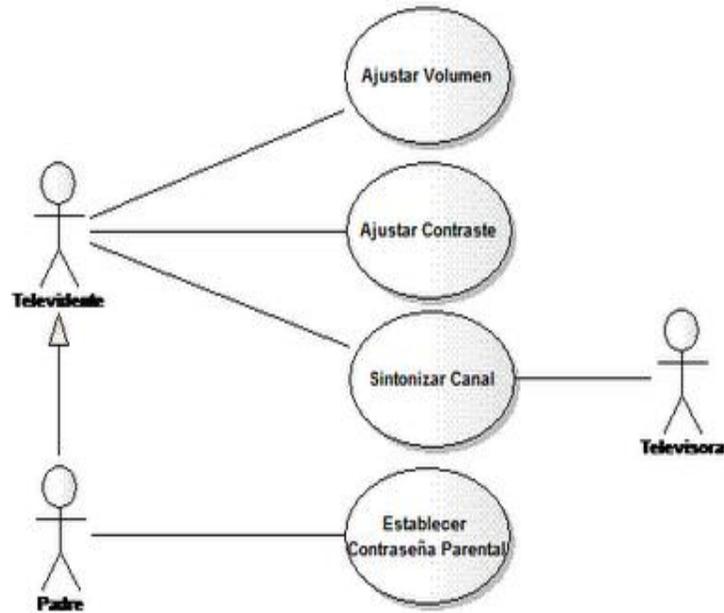


Ilustración 8. Ejemplo Diagrama de Actividades

¹⁸ <http://www.milestone.com.mx/articulos/imagenes/012.gif>

Diagrama de casos de uso

Es una especie de diagrama de comportamiento donde describe las transacciones que realiza un sistema, como resultado de alguna acción que ejecutó un actor.



19

Ilustración 9. Ejemplo Diagrama de Casos de Uso

¹⁹ http://4.bp.blogspot.com/_0QVJ1LDDLZg/RxiR8KaVqtl/AAAAAAAAA0/et5g3epgQ1U/s1600-h/CU+Televisor.gif

Diagrama de estados

Se usan para representar gráficamente máquinas de estados finitos

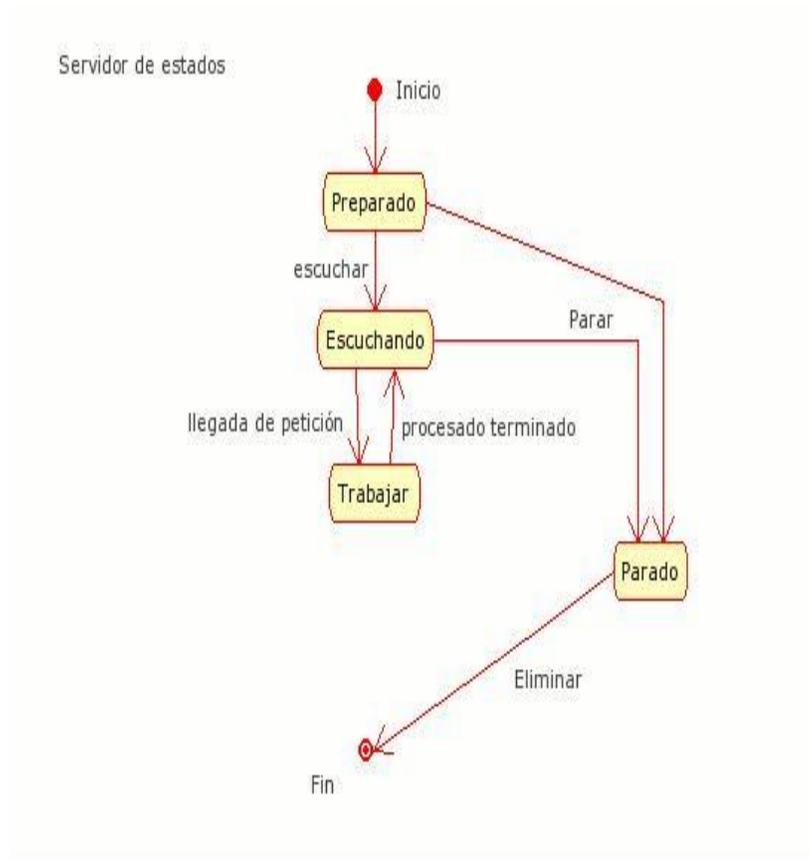


Ilustración 10. Ejemplo Diagrama de Estados

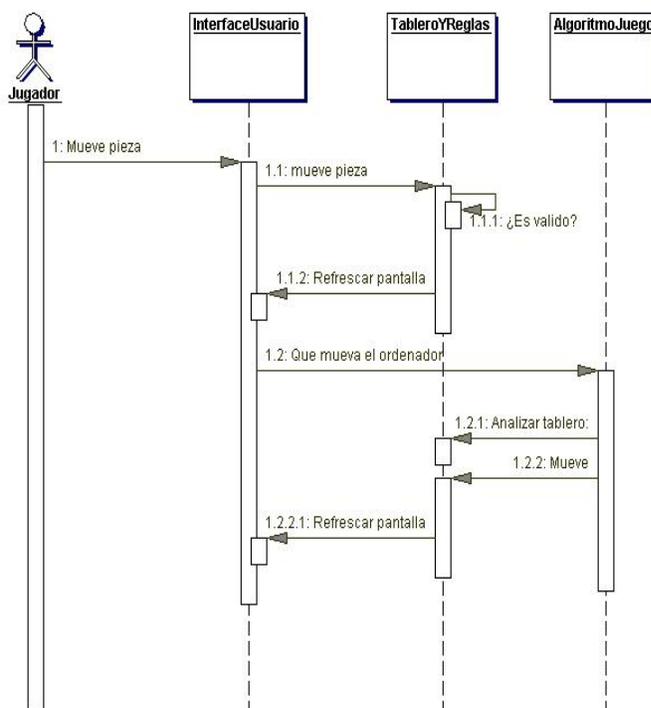
²⁰ <http://angy29.diinoweb.com/i/Que%20es%20un%20Diagrama%20de%20Estado/ejemplos.JPG>

Diagramas de Interacción

Son un subtipo de diagramas de comportamiento, que enfatiza sobre el flujo de control y de datos entre los elementos del sistema modelado:

Diagrama de secuencia

Sirve para modelar la interacción entre objetos en un sistema. Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada método de la clase.



21

Ilustración 11. Ejemplo Diagrama de Secuencia

²¹ <http://www.chuidiang.com/ood/metodologia/Secuencia.gif>

Diagrama de colaboración

Es un diagrama que muestra interacciones organizadas alrededor de los roles. A diferencia de los diagramas de secuencia, los diagramas de comunicación muestran explícitamente las relaciones de los roles.

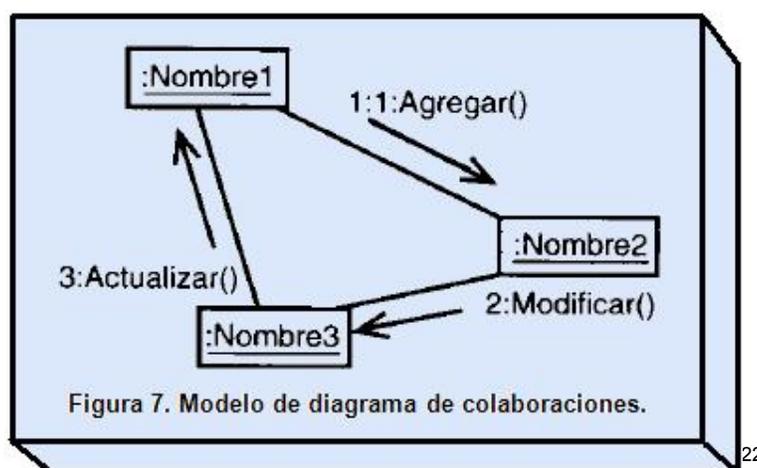


Ilustración 12. Ejemplo Diagrama de Colaboración

2.2.5 Inteligencia Artificial

Es la disciplina que se encarga de construir procesos que al ser ejecutados sobre una arquitectura física producen acciones o resultados que maximizan una medida de rendimiento determinada, basándose en la secuencia de entradas percibidas y en el conocimiento almacenado en tal arquitectura.

Existen distintos tipos de conocimiento y medios de representación del conocimiento. El cual puede ser cargado en el agente por su diseñador o

²² http://jaimemontoya.com/informaticprojects/uml_archivos/image007.png

puede ser aprendido por el mismo agente utilizando técnicas de aprendizaje.

También se distinguen varios tipos de procesos válidos para obtener resultados racionales, que determinan el tipo de agente inteligente. De más simples a más complejos, los cinco principales tipos de procesos son:

Ejecución de una respuesta predeterminada por cada entrada.

Búsqueda del estado requerido en el conjunto de los estados producidos por las acciones posibles.

Algoritmos genéticos.

Redes neuronales artificiales.

Razonamiento mediante una Lógica formal.

2.2.6 Redes Neuronales Artificiales (RNA)

El cerebro humano es uno de los órganos más increíbles que existe, utilizado como modelo para simulaciones desde hace mucho tiempo, siendo el órgano responsable de muchas funciones, tanto básicas como complejas, convirtiéndose la emulación del cerebro en una tarea muy difícil, para lo cual nació una Área de Investigación muy importante como es la Inteligencia Artificial, derivándose de ella una RNA.

Las Redes Neuronales Artificiales (ANNs de Artificial Neural Networks) fueron originalmente una simulación abstracta de los sistemas nerviosos biológicos, formados por un conjunto de unidades llamadas "neuronas" o

"nodos" conectadas unas con otras. Estas conexiones tienen una gran semejanza con las dendritas y los axones en los sistemas nerviosos biológicos.²³

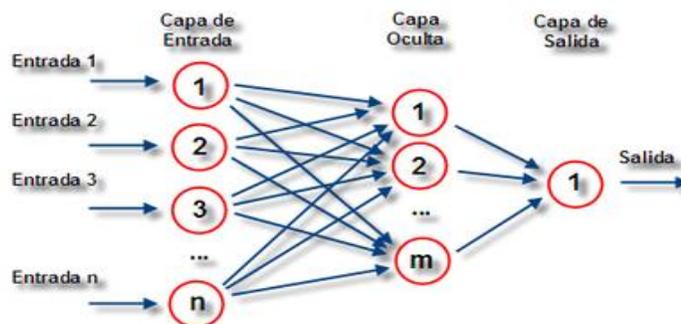


Ilustración 13. Red Neuronal Perceptrón

Red neuronal artificial perceptrón simple con n neuronas de entrada, m neuronas en su capa oculta y una neurona de salida.²⁴

2.2.7 Perceptrón

Éste algoritmo usa una matriz para simbolizar las redes neuronales, a través de ella traza un vector binario x como entrada a un único valor binario $f(x)$ de salida:

$$f(x) = \begin{cases} 1 & \text{if } w \cdot x - u > 0 \\ 0 & \text{else} \end{cases} \quad 25$$

Ilustración 14. Función Red Neuronal Perceptrón

w es un vector de pesos reales y $w \cdot x$ es el producto punto, u es el umbral que da el grado de inhibición de la neurona.

²³ <http://www.gc.ssr.upm.es/inves/neural/ann2/concepts/concepts.htm>

²⁴ <http://upload.wikimedia.org/wikipedia/commons/thumb/6/64/RedNeuronalArtificial.png/400px-RedNeuronalArtificial.png>

²⁵ <http://upload.wikimedia.org/math/1/f/0/1f0cdf8e3be7fa873469d9b64d9df860.png>

2.2.8 Hopfield

Este método de aprendizaje consiste en una red monocapa, que en su salida se obtienen números binarios 0/1 ó -1/+1, sus neuronas están conectadas entre sí (conexiones laterales) pero no entre sí mismas (no autorrecurrentes), la función de activación es tipo escalón:

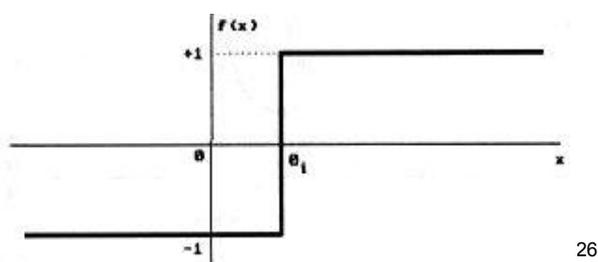


Ilustración 15. Función Tipo Escalón Red Neuronal Hopfield

$$f(x) = +1 \text{ si } x > 0; \quad f(x) = -1 \text{ si } x < 0$$

La red neuronal a usarse en el proyecto es la siguiente:

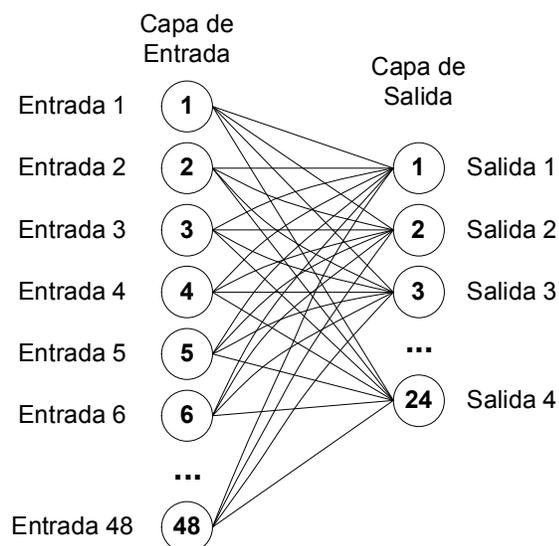


Ilustración 16. Red Neuronal de Hopfield Aplicada al Proyecto

²⁶ <http://thales.cica.es/rd/Recursos/rd98/TecInfo/07/5-2.jpg>

Al escoger los seriales de las partes principales del computador, se generará un número hexadecimal de doce dígitos (serial), el cual será transformado a notación binaria, la red neuronal cifrará estos números, dejando en la salida el código de activación.

2.2.9 BackPropagation

Propagación hacia atrás de errores, algoritmo de aprendizaje supervisado, que entrena una RNA. Radica en minimizar un error por medio de gradiente descendiente, por lo que la parte esencial es el cálculo de las derivadas parciales de dicho error con respecto a los parámetros de la RNA.

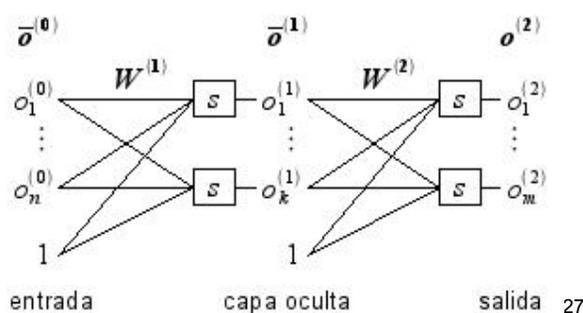


Ilustración 17. Red Neuronal BackPropagation

2.2.10 Seguridad Informática

Protección de Información y de Sistemas de Información, ya sea para evitar su acceso, uso, divulgación, interrupción o destrucción.

La seguridad informática consiste en asegurar que los recursos del sistema de información (material informático o programas) de una organización sean utilizados de la manera que se decidió y que el acceso a la

²⁷ <http://es.wikipedia.org/wiki/Archivo:RedNeuronal.png>

información allí contenida así como su modificación sólo sea posible a las personas que se encuentren acreditadas y dentro de los límites de su autorización.

Activo: Los activos son los elementos que la seguridad informática tiene como objetivo proteger.

Amenaza: es un evento que puede desencadenar un incidente en la organización, produciendo daños materiales o pérdidas inmateriales en sus activos.

Impacto: medir la consecuencia al materializarse una amenaza.

Riesgo: posibilidad de que se produzca un impacto determinado en un Activo, en un Dominio o en toda la Organización.

Vulnerabilidad: posibilidad de ocurrencia de la materialización de una amenaza sobre un Activo.

Ataque: evento, exitoso o no, que atenta sobre el buen funcionamiento del sistema.

2.3 Marco Legal

De acuerdo a las disposiciones legales de carácter general que se rigen los estudios universitarios de la Universidad Tecnológica Israel, revisando las leyes y reglamentos de la República del Ecuador, como la presente Ley de Comercio Electrónico y Ley de Propiedad Intelectual, concluimos que este trabajo no afecta en ninguna manera al marco legal en nuestro país.

La ley de Propiedad Intelectual, menciona de acuerdo al ~~%~~Art.5, *Se reconocen y se garantizan los derechos de los autores y los derechos de los demás titulares sobre sus obras. El derecho del autor nace y se protege por el sólo hecho de la creación de la obra, independientemente de su mérito, destino o modo de expresión.*

Se protege todas las obras, interpretaciones, ejecuciones, producciones o emisión radiofónica cualquiera sea el país de origen de la obra, la nacionalidad o el domicilio del autor o titular.

Esta protección también se reconoce cualquiera que sea el lugar de publicación o divulgación²⁸

~~%~~Art. 28, *Los programas de ordenador se consideran obras literarias y se protegen como tales. Dicha protección se otorga independientemente de que hayan sido incorporados en un ordenador y cualquiera sea la forma en que estén expresados, ya sea de una forma legible por el hombre (código fuente) o en forma legible por máquina (código objeto), ya sea por programas operativos y programas aplicativos, incluyendo diagramas de flujo, planos, manuales de*

²⁸ Ley de Propiedad Intelectual Libro I, Título I, Capítulo I, Sección I, Art 5.

*uso, y en general, aquellos elementos que conformen la estructura, secuencial y organización del programa.*²⁹

La apropiación, divulgación, publicación o copia de esta obra sin debido consentimiento o autorización previa del autor intelecto, estará sujeta a las Leyes de la República del Ecuador y de la Universidad Tecnológica Israel.

2.4 Marco Témporo / Espacial

La investigación y desarrollo de este proyecto será en Quito – Ecuador, destinado para la empresa ECUAKARAOKE y tendrá aproximadamente la duración de seis meses correspondientes al año 2009.

²⁹ Ley de propiedad Intelectual Libro I, Título I, Capítulo I, Sección V, Art.28

3. PROCESO METODOLÓGICO

3.1 Metodología de Investigación

3.1.1 Unidad de Análisis

Para la realización de este proyecto, se ha tomado como unidad de análisis a ECUAKARAOKE, empresa desarrolladora y comercializadora de un programa Karaoke, destinado al entretenimiento de los usuarios por medio de canciones para cantar.

3.1.2 Tipo de investigación

Para el desarrollo de este proyecto, se usará la investigación aplicada, que no es más que la puesta en marcha de los conocimientos adquiridos tanto a nivel académico como los obtenidos por experiencia laboral.

Uniendo la experiencia y lo aprendido, aplicando las teorías existentes, se llegará al objetivo planteado, obteniendo el producto que la organización necesita.

3.1.3 Métodos de investigación

Partiendo del Marco teórico, se utilizará el método deductivo, para llegar a la solución de las falencias que actualmente tiene el método de activación del programa ECUAKARAOKE.

3.1.4 Técnicas y herramientas de investigación

Para cumplir los requerimientos de ECUAKARAOKE, se realizarán entrevistas directas, con el personal de la empresa, con esto obtener un acercamiento a la problemática que atraviesa la organización y brindar la solución correspondiente a las necesidades.

3.2 Metodología Informática

3.2.1 Programación Orientada a Objetos

Para el desarrollo de este proyecto, se va a aplicar algunas de las propiedades más importantes que ofrece la Programación Orientada a Objetos, las cuales se explican a continuación:

Abstracción

Se utiliza para describir las características más importantes de un objeto y convertirla en una clase que nos permitirá utilizar dicha descripción en un programa.

En éste proyecto, se determinará las características principales de un computador como dirección MAC, Id del procesador, Id de la Placa Madre y el Id del Disco Duro, llevándola a una clase con sus respectivos atributos y métodos que servirán para la generación de seriales y códigos de activación.

Para el prototipo de control de seriales y códigos, se abstraerá lo más importante de una persona, por ejemplo: cédula, nombre, dirección, teléfono, país, provincia, cantón, fecha de nacimiento, correo.

Encapsulamiento

El encapsulamiento permite proteger el contenido de un objeto del exterior, como sus métodos y atributos, logrando con esto controlar el acceso a los mismos. Las clases para el módulo de activación estarán encapsuladas con el fin de permitir su uso y proteger los detalles internos y más importantes de la misma, de igual forma, para el prototipo de control de códigos, se aplicará la encapsulación declarando los atributos y métodos más importantes como privados.

Modularidad

Esta característica, se usa para descomponer un programa en partes más pequeñas llamadas módulos, ayudando con esto a que el desarrollo de un programa sea más sencillo.

En el proyecto, se descompondrá las partes más esenciales para la generación de códigos, obteniendo diferentes módulos que cumplirán el objetivo del programa, entre ellos, el módulo para generar los seriales y códigos como también el módulo para control de los mismos.

3.2.2 Microsoft Solutions Framework

Visión

Se realizará entrevistas con el personal de ECUAKARAOKE para obtener la información necesaria que ayude al desarrollo del proyecto, así como también la obtención de todos los problemas que tiene hoy en día la empresa, para esto, se realizará los siguientes pasos:

Identificar el problema

Obtener los requerimientos iniciales

Definir las metas y objetivos del proyecto

Especificar los Roles y Equipos de trabajo

Definir Alcance

Identificar Riesgos

Modelo de la Solución

Planificación

Se realizará un análisis en la empresa ECUAKARAOKE, donde se construirá la arquitectura y diseño del proyecto, donde se obtendrá las especificaciones funcionales y el cronograma del proyecto. Esta fase, tendrá los siguientes hitos:

Diseño Conceptual

Diseño Lógico

Desarrollo

En esta fase, se elaborará todos los diagramas del proyecto, para medir su progreso y verificar que todos los componentes puedan integrarse. Se cumplirá con los siguientes pasos:

Diagramas de Componentes

Diagramas de Despliegue

Diagrama de Base de Datos

Diagrama de Clases

Interfaces

Estabilización

Se realizarán las debidas pruebas en el producto, basándose en un plan, para alcanzar los criterios de aceptación del cliente y la entrega en producción. Se obtendrá un reporte de errores y pruebas.

Implantación

El módulo de activación de códigos se pondrá a prueba en el programa ECUAKARAOKE, como también el prototipo de control de Códigos, para demostrar su uso más no con el fin de implantarlo, llegando por esto al cierre del proyecto.

4. PROCESO DE DESARROLLO

4.1 Microsoft Solutions Framework

4.1.1 Visión

Planteamiento del problema

En la empresa ECUAKARAOKE existe actualmente un método de activación donde existen muchas falencias que permiten que el programa se instale en muchas computadoras sin la debida autorización legal, entre las falencias que existen, hay computadores que no tienen tarjeta de red o en algunos casos las desactivan ya sea desde el BIOS³⁰ o desde el Sistema Operativo, como también la clonación de la dirección MAC, al ocurrir esto, se crea un mismo Serial en varias computadoras, permitiéndose la instalación ilegal del producto.

Requerimientos iniciales

Requerimientos del Negocio

El módulo de activación de códigos propuesto, mejorará la seguridad del programa ECUAKARAOKE, usando el algoritmo de aprendizaje de Hopfield, junto con los nuevos periféricos del computador utilizados, se dará un mejor desempeño y un valor agregado a la empresa, controlando las copias ilegales del programa.

Con el desarrollo del prototipo de control de códigos y seriales, se obtendrá un mejor desempeño en las entregas de licencias a los clientes.

Requerimientos del usuario

³⁰ BIOS: (Basic Input-Output System) Sistema Básico de Entrada/Salida, software muy básico instalado en la placa base que permite que ésta cumpla su cometido.

El módulo de activación de códigos debe crear a través de periféricos del computador un código y un serial para la activación.

El código y el serial deben ser únicos para cada computador.

Desarrollar un prototipo de control de códigos y seriales con las siguientes características:

Manejar un perfil de usuarios para el ingreso, con nombre y contraseña.

Una pantalla principal con el logo de la empresa, que contenga el acceso a Clientes / Distribuidores y Salida, además que despliegue el nombre del usuario que está usando el programa.

En Clientes / Distribuidores una pantalla que contenga la información de un cliente (Cédula, Nombre, Dirección, Correo, Teléfonos, Fecha de nacimiento, País, Provincia, Cantón, Negocios, Vendedor); que la búsqueda se realice por número de cédula o por nombre.

Si no existe el registro buscado, crearlo y guardar con los datos mencionados (debe permitirse ingresar varios teléfonos y varios negocios con diferentes pantallas).

La pantalla de negocio debe tener Nombre del negocio, Dirección, Encargado, País, Provincia y Cantón.

Al existir el registro buscado, presentar los datos del cliente y además mostrar las opciones de crear códigos e historiales de los ya creados.

La pantalla de crear códigos tendrá que Versión se va a usar, el número de Actualización, a que negocio, el código y serial creado, y un campo para observaciones.

En la pantalla de historiales, que se pueda escoger de que negocio necesita el historial de códigos, desplegando la Versión, Actualización, Serial, Código, el usuario que lo creó, fecha y observación.

Requerimientos Operacionales

El módulo de activación deberá generar un código y serial único, para el respectivo registro del cliente.

El prototipo de control de códigos y seriales, deberá registrar los datos de cada cliente para una correcta administración de cartera, como también generar un historial de los códigos y seriales creados.

Requerimientos del Sistema

El módulo de activación de códigos y seriales, como también el prototipo de control, está orientado para el uso en cada computador, por lo que no se manejará en un ambiente distribuido, pero desarrollado con una arquitectura por capas.

Será desarrollado en ambiente Windows porque el programa ECUAKARAOKE trabaja en esa plataforma.

El prototipo de control de códigos y seriales tendrá una información confiable y segura al momento de realizar actualizaciones.

Metas y Objetivos del proyecto

Metas

Mejorar el proceso de activación actual de ECUAKARAOKE, que permitirá la generación de códigos y seriales.

Facilitar las actividades administrativas con un control automatizado de los clientes, códigos y seriales, para prestar mejor servicio a los usuarios de ECUAKARAOKE.

Objetivos

Obtener mayores ventas del producto ECUAKARAOKE, aprovechando las nuevas seguridades que se implementará en el módulo de activación del programa.

Mitigar el incremento de copias no autorizadas del programa ECUAKARAOKE, con el nuevo módulo de generación de códigos y seriales.

Suposiciones

Se logrará que el módulo de activación de códigos y seriales sea eficiente para la protección del programa ECUAKARAOKE.

Se minimizará los tiempos de consulta de clientes, códigos y seriales para un mejor desempeño de la empresa.

Se beneficiará ECUAKARAOKE con una mejor atención y servicio a sus clientes al automatizar la información

Restricciones

El módulo de activación de códigos y seriales, como el prototipo de control, se realizará única y exclusivamente para ECUAKARAOKE.

Roles y Equipos de trabajo

Roles

Tabla 10. Roles, Tareas y Entregables en la Fase de Visión

ROL	TAREAS	ENTREGABLES
Gerencia de Producto	Elaborar documento de visión Administrar expectativas del cliente Administrar riesgos	Documento de Visión
Gerencia de Programa	Describir el concepto de la solución Delinear le estructura del proyecto Administrar riesgos	Estructura Proyecto Matriz de Riesgos
Desarrollo	Diseño de prototipos Evaluar opciones de desarrollo Administrar riesgos	Modelo de Casos de Uso de alto nivel Modelo de Clases de alto nivel
Pruebas	Desarrollar estrategias de pruebas Especificar criterios aceptación Diseñar sistema de reporte defectos Diseñar sistema administración riesgos Administrar riesgos	
Educación a usuarios	Identificar necesidades de usuarios e implicaciones Administrar expectativas usuarios	
Logística	Identificar implicaciones de implementación Identificar implicaciones para soporte Administrar riesgos	

Tabla 11. Roles, Tareas y Entregables en la Fase de Planificación

Rol	Tareas	Entregables
Gerencia de Producto	Conducir proceso de relevar requerimientos Conducir proceso de diseño conceptual Plan de comunicación	
Gerencia de Programa	Conducir el diseño, sobre todo el diseño lógico	Especificaciones funcionales
	Elaborar especificaciones funcionales Determinar si se puede cumplir con planes y cronograma	Plan del Proyecto Matriz de Riesgos
Desarrollo	Diseño físico Estimación de tiempo y esfuerzo Desarrollo de pruebas de concepto	Casos de Uso Modelo de Clases Modelos Dinámicos Modelo Lógico y Físico de BDD
Pruebas	Preparar plan de pruebas Preparar métodos y métricas para seguimiento defectos Preparar estrategias de pruebas	
Educación a usuarios	Evaluar usabilidad del diseño Estimar tiempo y esfuerzo para elaboración materiales de soporte	
	Probar usabilidad de interfaz gráfico	
Logística	Evaluar diseño para instalación, administración y soporte	

Tabla 12. Roles, Tareas y Entregables en la Fase de Desarrollo

Rol	Tareas	Entregables
Gerencia de Producto	Administrar expectativas del cliente Elaborar plan de comunicación Preparar clientes para pruebas alfa y beta	
Gerencia de Programa	Coordinación Revisar especificaciones funcionales, plan y cronograma del proyecto y matriz de riesgos	Especificaciones funcionales Plan del Proyecto Matriz de Riesgos
Desarrollo	Crear código Pruebas iniciales funcionalidad	Código fuente y ejecutables
Pruebas	Crear plan de pruebas y casos de prueba Realizar pruebas Identificar defectos	Plan de pruebas y casos de prueba
Educación a usuarios	Pruebas usabilidad y rendimiento usuarios Coordinar realización pruebas alfa y beta Crear materiales iniciales de soporte	Materiales de soporte a usuarios (manuales, ayudas, cursos, etc.)
Logística	Soporte interno en IT Soporte a pruebas alfa y beta Crear materiales de soporte para instalación y administración	

Tabla 13. Roles, Tareas y Entregables en la Fase de Estabilización

Rol	Tareas	Entregables
Gerencia de Producto	Coordinar releases interinos con el cliente Planear lanzamiento producto	
Gerencia de Programa	Administrar pruebas beta y pruebas piloto Mantener el cronograma Coordinar la recepción por el cliente y por los grupos de operaciones y soporte	Especific. funcionales Plan del Proyecto Matriz de Riesgos Revisión de hitos
Desarrollo	Encontrar, reportar y arreglar defectos Asegurar y probar integración de componentes	Release Gold Notas del release Código fuente y ejecutables
Pruebas	Ejecutar plan de pruebas Encontrar, reportar y clasificar defectos Verificar resolución de defectos Mayor enfoque en pruebas de usabilidad, instalación y configuración	Resultados de las pruebas
Educación a usuarios	Terminar materiales de soporte a usuarios Coordinar capacitación	Materiales de soporte a usuarios
Logística	Instalación, configuración y soporte de releases interinos Entrenamiento a operaciones y soporte	

Tabla 14. Roles, Tareas y Entregables en la Fase de Implantación

Rol	Tareas	Entregables
Gerencia de Producto	Retro alimentación por parte de los usuarios, cierre del proyecto	
Gerencia de Programa	Solución en producción, comparación del alcance con los requerimientos, administración de la estabilización	Sistemas de operación y soporte Procedimientos
Desarrollo	Resolución de problemas, soporte, mantenimiento	Versiones finales del código generado
Pruebas	Ejecución del plan de pruebas, identificación de problemas	Pruebas de aceptación
Educación a usuarios	Entrenamiento (usuarios, técnicos), administración de los cronogramas de entrenamiento	Resultados de los planes de entrenamiento
Logística	Administración de la implantación de la solución, aprobación del producto	Acta de entrega recepción

Equipo de trabajo**Tabla 15. Equipo de Trabajo**

ROL	NOMBRE	TELÉFONO	CORREO
Líder de Proyecto	Ing. Vladimir Bonilla	099926868	ybonilla@yahoo.com
Líder de Programa	Diego Orellana	099496243	droo99@hotmail.com
Desarrollo	Diego Orellana	099496244	droo99@hotmail.com
Pruebas	Ing. Andrés Freire	095583355	kronus47@hotmail.com

Alcance

Características Funcionales

El prototipo creará códigos y seriales para la activación del producto ECUAKARAOKE, éstos serán únicos para cada computador

Se enfocará con perfiles de usuario para el ingreso, con nombre y contraseña.

Manejará el modulo de activación de códigos y seriales, creando el Serial de activación a través del código del computador, empleando Redes Neuronales para la encriptación.

El prototipo tendrá una pantalla principal, que contendrá el acceso a las Interfaces de Cliente / Distribuidor y Salir.

La Interfaz Cliente Distribuidor, tendrá dos pestañas con el acceso a las interfaces de Generar Códigos e Historial de Códigos; esta Interfaz de Cliente también llamará a una Interfaz Nuevo Cliente.

Desde la Interfaz Nuevo Cliente se tendrá el acceso a las interfaces de Nuevo Teléfono y Nuevo Negocio.

La Interfaz Nuevo Cliente, permitirá guardar los datos del Cliente (Cédula, Nombre, Dirección, Correo, Teléfonos, Fecha de nacimiento, País, Provincia, Cantón, Negocios, Vendedor)

En la Interfaz Generar Códigos se podrá generar y guardar el código y el serial de activación de acuerdo a la versión escogida, tipo de Actualización, que negocio y un campo para observaciones.

En la Interfaz de historiales, se escogerá el negocio y desplegará la Versión, Actualización, Serial, Código, el usuario que lo creó, fecha y observación de los códigos existentes.

La Interfaz Nuevo teléfono, tendrá un campo para la descripción y otro para escoger el tipo de teléfono.

En la Interfaz Nuevo Negocio, se ingresará Nombre del negocio, Dirección, Encargado, País, Provincia y Cantón

Fuera del alcance

El prototipo de control de códigos y seriales, solo permitirá guardar los datos del cliente, los códigos y seriales generados y el historial de información; no se permitirá modificar ni eliminar ningún dato.

El módulo de activación y el prototipo de control de códigos y seriales, será creado solo para pruebas con el programa ECUAKARAOKE.

El módulo y el prototipo a pesar de estar diseñados por capas, no trabajarán en ambiente distribuido, está destinado solo para el uso en un computador.

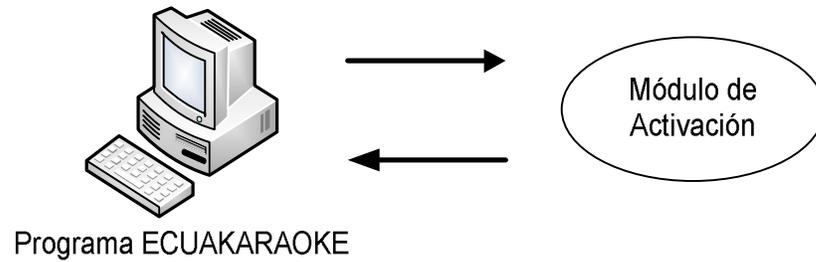
Riesgos

Tabla 16. Matriz de Riesgos

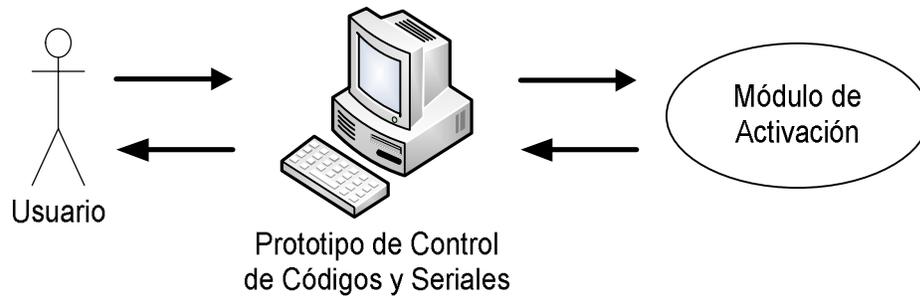
Valoración	Descripción					
1	Medio					
2	Sostenible					
3	Catastrófico					
4	Severo					
		Probabilidad	Impacto	Riesgo		
Condición	Consecuencia	P	I	P * I	Mitigación	
Clonación de dirección MAC	Utilización de ECUAKARAOKE con un mismo serial en varios computadores	30%	4	1,2	Resetear todas las tarjetas de red a su estado de fábrica	
Desactivación de tarjetas de red	Utilización de ECUAKARAOKE con un mismo serial en varios computadores	70%	4	2,8	Activar todas las tarjetas existentes	
Acceso de Usuarios no autorizados al prototipo	Modificación o pérdida de los registros de clientes de ECUAKARAOKE.	20%	3	0,6	Manejar correctamente los perfiles de usuario	
Personal inadecuadamente capacitado	No podrá operar debidamente el sistema	70%	3	2,1	Tener personal capacitado para agilizar los diferentes procesos	
Falla en la disponibilidad de la información	Los usuarios no tendrán acceso a tiempo de la información	30%	2	0,6	Tener la información siempre disponible	
Realizar las pruebas del software de forma incorrecta	El producto no tendrá calidad ni consistencia para salir a producción.	50%	3	1,5	Manejar de forma correcta las pruebas	

Modelo de la Solución

CREACIÓN DE CÓDIGO DEL COMPUTADOR



CREACIÓN DE SERIAL PARA LA ACTIVACIÓN DE ECUAKARAOKE



DISEÑO POR CAPAS



4.1.2 Planeación

Plan de Proyecto

Tabla 17. Plan de Desarrollo

DESCRIPCIÓN	RESPONSABLE
Definir Oportunidades del negocio Describir el concepto de la solución Delinear la estructura del proyecto Conducir el diseño Lógico Elaborar especificaciones funcionales Determinar cumplimiento con el cronograma Coordinar Proyecto Administrar Pruebas Comparar el alcance con los requerimientos Administración de la estabilización	Líder de Proyecto
Definir expectativas del cliente Determinar requerimientos Conducir el diseño conceptual Preparar clientes para pruebas Coordinar realizables con clientes Preparar producto para lanzamiento	Líder del Programa
Diseñar Prototipo Diseñar Modelo Casos de Uso Diseñar Modelo de clases Conducir el diseño físico Crear código Pruebas iniciales de funcionalidad Encontrar, reportar y arreglar defectos Asegurar y probar integración de componentes Resolución de problemas, soporte y mantenimiento	Desarrollo
Preparar plan de pruebas Realizar pruebas Identificar defectos Encontrar, reportar y arreglar defectos Verificar resolución de defectos	Pruebas

Escenarios de Uso

Módulo de activación de Códigos y seriales

- **Inicio programa ECUAKARAOKE**
 - Generar Código del computador
 - Activar programa ECUAKARAOKE

Prototipo de Control de Códigos y Seriales

- **Iniciar Sesión**
 - Verificar Usuario y Contraseña
- **Pantalla Principal**
 - Acceso a Pantalla Cliente
 - Salida del Prototipo
- **Cliente / Distribuidor**
 - Búsqueda de Cliente / Distribuidor
 - Registro de Clientes
 - Salida
- **Generar Seriales**
 - Generar y Guardar Serial de activación
 - Salir
- **Historial de Códigos y Seriales**
 - Escoger Negocio
 - Generar Consulta
 - Salir

Actores del Prototipo

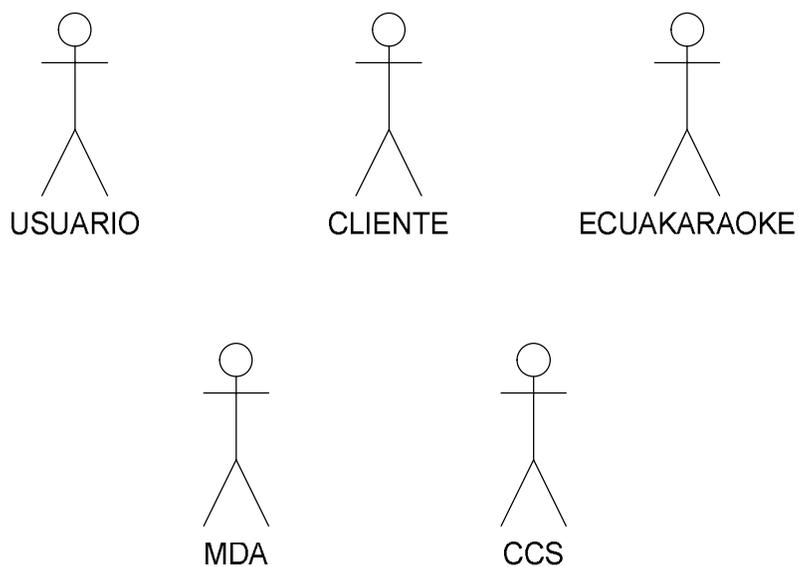


Tabla 18. Actores y Descripción que intervienen en el proyecto

ACTORES	DESCRIPCIÓN
CCS	Prototipo de Control de Seriales y Códigos de activación.
MDA	Módulo para crear Seriales y Códigos de Activación.
USUARIO	Persona encargada del manejo de CCS.
ECUAKARAOKE	Programa con pistas originales para cantar
CLIENTE	Persona que usa el programa ECUAKARAOKE

Casos de Uso

Caso de Uso Ingresar a ECUAKARAOKE

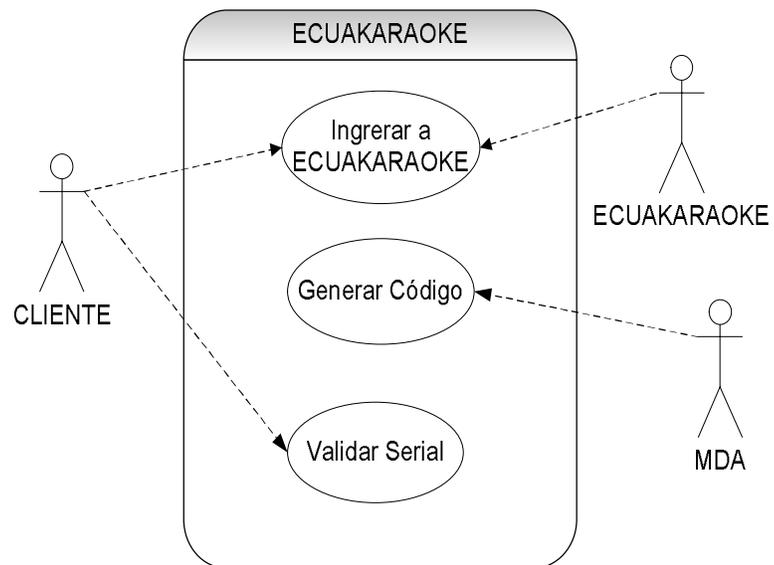


Ilustración 18. Caso de Uso ECUAKARAOKE

Caso de Uso Iniciar Sesión

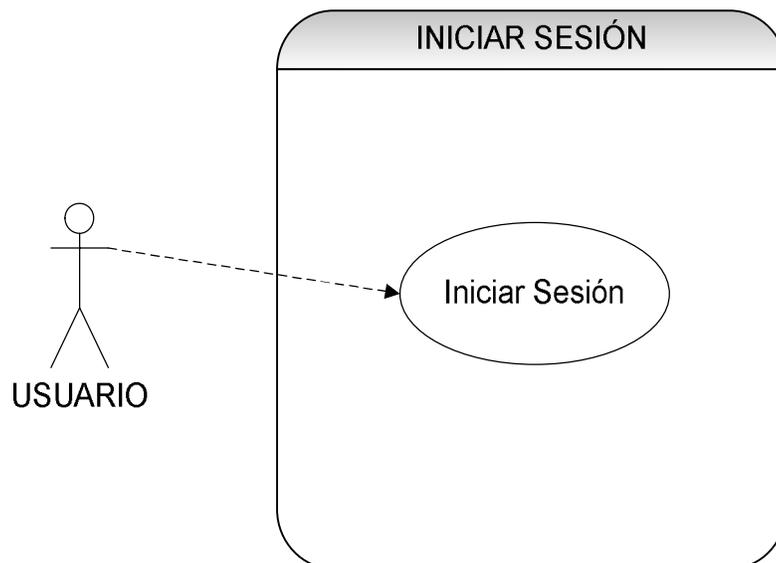


Ilustración 19. Caso de Uso Iniciar Sesión

Caso de Uso Pantalla Principal

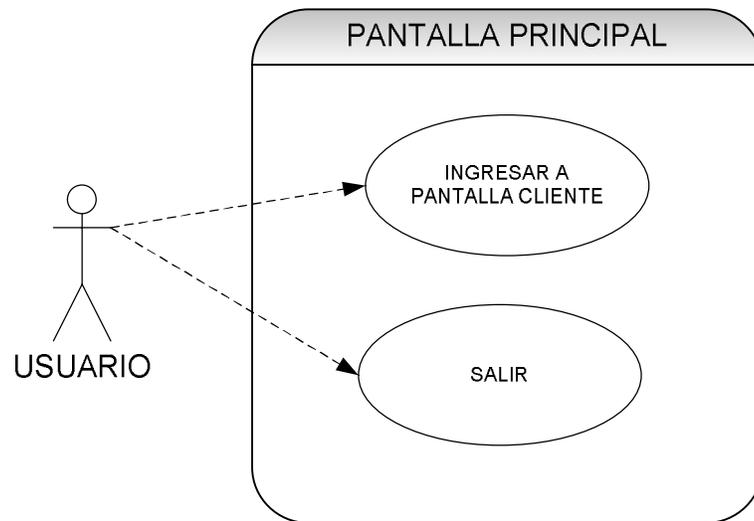


Ilustración 20. Caso de Uso Pantalla Principal

Caso de Uso Pantalla Cliente

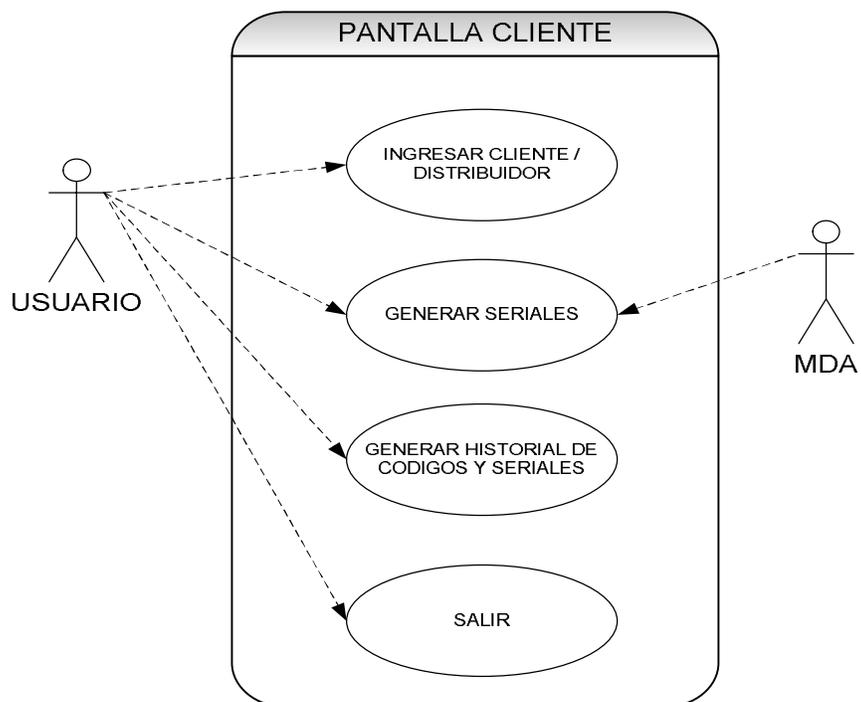


Ilustración 21. Caso de Uso Pantalla Cliente

Diagrama General de Casos de Uso

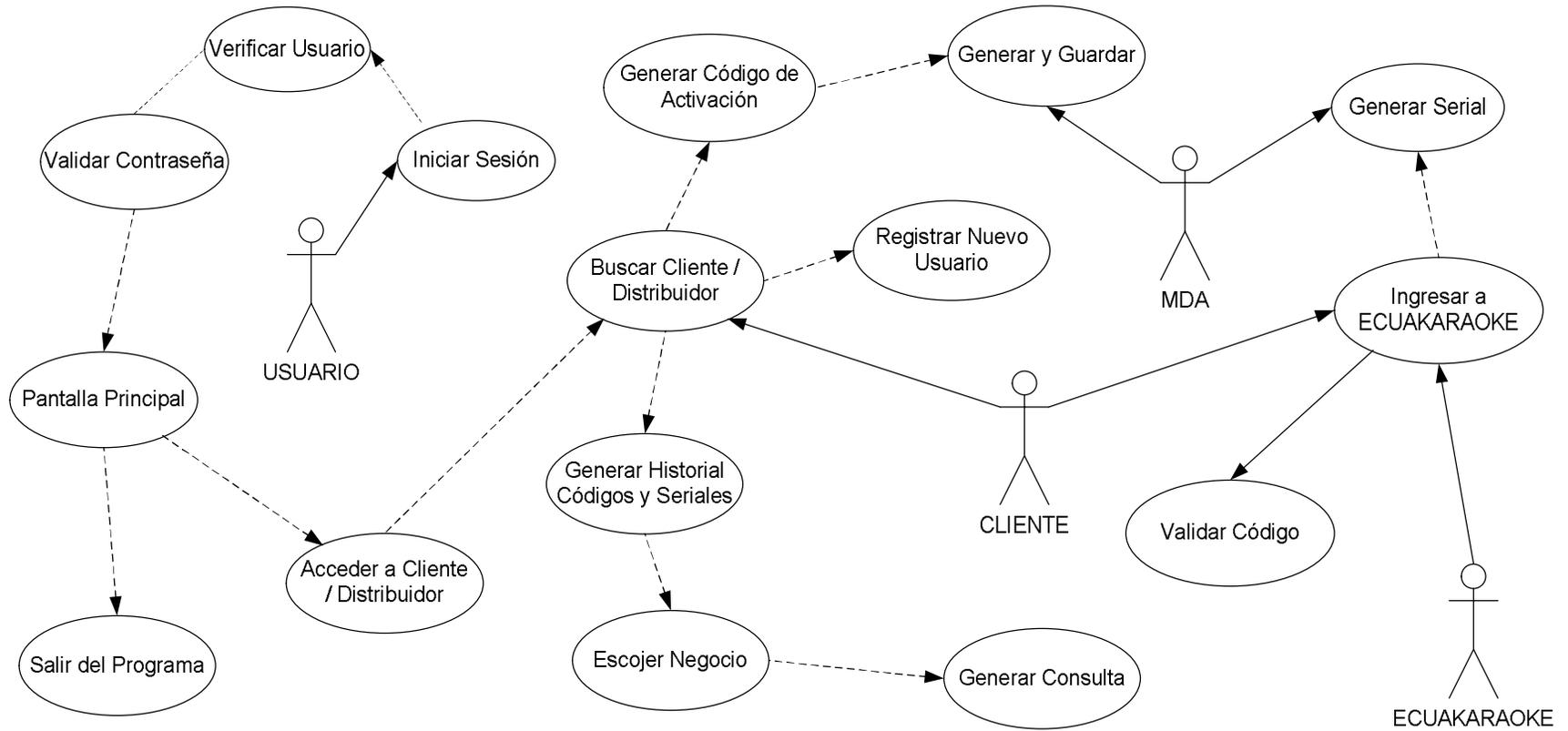


Ilustración 22. Diagrama General de Casos de Uso

Especificaciones de Casos de Uso

Tabla 19. Especificación Caso de Uso Ingresar a ECUAKARAOKE

Nombre del Diagrama de Caso de Uso: Ingresar a Ecuakaraoke	
<pre> graph TD MDA((MDA)) --> GS((Generar Serial)) GS --> IE((Ingresar a ECUAKARAOKE)) ECUAKARAOKE((ECUAKARAOKE)) --> IE IE --> VC((Validar Código)) </pre>	
Autor:	Diego Orellana
Identificador de Escenario	IE1
Descripción:	Describe el ingreso al programa Ecuakaraoke, para luego generar el código del programa.
Actores	Ecuakaraoke, Cliente, MDA,
Interface	InterfaceMDA
Precondición	El cliente debe instalar el programa Ecuakaraoke y ejecutarlo.
Acciones	<ol style="list-style-type: none"> 1. Llamar a la empresa Ecuakaroke. 2. Solicitar serial de activación. 3. Ingresar serial de activación. 4. Validar serial, si es correcto ingresa al programa, caso contrario emitir un mensaje de Número de serie erroneo.
Post- condición	Ninguna
Inclusiones	Generar Serial, Validar Código
Extensiones	Ninguna
Generalizaciones	Ninguno

Tabla 20. Especificación Caso de Uso Iniciar Sesión

Nombre del Diagrama de Caso de Uso: Iniciar Sesión	
<pre> graph TD Validar(Validar Contraseña) --> Verificar(Verificar Usuario) Verificar --> Iniciar(Iniciar Sesión) Usuario((USUARIO)) --> Iniciar </pre>	
Autor:	Diego Orellana
Identificador de Escenario	IS2
Descripción:	Describe el ingreso y validación de cada usuario al programa Control de Códigos y Seriales (CCS)
Actores	Usuario.
Interface	InterfaceSesion, InterfaceUsuario, interfaceBDD
Precondición	
Acciones	<ol style="list-style-type: none"> 1. Seleccionar Usuario e ingresar contraseña 2. Se verificará el usuario y contraseña, si es correcto ingresa al prototipo, si no emitirá un mensaje Clave incorrecta.
Post- condición	
Inclusiones	Verificar Usuario, Validar Contraseña
Extensiones	Ninguno
Generalizaciones	Ninguno

Tabla 21. Especificación Caso de Uso Pantalla Principal

Nombre del Diagrama de Caso de Uso: Pantalla Principal	
<pre> graph TD USUARIO((USUARIO)) --> PantallaPrincipal((Pantalla Principal)) PantallaPrincipal -.-> SalirdelPrograma((Salir del Programa)) PantallaPrincipal -.-> AccederClienteDistribuidor((Acceder a Cliente / Distribuidor)) </pre>	
Autores:	Diego Orellana
Identificador de Escenario	PP3
Descripción:	Contiene el acceso a toda la funcionalidad del prototipo para la creación de seriales y códigos de activación
Actores	Usuario
Interface	InterfaceCli_Dis, InterfaceUsuario
Precondición	Validación de Usuario y Contraseña en el Inicio de Sesión
Acciones	<ol style="list-style-type: none"> 1. Puede acceder a la Interfaz Cliente/Distribuidor 2. Puede salir del programa
Post- condición	
Inclusiones	Ninguna.
Excepciones	Ninguna.
Generalizaciones	Ninguna.

Tabla 22. Especificación Caso de Uso Cliente / Distribuidor

Nombre del Diagrama de Caso de Uso: Cliente/Distribuidor	
<pre> graph TD USUARIO((USUARIO)) --> AC[Acceder a Cliente / Distribuidor] CLIENTE((CLIENTE)) --> BC[Buscar Cliente / Distribuidor] MDA((MDA)) --> RG[Registrar Nuevo Usuario] AC --> BC BC --> GA[Generar Código de Activación] BC --> RNU[Registrar Nuevo Usuario] BC --> GHS[Generar Historial Códigos y Seriales] GA --> GG[Generar y Guardar] GHS --> EN[Escojer Negocio] EN --> GC[Generar Consulta] GG --> MDA </pre>	
Autores:	Diego Orellana
Identificador de Escenario	ACD4
Descripción:	Abarca la gestión de Clientes, Seriales y Códigos de activación
Actores	Usuario, Cliente, MDA
Interface	InterfaceCli_Dis, InterfaceUsuario, interfaceBDD, InterfaceMDA InterfaceCliente, InterfaceRegClient
Precondición	
Acciones	<ol style="list-style-type: none"> 1. Acceder a la pantalla Cliente/distribuidor. 2. Buscar por cedula o nombre al cliente que esté registrado, caso contrario desplegará un mensaje No existe ese registro, ¿Desea crearlo?. 3. Si el cliente está registrado, se presenta toda la información obtenida y se activaran las viñetas Generar Código e Historial de Códigos. 4. En la viñeta Generar Código, se presenta campor de Versión actualización, negocio, código, serial y observaciones 5. Una vez ingresados se procede a Generar y Guardar 6. La viñeta Generar Historial, permite escoger el negocio y posteriormente nos genera la consulta requerida.
Post- condición	
Inclusiones	Escojer Negocio, Generar Consulta, Generar y Guardar
Extensiones	Registrar Nuevo Usuario, Generar Historial Códigos y Seriales Generar Código de activación
Generalizaciones	Ninguno

Diagramas de Secuencia

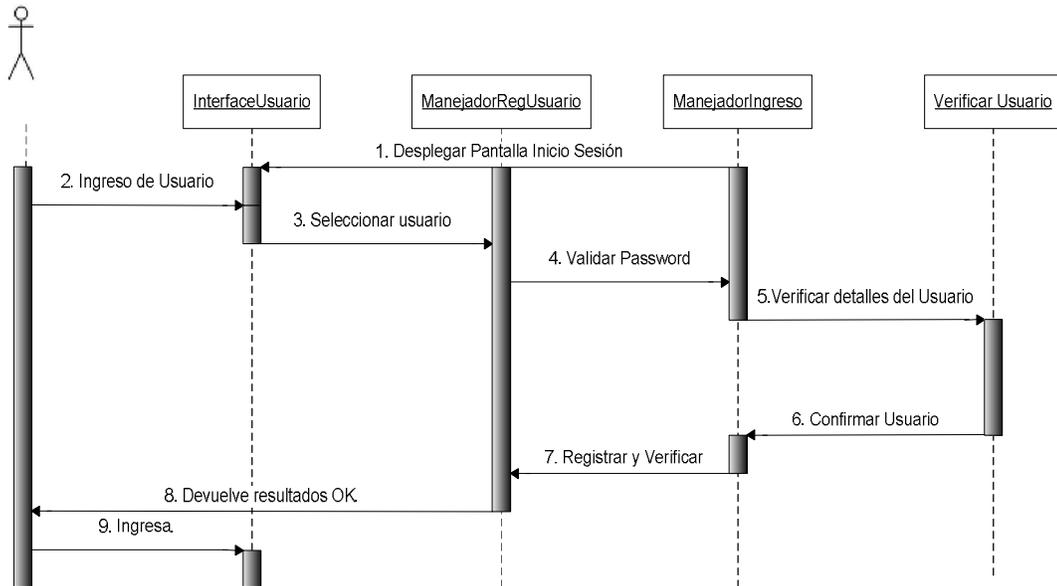


Ilustración 23 Diagrama de Secuencia Iniciar Sesión

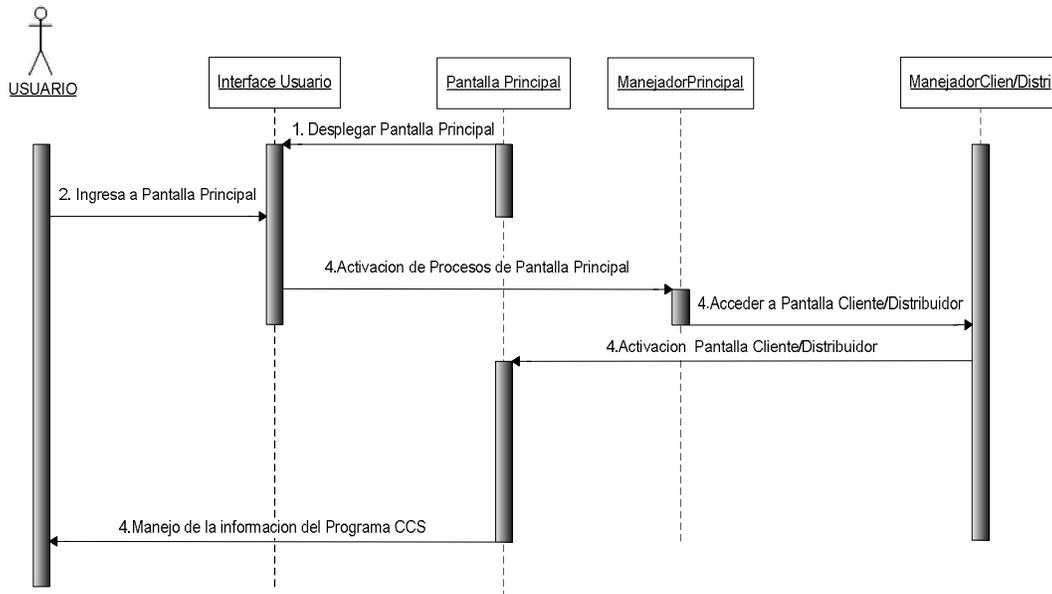


Ilustración 24 Diagrama de Secuencia Pantalla Principal

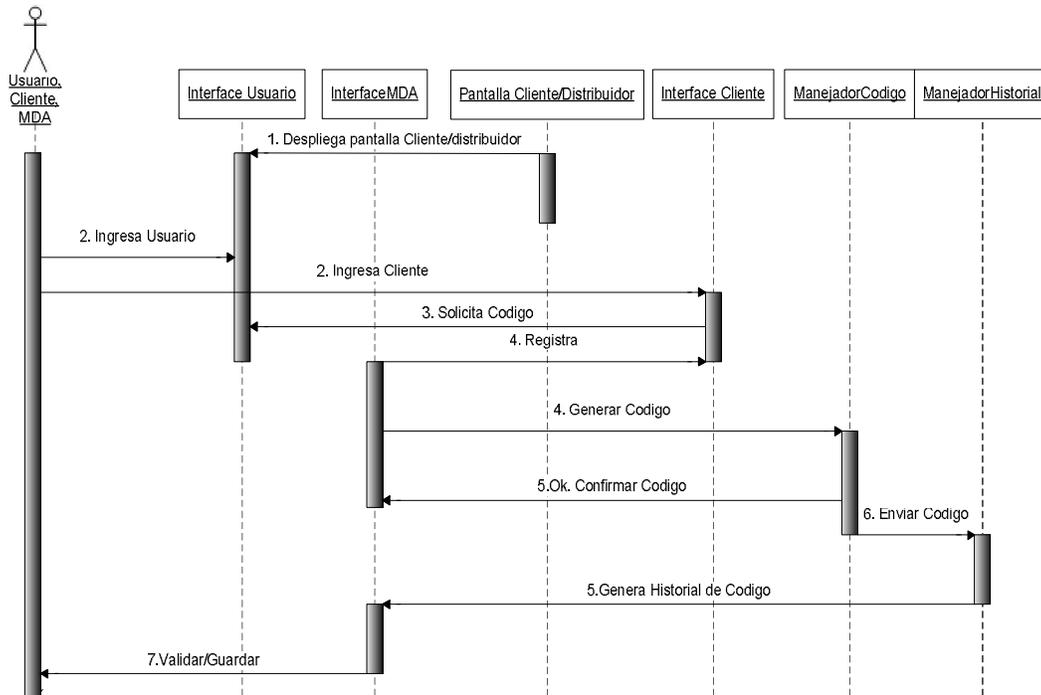


Ilustración 25 Diagrama de Secuencia Cliente / Distribuidor

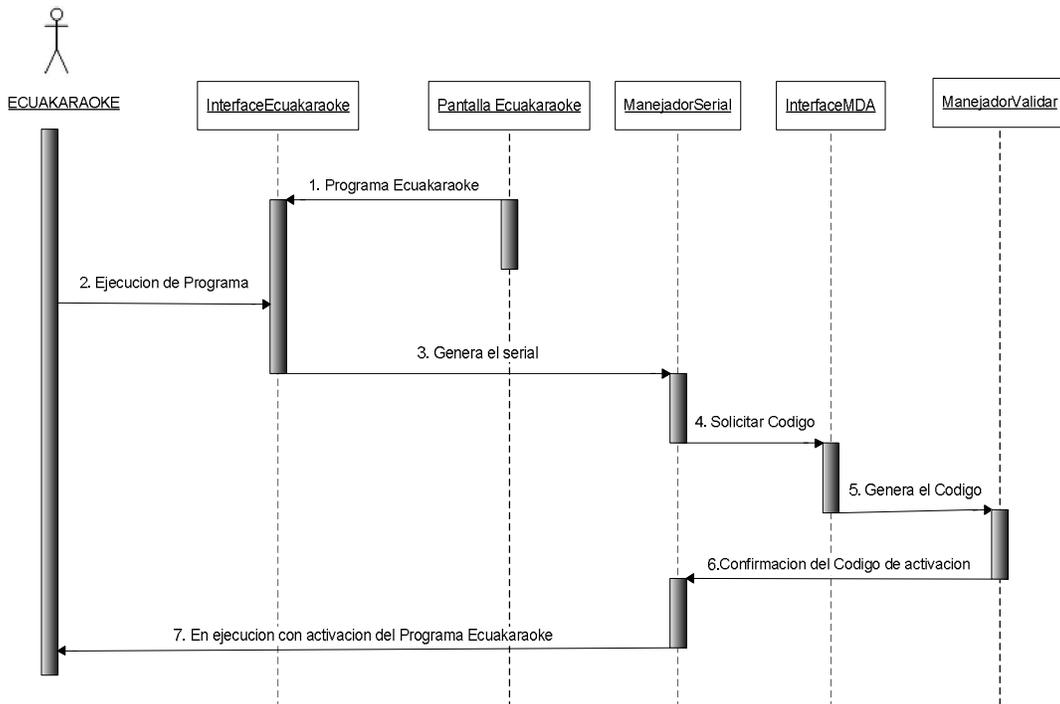


Ilustración 26 Diagrama de Secuencia ECUAKARAOKE

4.1.3 Desarrollo

Diagrama de Base de Datos

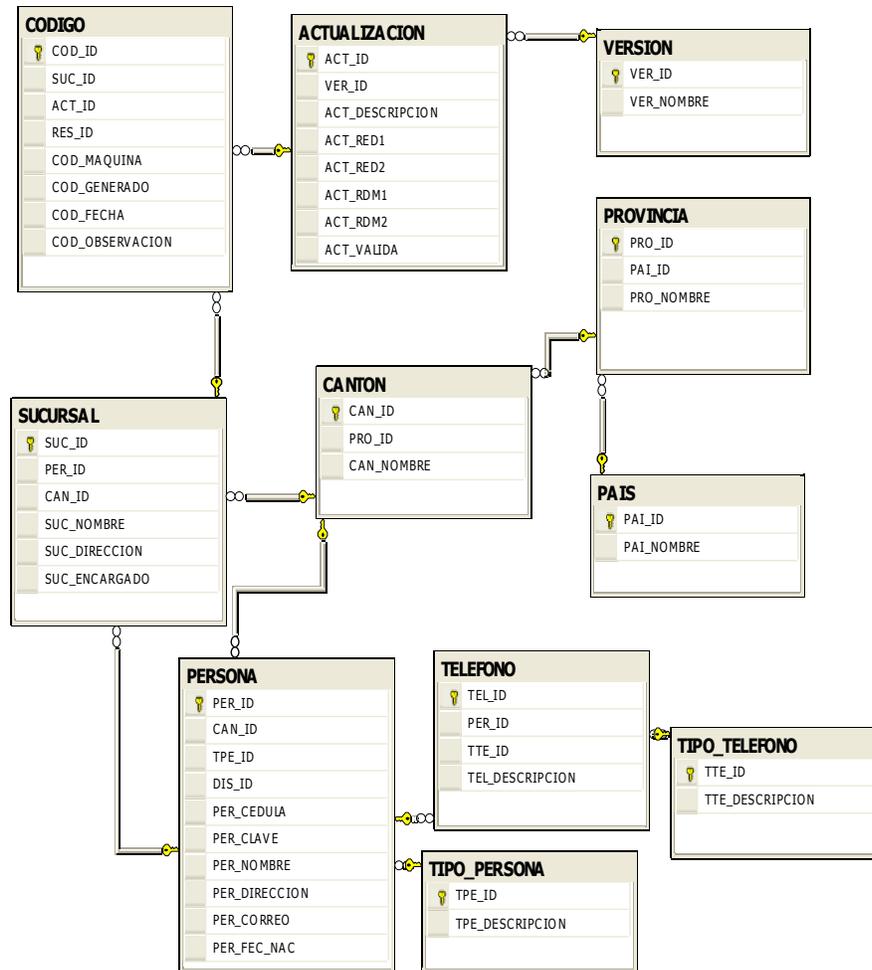


Tabla 23. Diagrama de Base de Datos

Diagramas de Clase

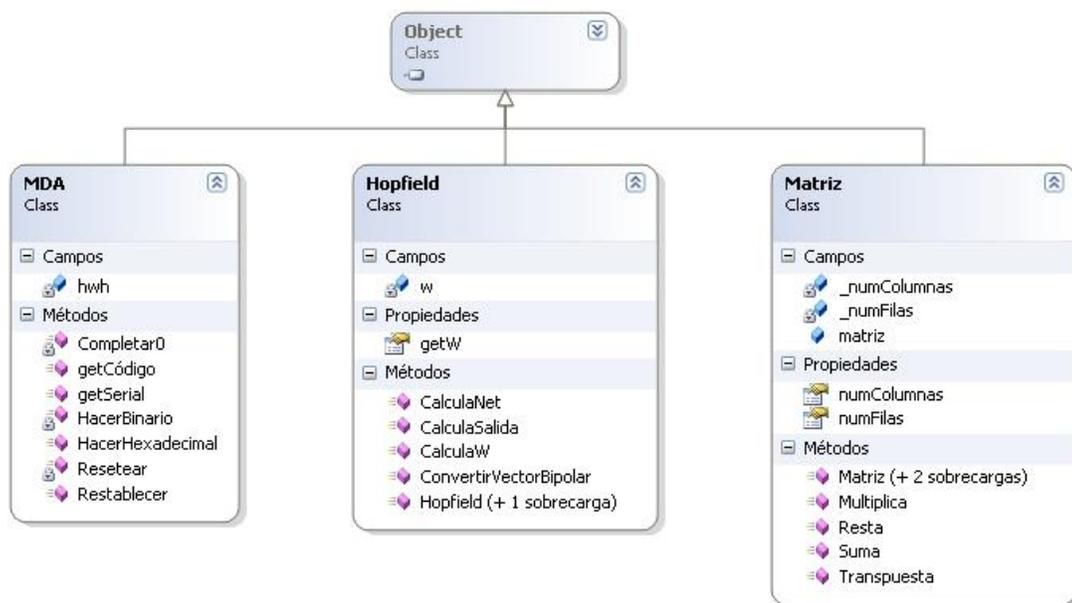


Ilustración 27. Diagrama de Clase Método de Activación (MDA)

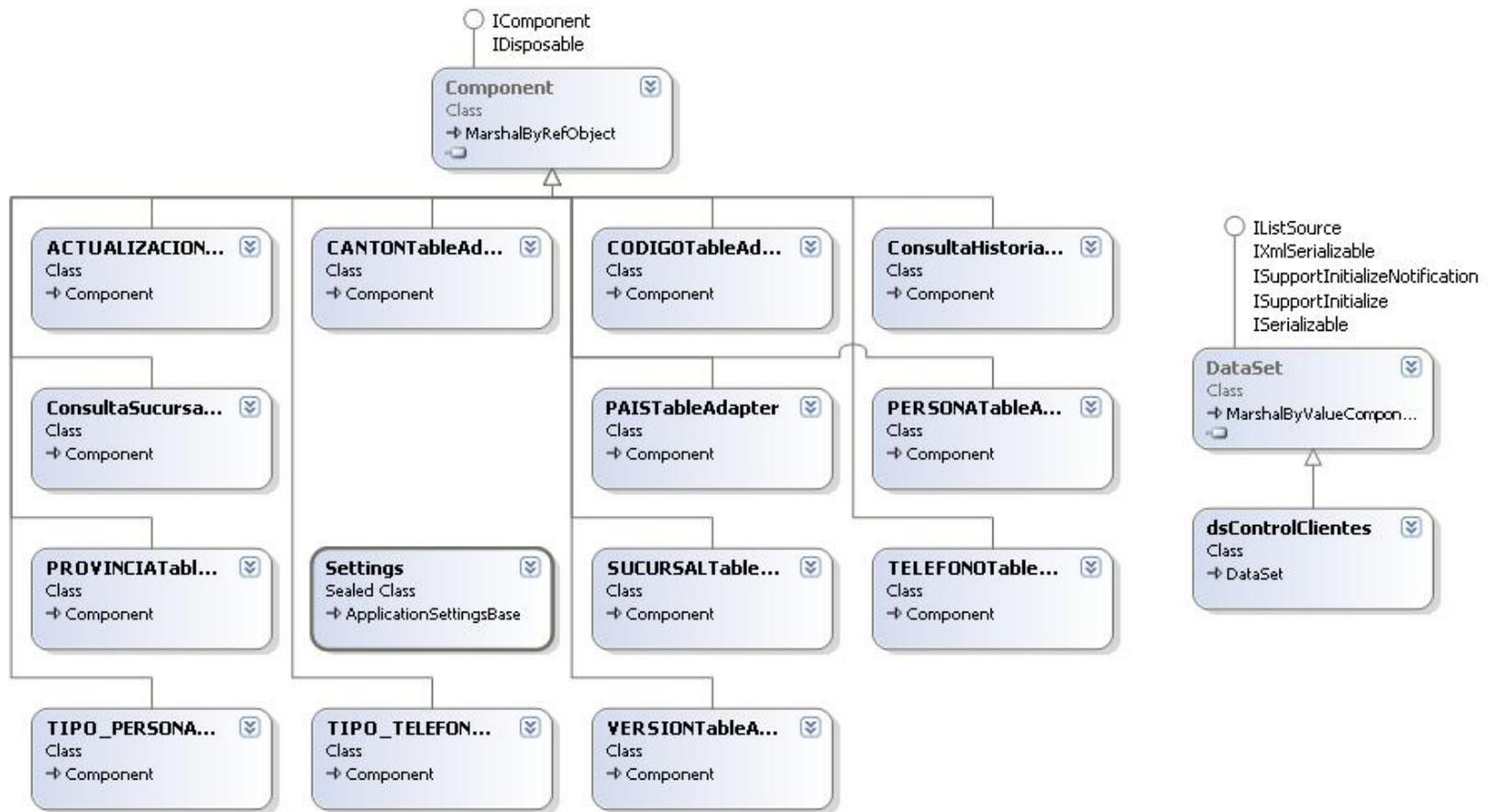


Ilustración 28. Diagrama de Clase Capa Entidad de Negocio Control de Clientes

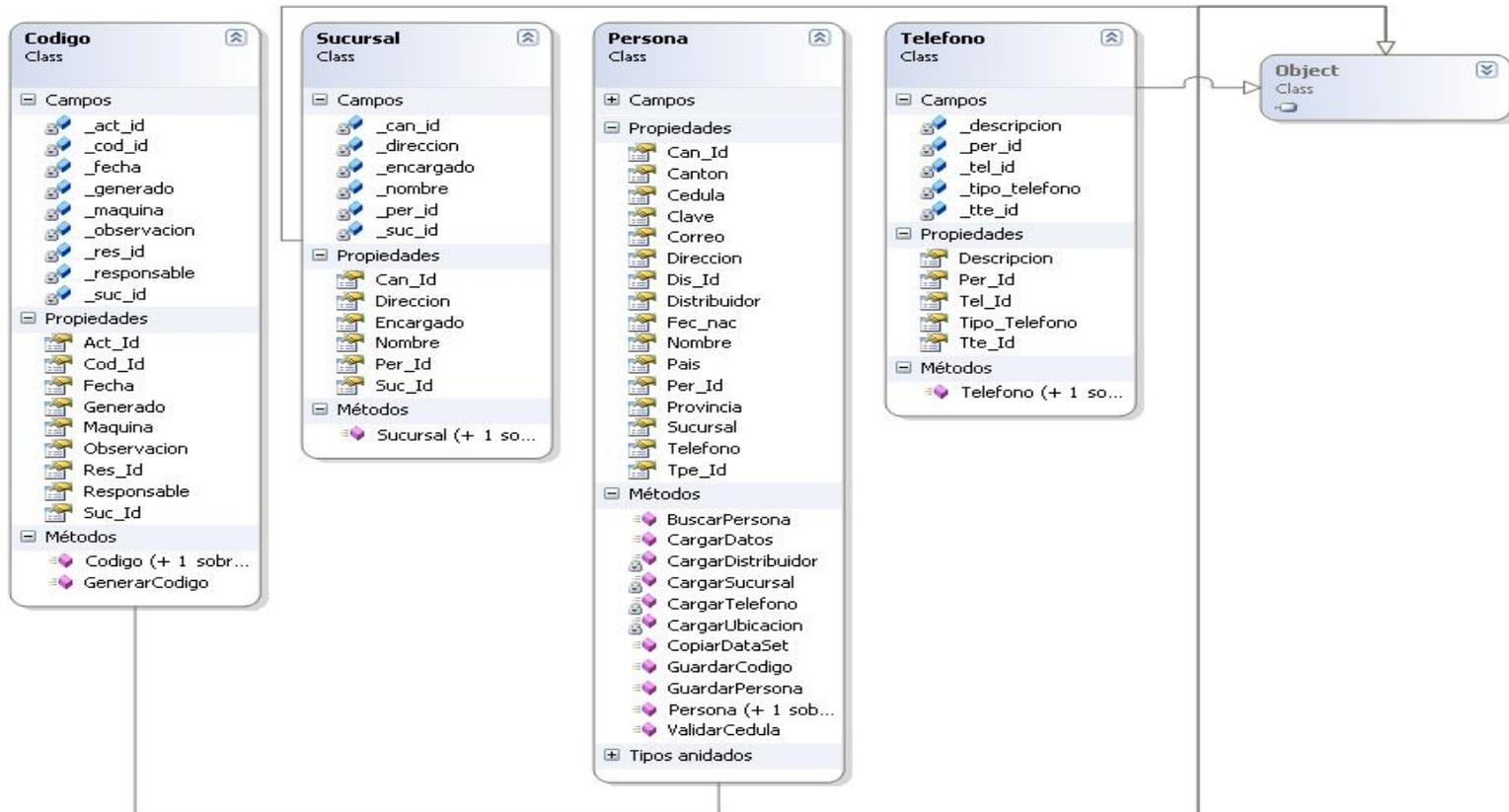


Ilustración 29. Diagrama de Clases Capa Lógica de Negocio Control de Clientes

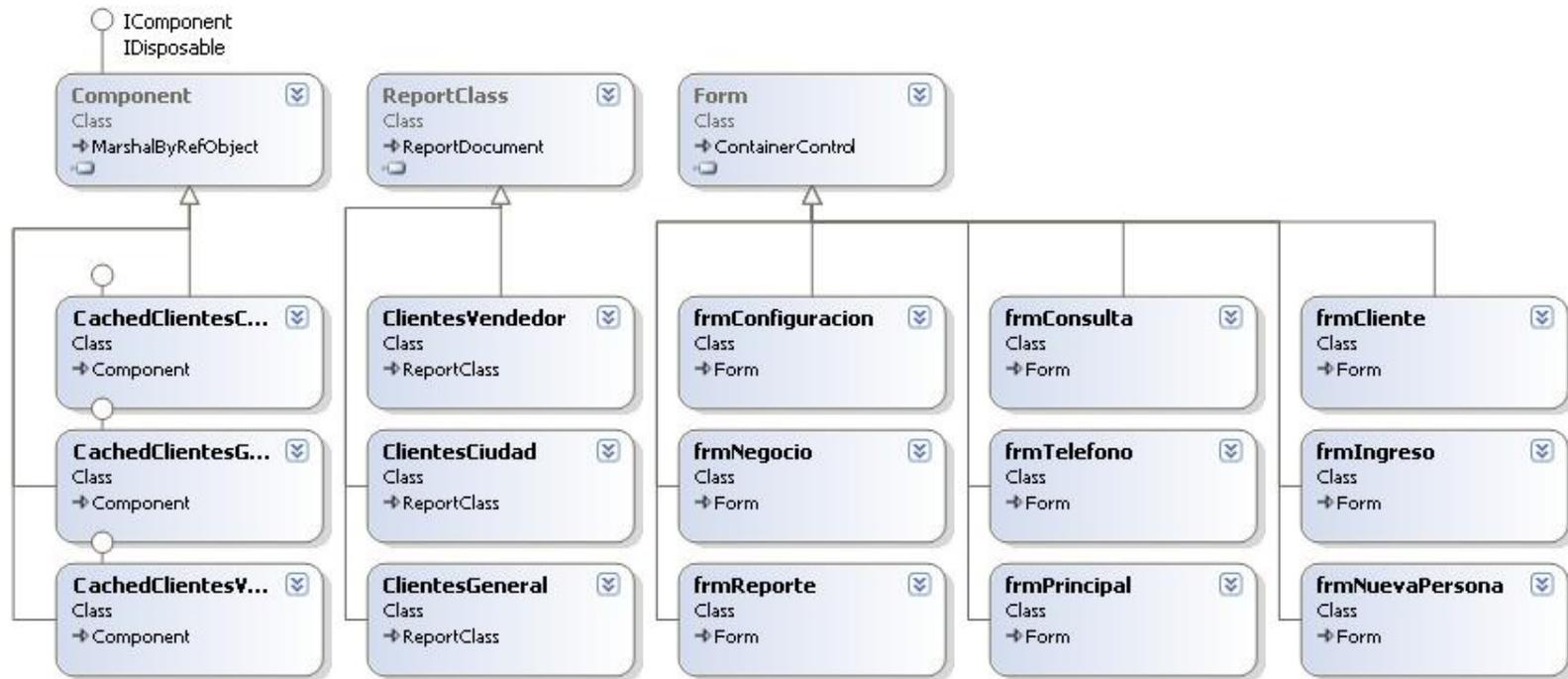


Ilustración 30. Diagrama de Clases Capa Presentación Control de Clientes

4.1.4 Estabilización

Pruebas con NUnit 2.0

NUnit realiza pruebas unitarias, procedimiento usado para validar que un módulo de código funciona apropiadamente.

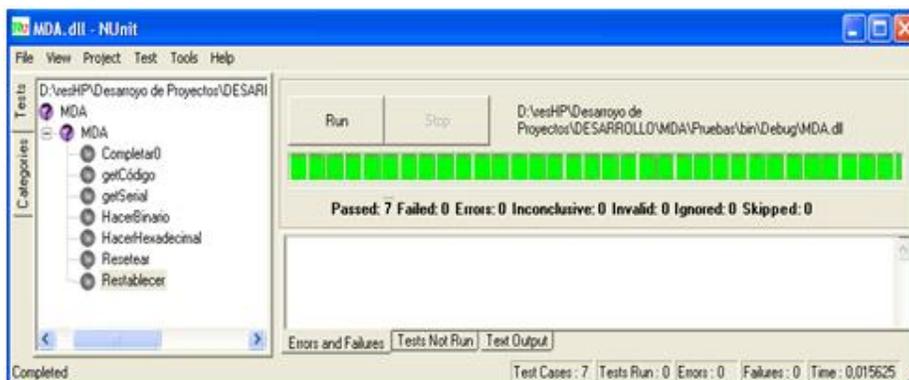


Ilustración 31. Reporte de Pruebas con NUnit 2.0

Se realizó las pruebas en la parte más importante del proyecto, NUnit analizó cada método existente en el Módulo de Activación, realizando las pruebas de código línea a línea, el programa para pruebas unitarias, no encontró errores en este módulo, demostrando la estabilidad del prototipo.

4.1.5 Implantación

Cierre de proyecto

El módulo de activación y el prototipo de control de seriales y códigos se probará con el programa ECUAKARAOKE, para la demostración de su funcionalidad, pero no se lo implantará.

5. CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

Los mejores periféricos a usarse en el sistema fueron la dirección MAC, el serial del procesador, el serial de la tarjeta madre y el serial del disco duro, los cuales se analizaron y son los más importantes en un computador.

La red que encajó en este proyecto fue Hopfield, por ser monocapa y aprendizaje no supervisado, con estas características se cumplió el objetivo de cifrar los datos ingresados.

Con el desarrollo del Método de Activación, al filtrar las direcciones MAC repetidas, restablecer las tarjetas de red con dirección MAC clonada y la combinación de los periféricos más importantes del computador, se llegó a mejorar la seguridad en la activación del programa ECUAKARAOKE.

El prototipo de Control de Códigos y Seriales, ayuda de una manera rápida y eficaz entregar los Códigos de activación cuando el cliente lo necesita.

5.2 RECOMENDACIONES

Llegar a implementar el módulo para crear seriales y códigos de activación junto con el prototipo de Control de Seriales y Códigos en un ambiente de trabajo real y obtener los resultados que se esperaban en los objetivos.

Completar los módulos para modificar y eliminar en el prototipo de Control de Códigos, así como también poder crear reportes de clientes, para poder brindar más funcionalidades y dar fuerza al proyecto.

Explotar las Redes Neuronales Artificiales, porque se las puede usar en infinitos escenarios, investigando todas sus bondades, ayudarían a desarrollar e implementar programas que cubran muchas necesidades.

BIBLIOGRAFÍA

1. Beltrán, Rafael. **Bioinformática: Simulación, vida artificial e inteligencia artificial**. Madrid, S.A. EDICIONES DIAZ DE SANTOS, 2004.
2. López, Raquel y Fernández, José. **Las redes neuronales artificiales**. Oleiros, Netbiblo, 2008.
3. Pressman, Roger. **Ingeniería del Software un enfoque práctico**. Madrid, Mc Graw Hill, 2002.
4. Turner, Michael. **Microsoft Solutions Framework Essentials: Building Successful Technology Solutions**. Microsoft Press, 2006
5. Andina de la Fuente, Diego. **Tutorial de Redes Neuronales**. Madrid, 2007
<http://www.gc.ssr.upm.es/inves/neural/ann2/concepts/concepts.htm>

ANEXO 1

1. Interface módulo de activación en ECUAKARAOKE



Ilustración 32. Nueva Pantalla de Ingreso a ECUAKARAOKE

2. Interfaces prototipo de control de códigos y seriales

2.1 Ingreso



Ilustración 33. Pantalla de Ingreso Prototipo de Control

2.2 Pantalla Principal



Ilustración 34. Pantalla Principal

2.3 Pantalla Cliente

Cédula	<input type="checkbox"/> <input checked="" type="checkbox"/>	1709218711	Nombre	<input type="radio"/>	EDISON FABIAN ORELLANA
Dirección	ONTANEDA				
Teléfono	095652511	Celular	Nacimiento	lunes, 24 de septiembre de 2007	
País	ECUADOR				
Provincia	PICHINCHA	Correo			
Cantón	QUITO	Vendedor	ALEJANDRO ORELLANA		
Negocio	DAVIDO KARAOKE --- DAVIDO --- ONTANEDA				

Ilustración 35. Pantalla Cliente

2.4 Pantalla Generar Códigos

Versión: Ecuakaraoke Profesional
 Actualización: Principal
 Negocio: DAVIDO KARAOKE
 Código:
 Serial:
 Observaciones:
 Generar y Guardar

Ilustración 36. Pantalla Generar Códigos

2.5 Pantalla Historial de Códigos

Negocio: PERSONAL

	VERSIÓN	ACTUALIZACIÓN	LICENCIA	CÓDIGO
▶	Ecuakaraoke Profesional	Principal	EP50D-5E6033E-46F3383-46FCC20-237B4	3EF56E16F55A98E0
	Ecuakaraoke Profesional	2501 - 3000	EP01-2CB520-9C069-154CC8	472AA7F07A158E0
	Ecuakaraoke Profesional	3001 - 3500	EP02-279D40-30C20D-3C18A2	4806EE9497EEDE0
	Ecuakaraoke Profesional	3501 - 3600	EP03-10FA40-1B88B0-484482	45D7105E2166880
	Ecuakaraoke Profesional	3601 - 3650	EP04-270C60-31E7BF-39D068	65113EB787552A0
	Ecuakaraoke Cristiano	Principal	EC4F7-22C9045-91CD7F-8C4457-7E84	4D9F9B6B9F0FDE1
	Ecuakaraoke Cristiano	1 - 50	EC01-309DD-54DA1-5E7EB	8F7C8A70D48F71

Ilustración 37. Pantalla Historial de Códigos

2.6 Pantalla Nuevo Cliente



The screenshot shows a web form for creating a new client. The form is titled 'Nuevo Cliente' and has a blue header bar with a 'Guardar' button on the left and a 'Salir' button on the right. The form fields are as follows:

Cédula	<input type="text" value="1720205358"/>	Nombre	<input type="text"/>
Dirección	<input type="text"/>		
Teléfono	<input type="text"/>	Nacimiento	<input type="text" value="Lunes , 24 de Septiembre de 2007"/>
Pais	<input type="text" value="ECUADOR"/>	Correo	<input type="text"/>
Provincia	<input type="text" value="PICHINCHA"/>	Vendedor	<input type="text" value="ALEJANDRO ORELLANA"/>
Cantón	<input type="text" value="QUITO"/>	Negocio	<input type="text"/>

Ilustración 38. Pantalla Nuevo Cliente

2.7 Pantalla Nuevo Teléfono



The screenshot shows a dialog box titled 'Nuevo Teléfono'. It has a blue header bar with a close button (X) on the right. The form fields are as follows:

Teléfono	<input type="text"/>
Tipo	<input type="text" value="Casa"/>

At the bottom of the dialog box is a 'Guardar' button.

Ilustración 39. Pantalla Nuevo Teléfono

2.8 Pantalla Nuevo Negocio



The image shows a software window titled "Nuevo Negocio" with a close button in the top right corner. The window contains a form with the following fields:

- Nombre:** A text input field.
- Dirección:** A text input field.
- Encargado:** A text input field.
- Pais:** A dropdown menu with "ECUADOR" selected.
- Provincia:** A dropdown menu with "PICHINCHA" selected.
- Cantón:** A dropdown menu with "QUITO" selected.

At the bottom center of the window is a button labeled "Guardar".

Ilustración 40. Pantalla Nuevo Negocio