



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

"DESARROLLO DEL TUTORIAL DEL MICROCONTROLADOR
HC11 PARA EL LABORATORIO DE MICROCOMPUTADORAS
CON MULTIMEDIA"

T E S I S
QUE PARA OBTENER EL TÍTULO DE
INGENIERO EN COMPUTACIÓN
P R E S E N T A :
MIRIAM SÁNCHEZ FLORES

DIRECTOR: ING. NORMA ELVA CHAVEZ RODRÍGUEZ



MÉXICO, D. F.

MARZO DE 2004



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

AGRADECIMIENTOS

A DIOS:

Por la oportunidad de cumplir con una de las metas más importantes de mi vida.

A LA UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO:

Por proporcionarme los conocimientos necesarios para la vida profesional.

A LA ING. NORMA ELVA CHÁVEZ RODRÍGUEZ:

Por su asesoría y apoyo en la realización de la tesis.

A PEDRO GONZÁLEZ GÓMEZ:

Por su apoyo incondicional en la realización de éste proyecto.

AL LABORATORIO DE MULTIMEDIA E INTERNET:

Por la oportunidad de haber colaborado en éste proyecto, pero especialmente a cada uno de sus integrantes que me apoyaron con sus conocimientos.

DEDICATORIAS

A MIS PAPÁS:

Por el amor y apoyo incondicional que me han demostrado hasta éste día de mi vida, por su sabiduría y experiencia.

A MIS HERMANOS:

Por ser parte importante de mi vida, por lo gratos momentos juntos y sobre todo por el lazo que nos ha unido ante cualquier adversidad.

A J. HIRAM SÁNCHEZ DELOYA:

Por compartir aspectos importantes de su vida, por su amistad sincera y por su gran motivación para seguir siempre adelante.

A MIS GRANDES AMIGAS MARTHA A. GALICIA HERNÁNDEZ Y LIZ M. GARCÍA ANAYA:

Por sus consejos y complicidad siempre oportunos, que han hecho que tenga una mejor visión de la vida.

A JORGE A. SEGOVIANO VELÁZQUEZ:

Por su amistad, por aquellos momentos juntos y por su gran alegría.

INTRODUCCIÓN	1
1. DESCRIPCIÓN DE LA METODOLOGÍA	3
1.1 Metodología TSPi	3
1.1.1 Principios de TSPi	3
1.1.2 Diseño de TSPi	4
1.1.3 Estructura y flujo de TSPi	5
1.1.4 Lógica de TSPi	6
1.1.5 Roles de Equipo de TSPi	9
1.1.6 Fases de TSPi	11
1.1.6.1 Fase Lanzamiento del proceso de software	11
1.1.6.2 Fase Estrategia de desarrollo del proceso de software	11
1.1.6.3 Fase Planeación del proceso de software	12
1.1.6.4 Fase Requerimientos del proceso de software	18
1.1.6.5 Fase Diseño del proceso de software	19
1.1.6.6 Fase Implementación del proceso de software	22
1.1.6.7 Fase Pruebas del proceso de software	23
1.1.6.8 Fase Postmortem del proceso de software	25
1.2 Proceso Personal de Software (PSP)	26
1.2.1 Cuaderno del ingeniero	26
2. PROGRAMACIÓN ORIENTADA A OBJETOS	29
2.1 Programación Orientada a Objetos	29
2.2 Lenguaje Unificado de Modelado (UML)	31
2.2.1 Modelado de Clases	31
2.2.1.1 Elementos del modelado de clases	32
2.2.1.2 Relaciones entre Clases	33
2.2.2 Diagramas principales de UML	34
2.2.2.1 Diagrama de Interacción	35
2.2.2.2 Diagrama de Actividad	36
2.2.2.3 Diagrama de Casos de Uso	38
2.2.2.4 Diagrama de Estados	41
2.2.2.5 Diagrama de Componentes	43
3. SISTEMA MULTIMEDIA	45
3.1 Definición de sistema multimedia	45
3.2 Principios para crear un sistema multimedia	45
3.3 Producción multimedia	46
3.4 La navegación multimedia	51
3.5 El sistema multimedia en el proceso pedagógico	53
3.5.1 Funciones pedagógicas de los sistemas multimedia	53
3.5.2 Posibilidades didácticas	55
3.5.3 Limitaciones didácticas	57
3.6 Aplicación de los sistemas multimedia en diversas áreas	58
4. ANÁLISIS DEL SISTEMA	61
4.1 Planteamiento del problema	61
4.2 Lanzamiento	62
4.3 Estrategia	63
4.3.1 Lista de necesidades del Sistema	63
4.3.2 Gráfica de dependencias del sistema	64
4.3.3 Estimaciones	64
4.3.4 Forma Registro de Riesgos (Forma RR)	66
4.3.5 Forma ESTRA	69
4.4 Planeación	70
4.4.1 Plan de calidad	70
4.4.2 Forma Resumen del Tamaño (RESTM)	71
4.4.3 Forma Tarea Semanal	73
4.4.4 Diagrama de Gannt	74

4.5	Requerimientos	76
4.5.1	Requerimientos del sistema	76
4.5.1.1	Alcances del proyecto	76
4.5.1.2	Resultado final esperado	76
4.5.1.3	Justificación y Beneficios	77
4.5.1.4	Identificación de los objetivos del Laboratorio de Multimedia e Internet que se relacionan con el proyecto	77
4.5.1.5	Revisión de la situación actual	77
4.5.1.6	Definición de Requerimientos	78
4.5.2	Requerimientos del Cliente	79
4.5.2.1	Diagramas de Casos de Uso	81
4.5.2.2	Diagramas de Actividad	86
4.5.2.3	Diagramas de Secuencia	93
4.5.3	Requerimientos de Interfaz	104
4.5.3.1	Requisitos del sistema para el desarrollo de la interfaz	104
4.5.3.2	Restricciones en diseño/implementación	105
4.5.3.3	Criterios de medición	106
4.6	Diseño	110
4.6.1	Guión del Tutorial del Microcontrolador HC11	110
4.6.1.1	Ventana Teoría Básica	110
4.6.1.2	Ventana Copiar	111
4.6.1.3	Ventana Buscar	111
4.6.1.4	Ventana Prácticas	112
4.6.1.5	Ventana 2 Prácticas	112
4.6.1.6	Ventana Software	113
4.6.1.7	Ventana Manuales	113
4.6.1.8	Ventana 2 Manuales	114
4.6.1.9	Ventana Acerca de	114
4.6.2	Especificaciones de la Interfaz del Tutorial del Microcontrolador HC11	114
5.	IMPLEMENTACIÓN DEL SISTEMA	119
5.1	Implementación	119
5.1.1	Sección TEORÍA	125
5.1.2	Sección PRÁCTICAS	128
5.1.3	Sección MANUALES	136
5.1.4	Sección SOFTWARE	140
5.1.5	Sección BUSCAR	144
5.1.6	Sección COPIAR	146
5.1.7	Sección Acerca de	149
5.2	Pruebas	150
5.2.1	Estrategia de Pruebas	150
5.2.2	Lista de pruebas	150
5.2.3	Forma Registro de Pruebas (REGPR)	152
5.2.4	Resultados finales de las pruebas	155
5.3	Pruebas del sistema por medio de encuestas	156
5.4	Postmortem	158
	CONCLUSIONES	159
	ANEXO A	161
	ANEXO B	165
	ANEXO C	166
	ANEXO D	167
	BIBLIOGRAFÍA	169

INTRODUCCIÓN

El presente trabajo se plantea como respuesta a una necesidad de contar con un elemento de software (Tutorial) como apoyo didáctico para el Laboratorio de Microcomputadoras de la Facultad de Ingeniería.

En la actualidad la tecnología multimedia nos permite unir dos herramientas básicas dentro de los nuevos modelos educativos: la computadora y los medios audiovisuales, es decir el empleo combinado del sonido y la imagen fija o con movimiento. El uso de los sistemas multimedia facilita al estudiante la adquisición de nuevos conceptos gracias a los modelos animados de diversos procesos mientras que el alumno dirige de manera interactiva con la computadora su propio aprendizaje.

Es por ello que se considera de gran importancia, contar con una herramienta didáctica para la comunidad de la Facultad de Ingeniería, específicamente para los alumnos del Laboratorio de Microcomputadoras.

El trabajo consta de 5 capítulos en los cuales se resaltan los aspectos más importantes para la construcción del Tutorial del Microcontrolador HC11.

En el Capítulo 1 se describe la metodología a seguir para la elaboración del Tutorial del Microcontrolador HC11. Esta metodología es denominada Proceso de Software en Equipo (TSPi), el cual es un marco de trabajo definido para la realización de software realizado en equipo. Se presenta como una guía de los pasos que deben seguir porque indica como aplicar el conocimiento de la ingeniería de software y los principios del proceso en un ambiente de equipo de trabajo que pretenda desarrollar software.

Otro proceso que se sigue es el Proceso Personal de Software (PSP) con el cual se lleva un control de las actividades personales que se tienen por cada persona que esta integrada en un equipo que participa en un proyecto de software.

El Capítulo 2 da una visión general de la programación orientada a objetos, la utilización de Lenguaje Unificado de Modelado (UML) el cual es una herramienta fundamental en el diseño de software.

UML también intenta solucionar el problema de propiedad de código que se da con los desarrolladores, al implementar un lenguaje de modelado común para todos los desarrollos se crea una documentación también común, que cualquier desarrollador con conocimientos de UML será capaz de entender, independientemente del lenguaje utilizado para el desarrollo.

Su utilización es independiente del lenguaje de programación y de las características de los proyectos, ya que UML ha sido diseñado para modelar

cualquier tipo de proyectos, tanto informáticos como de arquitectura, o de cualquier otro ramo.

El Capítulo 3 contiene la información relacionada con un sistema multimedia, cuales son sus objetivos en la educación y su campos de aplicación en diversas áreas.

En cuanto al Capítulo 4 Se hará un análisis del sistema que se va a desarrollar; ésta sección abarcará las siguientes fases de TSPi: Lanzamiento, Estrategia, Planeación, Requerimientos y Diseño.

El Capítulo 5 abarcará las últimas fases de la metodología utilizada en este proyecto, es decir, Implementación, Pruebas y Postmortem.

Al finalizar el Capítulo 5 se presentan las conclusiones relacionadas con el proyecto.

Finalmente se presentan los anexos A, B, C Y D correspondientes a algunas secciones del Capítulo 4.

1. DESCRIPCIÓN DE LA METODOLOGÍA

1.1 Metodología TSPi

"El Proceso de Software en Equipo (TSP) es un marco de trabajo definido para cursos de Ingeniería de Software realizados en equipo"¹.

Se presenta como una guía de los pasos que deben seguir, porque indica cómo aplicar el conocimiento de la ingeniería de software y los principios del proceso en un ambiente de equipo de trabajo. También define los roles para cada uno de los integrantes. Está diseñado con base en el Proceso de Software en Equipo (TSP)

Necesidad de un proceso en la Ingeniería

Al desarrollar un proyecto de software con frecuencia los integrantes pierden demasiado tiempo en establecer un mecanismo de trabajo que solucione las diferentes necesidades, las cuales se clasifican en los siguientes tipos: laborales, afectivas y de apoyo.

Las necesidades laborales indican como trabajar juntos, definir funciones, establecer una estrategia para hacer el trabajo, coordinar las tareas y reportar el progreso. Las necesidades afectivas, indican como construir y mantener relaciones de trabajo efectivas, las cuales requieren de objetivos en común, un plan de acción y una dirección apropiada. Las necesidades de apoyo, indican como entender las habilidades y debilidades de los demás, apoyar a los compañeros del equipo y ser capaces de pedir ayuda cuando sea necesario.

TSPi mejora la productividad; aunque en un principio, la planeación y los pasos para formar el equipo parezcan tomar gran tiempo, son una parte esencial para hacer un proyecto en equipo.

1.1.1 Principios de TSPi

TSPi se basa en los siguientes cuatro principios básicos:

1. Tener como guía un proceso definido. Con ello se adquiere un aprendizaje más efectivo y una retroalimentación rápida. Eso es posible, con el marco de trabajo definido, medible y repetible de los guiones y formas del proceso. Y también, con los resultados de las evaluaciones realizadas en cada ciclo.
2. Establecer una combinación de objetivos específicos, un buen ambiente de trabajo, una guía y un líder capaz de lograr un equipo de trabajo productivo.

¹ Watts S. Humphrey Introduction to the Team Software Process Pág.6

3. Definir roles, prácticas y métodos de desarrollo, para generar beneficios necesarios al enfrentarse con problemas del proyecto actual y que permitan obtener soluciones efectivas.
4. Construir la instrucción sobre un esquema disponible de conocimiento anterior, para hacerla más efectiva. TSPi se basa en este aspecto por la experiencia obtenida a partir del trabajo con equipos de software y en cursos académicos.

1.1.2 Diseño de TSPi

A continuación se describen las siete decisiones de diseño.

- ❖ Proporcionar un marco de trabajo simple que se construya sobre la base de PSP

El propósito de un proceso es ayudar a hacer una tarea como desarrollar un producto o aprender a hacer un proyecto en equipo. Aunque TSPi proporciona formas y guiones, la mayoría son similares a los que se utilizan en Proceso Personal de Software (PSP). Es por eso, que un entrenamiento previo en PSP es un requisito para TSPi.

Con la asignación de roles, TSPi permite enfocarse en los elementos que personalmente se deberán aprender. Esos roles especifican quien hará cada una de las tareas de planeación, seguimiento, calidad, apoyo y liderazgo. Cada integrante debe concentrarse en las tareas específicas de acuerdo a su rol y no preocuparse por las responsabilidades de los demás.

Durante un proyecto multiciclo de TSPi, se aconseja obtener experiencia desempeñando varios roles. Eso proporcionará un amplio conocimiento en el proceso y permitirá tener un entendimiento más profundo sobre los aspectos del desarrollo de software en equipo. También ayudará a obtener una mejor apreciación de las habilidades e intereses personales.

- ❖ Desarrollar productos en varios ciclos

En la creación de proyectos por medio de TSPi, se realizarán dos o tres ciclos de desarrollo. En cada uno se definen los requerimientos, diseño, implementación y un proceso de pruebas.

En el primer ciclo, se construye un subconjunto con funciones mínimas del producto final. El segundo ciclo se construye con base en los resultados del primer ciclo; entonces se pueden cambiar los roles del equipo, ajustar el proceso o aplicar más disciplina en los métodos de calidad. Si existiera tiempo se podrían repetir esos ciclos por tercera y cuarta vez.

- ❖ Establecer medidas estándares para la calidad y desempeño

Las medidas son una parte esencial para hacer un trabajo de alta calidad. PSP proporciona las medidas básicas y la medición de destrezas que se necesitan para medir y evaluar la calidad de trabajo de cada integrante del equipo. TSPi indica cómo interpretarlas y aplicarlas a un proyecto.

- ❖ Proporcionar medidas precisas para los equipos y los integrantes

Con las medidas de TSPi, el desempeño de cada uno de los integrantes del equipo será obvio. Aunque el propósito principal de esas medidas, es ayudar a hacer un mejor trabajo, cada uno conocerá lo que los demás están haciendo. Por esto, si alguien no desarrolla sus actividades adecuadamente tendrá comentarios de los demás integrantes del equipo.

- ❖ Aplicar evaluaciones de rol y de equipo

TSPi proporciona la forma Evaluación del Equipo. El objetivo es evaluar el desempeño de cada rol y el proceso de trabajo. Lo cual permitirá a cada integrante tener conocimientos sobre esa información.

- ❖ Requerir una disciplina en el proceso

Porque la Ingeniería de Software no implica un desempeño personal disciplinado, debido a que el proceso de software no impone una disciplina natural en los ingenieros y porque un trabajo disciplinado requiere conseguir altos estándares y un apoyo competente, se aconseja al instructor de TSPi seguir el proceso y reunir los datos de las formas para un análisis posterior.

- ❖ Proporcionar guía en los problemas del trabajo en equipo

Ocasionalmente, existen problemas entre los integrantes de un equipo, ya sea, por ser personas sin experiencia o por su propia personalidad. Sin embargo, con guía y apoyo, la mayoría de los ingenieros pueden ser integrantes efectivos.

1.1.3 Estructura de TSPi

La figura 1.1 muestra la estructura de TSPi. "El proceso está constituido por múltiples ciclos de desarrollo. En cada uno se definen los objetivos del producto que se cumplirán. Posteriormente, en cada ciclo se desarrollan las siete fases: Lanzamiento, Estrategia, Planeación, Requerimientos, Diseño, Implementación, Pruebas y Postmortem. La estructura de TSPi está siendo modelada con un diagrama de clases"², donde se utiliza "el símbolo de agregación"³,

² Diagrama que muestra un conjunto de clases y sus relaciones.

³ El símbolo de agregación forma parte de los diagramas de clases. Modela una relación "Todo/Parte", en donde la clase señalada por el rombo representa un elemento más grande (todo), la cual está formada por elementos más pequeños (partes) que están representadas por la clase que está en el otro extremo de la línea dirigida.

representado por una línea dirigida con punta en forma de rombo, y cuyo significado corresponde a las frases "está constituido" y "está integrada".

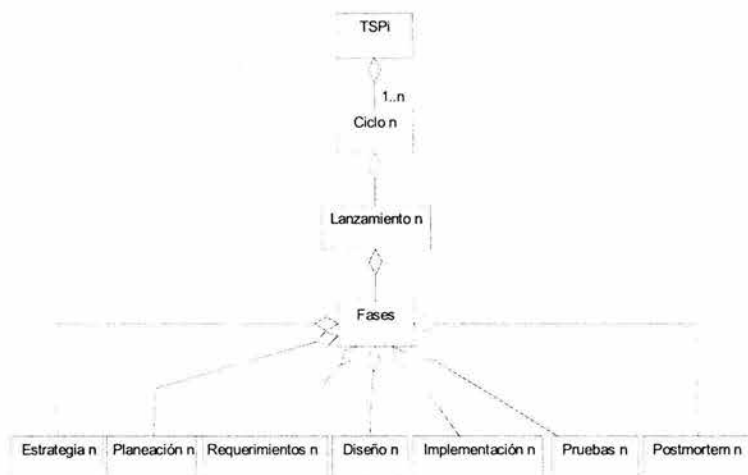


Figura 1.1 Estructura de TSPi

Al inicio se realiza la definición del problema que el cliente está requiriendo. Se describen los objetivos del producto y los equipos generan la estrategia de desarrollo. Se aconseja que en la primera versión del producto se defina un número menor de funcionalidades para el primer ciclo. Después el equipo estimará el tamaño de las mismas y decidirá cuáles integrar en cada ciclo. En el segundo ciclo, los integrantes repiten los mismos pasos, pero esta vez, el producto base del ciclo anterior se utiliza para generar un producto con incremento de funcionalidades. Si se tuviese más tiempo, se podrían desarrollar ciclos subsecuentes.

1.1.4 Lógica de TSPi

El objetivo principal de la lógica de TSPi es definir los equipos, explicar cómo trabajan e indicar algunos problemas comunes.

Un problema significativo de las personas que trabajan en los equipos de software es la incapacidad para manejar la presión del tiempo. Los equipos necesitan saber cómo trabajar eficientemente y generar productos de calidad, especialmente cuando están bajo un horario intenso. Al guiarlos a través de un proceso de planeación y una estrategia; TSPi indica cómo manejar la presión. Para hacer el trabajo, establecen una estrategia, estiman el tamaño de los productos que construirán y realizan un plan.

Algunos problemas muy comunes que se presentan en los equipos de trabajo son los siguientes:

- ❖ Liderazgo inefectivo, provoca que los equipos tengan problemas para apegarse a los planes y mantener una disciplina personal.
- ❖ Falta de cooperación entre los integrantes que no muestran voluntad para trabajar cooperativamente con el equipo.
- ❖ Diferente grado de desempeño en el equipo, sucede cuando los integrantes tienen diferentes destrezas y habilidades, así como diferentes motivaciones y energía.
- ❖ La falta a los tiempos de entrega y metas concretas, así como establecimientos de tiempos que nunca se cumplen; todo eso es consecuencia de tener asignado un líder sin experiencia, carencia de objetivos claros, de un proceso o plan definido.
- ❖ La falta de calidad en los productos es consecuencia de inspecciones superficiales en los requerimientos, un diseño documentado pobremente, o prácticas de implementación mal hechas.
- ❖ Generalmente, durante las fases de diseño o implementación los ingenieros piensan en mejorar los productos. Cuando esto sucede la razón consiste en que no hubo una definición clara de los requerimientos del cliente, y eso se puede confundir entre las adiciones reales y modificaciones por parte de los ingenieros
- ❖ Evaluación irreal entre los integrantes del equipo al rechazar evaluar el desempeño de sus compañeros.

Definición de Equipo

"Un equipo está formado por lo menos de dos personas, cuyo trabajo está enfocado hacia un objetivo en común, donde a cada una se le han asignado roles o funciones específicas que deben desempeñar; y existe dependencia entre ellas mismas para poder finalizar tal objetivo"⁴.

TSPi se diseñó para equipos de cinco o seis integrantes, tomando en cuenta la importancia del tamaño de un equipo, donde lo más conveniente es que esté formado de cuatro a ocho personas.

Las condiciones que a continuación se mencionan son necesarias para que un grupo opere como equipo.

1. Clara definición de las tareas, de los roles y conocimiento del objetivo a

⁴ Watts S. Humphrey Introduction to the Team Software Process Pág.13

cumplir por parte de cada integrante.

2. Conocimiento de cada integrante sobre el trabajo a cumplir y el rol de los demás.

3. Control del equipo sobre las tareas a realizar porque cada integrante conoce sus responsabilidades, cómo debe realizarlas, a partir de que fecha iniciarlas y cuándo finalizarlas.

TSPi guía a los equipos en su propia construcción estableciendo objetivos, asignando roles, estableciendo planes y manteniendo comunicación entre los integrantes del equipo.

Objetivos	TSPi proporciona un conjunto inicial de objetivos para el equipo y cada uno de sus integrantes, los cuales son necesarios para lograr un acoplamiento eficaz entre ellos mismos. En ciclos subsecuentes el equipo creará nuevos objetivos de acuerdo a la experiencia de la primera interacción.
Roles	El siguiente paso es indicar las responsabilidades mediante la asignación de roles a cada integrante. TSPi propone los siguientes roles: Líder del equipo (LE), Administrador de Desarrollo (AD), Administrador de Planeación (AP), Administrador de Calidad y Proceso (ACP) y Administrador de Apoyo (AA).
Planes	Posteriormente se define la estrategia para lograr los objetivos. Eso consiste en dividir el trabajo total en partes para los diferentes ciclos de desarrollo. El equipo define el contenido funcional del producto para cada ciclo, el tamaño esperado, las maneras en que se integrarán y evaluarán las piezas para generar el producto final. Posteriormente se realiza el plan de desarrollo del proyecto; en el cual se especifica el tamaño de los productos de cada ciclo, el tiempo para producir cada uno, el orden de las actividades y los responsables de cada tarea.
Comunicación	Mantener una comunicación efectiva es un problema en los equipos de trabajo; para eso TSPi aconseja realizar reuniones semanales entre los integrantes.
Comunicación externa	Se refiere a la comunicación entre el equipo y otras partes (administrador o instructor). Con frecuencia, los equipos se comunican con el instructor o administrador cuando tienen problemas. TSPi pide al Líder de Equipo que haga un reporte que consiste en un resumen semanal y lo presente al Instructor. Ahí se muestra lo que el equipo ha realizado y lo que sus integrantes planean para la siguiente semana.

1.1.5 Roles de Equipo de TSPi**Rol: Líder de Equipo**

Este rol tiene como objetivos:

- ❖ Construir y mantener un equipo eficaz.
- ❖ Motivar a todos los integrantes del equipo a comprometerse en el trabajo del proyecto.
- ❖ Resolver todos los conflictos entre los integrantes del equipo.
- ❖ Mantener al instructor completamente informado sobre el avance del equipo.

El Líder de Equipo tiene que establecer tiempos para realizar el trabajo requerido que permita generar productos de calidad.

Rol: Administrador de Planeación

El objetivo que tiene este rol es apoyar y guiar a los integrantes del equipo en la planeación y seguimiento de su trabajo.

Las principales actividades que debe desempeñar dentro de éste rol son:

- ❖ Conducir al equipo en la generación del plan para el siguiente ciclo de desarrollo.
- ❖ Dirigir al equipo en la generación del calendario para el siguiente ciclo de desarrollo.
- ❖ Dirigir al equipo en la generación del plan balanceado.
- ❖ Obtener los planes detallados de cada ingeniero.
- ❖ Mantener seguimiento del avance del equipo de acuerdo al plan.
- ❖ Participar en la generación del reporte del ciclo de desarrollo.

Rol: Administrador de Desarrollo

El objetivo del administrador de desarrollo es conducir y guiar al equipo en la definición, diseño, desarrollo y pruebas del producto.

Y debe cumplir con las siguientes actividades principales:

- ❖ Dirigir al equipo en la generación de la estrategia de desarrollo.
- ❖ Dirigir al equipo en las estimaciones de tamaño y tiempo preliminares.
- ❖ Dirigir el desarrollo de la especificación de requerimientos. (ERS)
- ❖ Dirigir al equipo en la implementación del producto.
- ❖ Dirigir al equipo en el desarrollo de la construcción, integración y planes de prueba del sistema.
- ❖ Dirigir al equipo en el desarrollo de materiales de prueba y ejecución de las pruebas.

- ❖ Dirigir al equipo en la generación de la documentación de los productos de usuario.
- ❖ Participar en la generación del reporte del ciclo de desarrollo.

Rol: Administrador de Calidad y Proceso

Cuyo objetivo es apoyar al equipo en la definición de las necesidades del proceso, en elaborar el plan de calidad y en mantener seguimiento en el proceso y en la calidad del producto.

Y sus principales actividades a desarrollar son:

- ❖ Dirigir al equipo en la generación y seguimiento del plan de calidad.
- ❖ Informar al equipo de los problemas de calidad.
- ❖ Dirigir al equipo en la definición y documentación de procesos y en la mejora de los mismos.
- ❖ Establecer y mantener los estándares de desarrollo del equipo.
- ❖ Dirigir las inspecciones del equipo.
- ❖ Registrar las reuniones del equipo.
- ❖ Participar en la generación del reporte del ciclo de desarrollo

Rol: Administrador de Apoyo

El administrador de apoyo debe cumplir con el objetivo de colaborar con el equipo en la determinación, obtención y mantenimiento de las herramientas necesarias para cumplir con las necesidades administrativas y aplicar la tecnología definida.

Y desarrolla las actividades siguientes:

- ❖ Dirigir al equipo a determinar las necesidades de apoyo y a obtener las herramientas y facilidades.
- ❖ Revisar los cambios en los productos.
- ❖ Evaluar el impacto y beneficio de cada cambio.
- ❖ Recomendar al equipo que cambios hacer.
- ❖ Administrar el sistema de administración de configuración.
- ❖ Mantener una copia protegida de todos los elementos controlados.
- ❖ Hacer los cambios aprobados sólo a la versión controlada.
- ❖ Mantener copias de los elementos y versiones.
- ❖ Mantener el glosario del sistema.
- ❖ Mantener el sistema de seguimiento de riesgos y asuntos del equipo.
- ❖ Fomentar el reuso en el equipo.
- ❖ Participar en la generación del reporte del ciclo.

1.1.6 Fases de TSPI**1.1.6.1 Fase Lanzamiento del proceso de software**

El Objetivo de ésta fase es describir el proceso de lanzamiento del equipo.

En la fase de lanzamiento se definen los objetivos del equipo y cómo establecerlos, además se presenta un horario de actividades de acuerdo a cada integrante, en el que se incluye el tiempo destinado al desarrollo del proyecto. Esta información se podrá actualizar en ciclos siguientes de trabajo.

Los equipos necesitan establecer sus relaciones de trabajo, determinar los roles de sus integrantes y estar de acuerdo en los objetivos.

Objetivos del equipo

Los objetivos son un paso esencial en la formación del equipo y se definen al inicio de cada proyecto porque establecen el marco para la estrategia y el plan; y proporcionan la base para cualquier producto que el equipo generará.

Pasos para establecer objetivos

1. Definir los objetivos del proyecto.
2. Especificar cómo medirlos.
3. Describir la razón por la que se van a utilizar los objetivos creados por el equipo y no los proporcionados por TSPI.
4. Proporcionar una copia de los objetivos a los integrantes del equipo y al instructor.
5. Proporcionar una copia de los objetivos al AA para que los anexe en la carpeta del proyecto.

1.1.6.2 Fase Estrategia de desarrollo del proceso de software

Esta fase de TSPI tiene como objetivo crear una estrategia para realizar el trabajo. Crear el diseño conceptual del producto. Hacer estimaciones preliminares sobre el tamaño del producto y el tiempo de desarrollo. Finalmente, documentar la estrategia.

Es conveniente planear antes de iniciar un proyecto porque al momento de desarrollar el plan los equipos obtienen una apreciación común del trabajo que deben hacer. Tal documento se utiliza para mantener el seguimiento de las actividades a realizar y establecer fechas de terminación para las mismas. También permite que los equipos desde un inicio mantengan ciertos cuidados sobre los problemas comunes que pueden ocurrir.

Estrategia de TSPi

Consiste en aplicar un proceso cíclico para desarrollar el producto. En cada ciclo se deciden cuáles funcionalidades desarrollar. En el primero de ellos se diseña, implementa y evalúa una primera versión del producto; en el segundo ciclo, se incrementan las funcionalidades del producto para generar una segunda versión. Si hay tiempo, se produce una tercera versión.

El diseño conceptual

El diseño conceptual es el punto inicial para la planeación del proyecto. Es importante que los participantes determinen cómo construirán el producto, cuáles serán los componentes principales y las funcionalidades de cada uno; por último, deberán estimar el tamaño y tiempo requerido para el desarrollo.

Administración de riesgos

Un asunto se define como un suceso real; cada equipo tiene que planear asignaciones de tiempo para manejar los asuntos conocidos. Un riesgo se detecta cuando algo puede o no suceder; para evitarlos o controlarlos, es conveniente tomarlos en cuenta desde el principio y determinar las precauciones adecuadas para ello.

Al establecer una estrategia se debe definir el criterio de la misma; identificar diferentes estrategias que se podrían utilizar; identificar riesgos y beneficios en cada una; hacer una evaluación comparativa de las mismas; decidir cuál utilizar y documentarla.

Estrategia de reuso

Es conveniente que desde el inicio de un proyecto de software se tome en cuenta el reuso en el código. Identificar casos donde un solo componente o función pudiera satisfacer varias necesidades funcionales, permitiría ahorrar tiempo en el desarrollo de los ciclos.

1.1.6.3 Fase Planeación del proceso de software

Esta fase tiene como finalidad realizar el plan de desarrollo del proyecto y el plan de calidad. Aplicar el criterio de estándares de calidad para generar un producto de alta calidad.

Plan balanceado

Un proceso de planeación es el contexto mediante el cual se diseña la manera en la que se hará un trabajo. Al planear es común entablar compromisos en los

cuales se establezcan fechas y se detecten los recursos necesarios. Una de las causas principales que originan problemas de horario es la carga de trabajo no balanceada, sobre todo cuando se asigna mayor trabajo a algunos ingenieros que a otros. Un plan balanceado es aquel en el que todos los integrantes del equipo terminan las tareas planeadas en el orden apropiado y, aproximadamente al mismo tiempo.

Criterio de entrada

Realizar una lista de los productos que serán generados en el ciclo de desarrollo y estime sus tamaños

- ❖ Asigne nombres a las partes del sistema (especificadas en el diseño conceptual) y estime el tamaño de cada una.
- ❖ Identifique otros productos que se puedan desarrollar en este ciclo y estime sus tamaños (documentos referentes a los requerimientos, las páginas de diseño de alto nivel, documentación para el usuario, materiales de prueba, código, etc.).

Producción del plan de calidad

A continuación se describen los pasos para realizar el Plan de Calidad, las secciones del plan se describen a detalle mas adelante, en este capítulo.

1. Estime cuántos defectos se introducen aproximadamente en cada fase. Base la estimación en el tiempo dedicado en cada fase multiplicándolo por la tasa de inyección de defectos
2. Estime la tasa de eliminación de defectos que el equipo aplicará en cada fase.
3. Establezca los objetivos para las tasas en la fase de pruebas de acuerdo a los defectos encontrados, tomando en cuenta el criterio de la tabla mencionada anteriormente.
4. Genere el plan de calidad.
5. Compare los valores del plan de calidad con los que integra la tabla Criterio estándar de calidad.
6. Si las tasas de eliminación de defectos y revisiones e inspecciones no están cercanas al criterio, ajuste el tiempo dedicado a la parte de eliminación de defectos o modifique las tasas de eliminación de defectos hasta que los valores estén cercanos.
7. Si las tasas de defectos son menores, revise los datos para asegurarse que todos los criterios se cumplieron.
8. Si los valores de los defectos/KLOC son altos en la compilación o en pruebas revíselos hasta que estén dentro del rango. También revise las tasas para las revisiones e inspecciones; si son menores increméntelas hasta que sean las adecuadas. Posiblemente, también se deberá incrementar el tiempo asignado a revisiones o inspecciones para mantener las tasas de eliminación de defectos dentro del rango.

9. Repita los pasos desde el número 4 hasta el 8 hasta que el plan de calidad cumpla con los objetivos del equipo.

Proporciones de resumen

Las tres medidas de resumen proporcionan una perspectiva completa en la calidad del proceso. La medida de LOC/hr mide la productividad completa del equipo, un número mayor indica una productividad mayor y costos menores. El porcentaje de reuso mide el porcentaje de las LOC totales del producto que fueron reutilizadas a partir de productos previamente desarrollados. Similarmente, el porcentaje de reuso nuevo mide la contribución del ciclo en la mejora de productividad para ciclos o proyectos futuros.

Porcentaje libre de defectos (PLD)

Mide el porcentaje de los componentes de un producto que no tuvieron defectos en una fase determinada. Por ejemplo, si un ensamble estuvo conformado por cinco partes y cuatro de ellas tuvieron defectos de compilación en el primer ciclo, tal ensamble tuvo 20% libre de defectos en la compilación. Los productos de calidad alta deberían alcanzar o exceder un 90% en las pruebas del sistema.

Defectos por página

Esta medida muestra el número promedio de defectos eliminados en cada página de los documentos de requerimientos y de diseño de alto nivel.

Defectos por KLOC

Un defecto es cualquier elemento que puede afectar la buena calidad de los productos. Un defecto mayor es cualquier problema que al momento de arreglarse origina cambios en el programa ejecutable.

El número de defectos por KLOC encontrados en la fase de pruebas indica la calidad del producto que entra y sale de esa fase. Cuando un producto tiene bastantes defectos al momento de realizar las pruebas se encontrarán muchos de ellos, pero también habrá otros que quedarán si detectarse.

La experiencia ha mostrado que cuando los productos tienen menos de 0.5 defectos/KLOC en las pruebas de construcción e integración y menos de 0.2 defectos/KLOC en las pruebas del sistema los usuarios no llegan a encontrar tales defectos.

Proporciones de defectos

Proporciona un panorama acerca de la calidad en las revisiones del diseño y código. Por ejemplo, la experiencia muestra que cuando los ingenieros encuentran dos veces los mismos defectos tanto en la revisión de código como en la compilación, se dice que han hecho una buena revisión de código. Esto daría una tasa de defectos en la revisión/compilación de código de 2.0.

Tasas en tiempo de desarrollo

Otra manera de evaluar un proceso es identificar el tiempo permanecido en cada fase de desarrollo. Generalmente, cuando los ingenieros permanecen mayor tiempo en el diseño detallado que en la codificación, la experiencia muestra que producen un diseño de calidad; y cuando pasan más del 50% del tiempo que dedicaron al diseño detallado en su propia revisión, de acuerdo a la experiencia, se dice que la revisión de diseño fue esmerada.

Para los requerimientos, los datos son menos claros, pero se aconseja dedicar 25% o más del tiempo que se utilizó en el desarrollo de los mismos para su revisión. Para el diseño de alto nivel se aconseja dedicar 50% del tiempo que se utilizó para el desarrollo (de tal diseño) en las revisiones e inspecciones. También la tasa que se aconseja aplicar para la revisión del código de acuerdo al tiempo de codificación dedicado debe ser mayor al 50%.

Cuando los ingenieros no tienen mucha experiencia en un lenguaje de programación es mejor utilizar los valores de defectos/KLOC obtenidos a partir de la compilación y pruebas unitarias como guías indicadoras de calidad.

Actividades de Validación de fallas (A/FR)

Otra tasa de tiempo de desarrollo útil es A/FR, la cual es una tasa de validación de fallas. Se refiere a la proporción de tiempo dedicado a las actividades de tipo-validación (tales como revisiones e inspecciones) hasta el tiempo dedicado a las actividades en las que se encuentran fallas (tales como compilación y pruebas). Se aconseja que el A/FR sea mayor a 2.0 para programas solamente estándares. Para programas mayores como los producidos por TSPI, se considera adecuado 1.0. La razón es que los productos más complejos y grandes necesitan mayor tiempo de pruebas aún cuando tengan pocos o ningún defecto.

Tasas en revisión e inspección

Para llevar a cabo una alta calidad se debe hacer una planeación que permita permanecer tiempo suficiente en las revisiones e inspecciones. Se recomienda utilizar las siguientes líneas guía:

- ❖ Revisiones e inspecciones en requerimientos: <2.0 páginas por hora cuyo texto tendrá interlineado sencillo.
- ❖ Revisiones e inspecciones en diseño de alto nivel: <5 páginas por hora de diseño.
- ❖ Revisiones e inspecciones en diseño detallado: <100 líneas por hora en pseudocódigo.
- ❖ Revisiones e inspecciones de código fuente: <200 LOC/hora.

El tiempo de inspección total es igual a la suma de los tiempos dedicados en la preparación de las inspecciones más el tiempo que se utilizó en la reunión, multiplicado por el número de ingenieros que participaron.

Al momento de hacer revisiones no se asegura que se encontrarán defectos, por eso es necesario utilizar métodos de revisión efectivos.

Tasas en inyección de defectos

Al tener datos sobre las tasas de inyección de defectos se tiene una base para estimar cuántos defectos se inyectarán al momento de programar, esto se refiere al número de defectos que se introducen en una medida de tiempo, por ejemplo, dos defectos por hora se introducen en el diseño detallado y seis defectos por hora en la programación. Sólo será necesario multiplicar la tasa de inyección de defectos esperada por el tiempo estimado que se dedicará a cada fase. Tal cálculo permitirá obtener una estimación del número posible de defectos que, generalmente, se pueden inyectar.

Tasas de eliminación de defectos

Permite calcular las tasas de eliminación de defectos al momento de programar. Aunque aquí los datos varían, eliminando desde cero hasta veinte defectos por hora; pero se ha encontrado que la tasa más común consiste en eliminar seis defectos por hora. También en las revisiones del diseño los datos varían, generalmente se eliminarían dos o más defectos por hora en los programas finales.

Producción por fase

En TSPi, la producción por fase se refiere al porcentaje de defectos de un programa que se eliminaron en una fase dada.

Si se tuvieron 19 defectos en un programa al realizar la revisión de código y se introdujo 1 defecto durante esa revisión, y se encontraron 15 de esos defectos en la revisión, se tendrá un 75% de producción en la revisión de código. Esto es:

$$\text{Producción} = 100 * (\text{defectos encontrados}) / (\text{defectos del producto}) = \\ 100 * 15 / (19 + 1) = 75\%$$

Esta misma fórmula se puede aplicar a cualquier fase del proceso. Se debe calcular la producción por fase, hasta que el programa se haya terminado, probado y utilizado por un período determinado; porque en caso contrario faltarían defectos por encontrar y con ello la producción sería menor.

Producción del proceso

La producción del proceso mide el porcentaje de defectos eliminados antes de determinada fase. Por ejemplo, la producción del proceso durante o antes de la compilación mide el porcentaje de defectos inyectados antes de la compilación y que fueron eliminados antes de la misma. TSPi tiene como objetivo eliminar el 97.5% o más defectos antes de la construcción e integración, y eliminar más del 99% antes de la prueba del sistema. Así se producirán productos de alta calidad. Una manera de alcanzar tales porcentajes sería obtener un 75% antes de la primera compilación y 85% antes de la primera prueba unitaria.

Para alcanzar esos objetivos también es importante que los integrantes del equipo actualicen sus listas de verificación de revisión, hagan revisiones personales esmeradas e inspecciones en los requerimientos, el diseño de alto nivel, el diseño detallado y el código.

Partes de calidad baja

Cuando los datos obtenidos en calidad no son óptimos de acuerdo al estándar definido, localice las partes que tengan un alto número de defectos, tasas y producciones bajas. Después revise el diseño y el código fuente para detectar el origen del problema. Generalmente, pudo haber sucedido por haber realizado un diseño pobremente documentado; o que la codificación, revisiones e inspecciones hayan sido realizadas sin cumplir con las indicaciones establecidas. Cuando se detecten los problemas, y que generalmente sean los mismos, se arreglan antes de la construcción, integración y pruebas del sistema.

Criterio de calidad

El criterio de esta sección se resume en la tabla que contiene el criterio estándar de calidad. Se utiliza la primera vez que se aplica el proceso TSPi. Después de haber terminado el primer ciclo del proyecto y haber adquirido cierta experiencia se podrán hacer modificaciones a los valores del criterio; aunque lo más recomendable sería después de haber aplicado dos o más ciclos. Todos los criterios deben documentarse en la carpeta y en el ciclo correspondiente.

Forma RESTM

Tiene como propósito resumir los datos para el tamaño del producto.

- ❖ El nivel más bajo resume los datos sobre el tamaño de cada parte.
- ❖ Los niveles más altos resumen los datos de tamaño para un ensamble y sus partes.

- ❖ Se utiliza esta forma para documentar el tamaño de las partes de un ensamble.

1.1.6.4 Fase Requerimientos del proceso de software

El Objetivo de esa fase es describir el proceso de requerimientos en TSPi. Explicar qué son, por qué son necesarios y discutir las características más importantes.

¿Qué son los requerimientos?

En esta fase, el equipo genera la Especificación de Requerimientos de Software (ERS). En tal documento se hace una descripción clara de lo que será el producto, deberá incluir el criterio preciso para evaluarlo cuando esté terminado y asegurar que las funcionalidades sean las correctas. También proporciona retroalimentación al cliente acerca de lo que se pretende construir.

Se debe saber exactamente lo que el producto debe hacer antes de construirlo. En el inicio de esta fase se realiza la definición de necesidades por parte del cliente, si existieran necesidades que no son claras se debe redactar un documento con preguntas que hagan referencia a todas aquellas dudas para obtener una respuesta posteriormente.

Cambios en los requerimientos

Con frecuencia se suele hacer cambios a los requerimientos; generalmente, los usuarios no pueden saber lo que necesitan en forma precisa hasta que tratan de usar el producto terminado. Sin embargo, cuando se introduce un sistema nuevo se cambia la manera en que ellos trabajan. Conforme el trabajo de desarrollo progresa empiezan a apreciar la manera en cómo afectará su ambiente, mientras más claro es eso para los usuarios, pensarán en nuevas funciones y características, por esto, los requerimientos constantemente cambian hasta que se congelan en un producto.

Extracción de los requerimientos

Generalmente, en proyectos de software industriales el primer paso en el proceso de requerimientos es la extracción de los mismos. En este momento se interroga a los clientes, usuarios u otros involucrados importantes para descubrir lo que necesitan realmente.

Los pasos principales de la extracción de los requerimientos son los siguientes:

- ❖ Evaluar la factibilidad del sistema.
- ❖ Conocer y comprender los asuntos organizacionales.
- ❖ Identificar a todos los involucrados en el proyecto.

- ❖ Registrar las fuentes de los requerimientos.
- ❖ Definir el ambiente operativo del sistema.
- ❖ Evaluar asuntos del negocio.
- ❖ Definir las restricciones del dominio.
- ❖ Registrar la razón de los requerimientos.
- ❖ Realizar un primer prototipo de los requerimientos (probablemente pobremente entendido).
- ❖ Definir los escenarios de uso.
- ❖ Definir los procesos operacionales.

Especificación de Requerimientos de Software (ERS)

En este documento el equipo describe las funciones que se planean desarrollar y cómo se pretende generar el producto, evitando describir la implementación, puesto que es una responsabilidad de la fase de diseño.

- ❖ Requerimientos funcionales: entradas, salidas, procesos y casos de uso.
- ❖ Requerimientos de interfaz externos: usuario, hardware, software y comunicaciones.
- ❖ Restricciones de diseño: formatos de archivo, lenguajes, estándares, compatibilidad, etc.
- ❖ Atributos: disponibilidad, seguridad, mantenimiento, conversión, etc.
- ❖ Otros requerimientos: base de datos, instalación, etc.

1.1.6.5 Fase Diseño del proceso de software

Esta fase tiene como principal objetivo proporcionar un diseño completo y de alta calidad que se utilice como base para la fase de Implementación.

Se enfoca en los principios de diseño y su proceso; los cuales se pueden utilizar con cualquier método de diseño que parezca más apropiado. En TSPi, la fase de diseño se enfoca a la estructura completa del sistema.

En equipo se puede trabajar más rápido dividiendo el producto en partes componentes y que cada uno de los integrantes diseñe e implementen una o más de esas partes.

Principios de diseño

El diseño es el proceso creativo mediante el cual se decide cómo construir un producto, debe contener una especificación completa y precisa de la construcción del producto. Un diseño completo define las partes principales de un producto, describe cómo esas partes interactúan y especifica cómo unirlas para producir el resultado final.

El diseño de alto nivel difiere del diseño detallado y de la implementación sólo en el alcance y detalle. El diseño de alto nivel (DAN) tiene que producir una especificación que varios ingenieros puedan usar para que eso les permita diseñar cada parte independientemente de los demás.

Cuando el diseño de alto nivel es completo y preciso, los ingenieros rápidamente pueden producir los diseños detallados de los componentes; para eso, necesitan conocer las especificaciones funcionales completas de cada componente, sus interfaces y comportamientos de estado. Entonces, para generar el producto final, los ingenieros de implementación necesitan un diseño detallado que defina la estructura lógica, las condiciones de inicialización y de cada paso, la estructura de estado detallada y las transiciones de estado para cada programa. Entonces, los ingenieros de implementación producen el código que implementa el diseño.

Su objetivo es que el código fuente realice correctamente todas las funciones especificadas, que utilice adecuadamente todas las facilidades del sistema, que incorpore las funciones de reuso disponibles y que siga todos los estándares de codificación y del sistema y sus convenciones. El producto final implementado tiene que ser un programa fuente que se compile y ejecute sin errores o problemas.

Diseñando en equipo

Un problema común en el diseño de sistemas grandes de software es la necesidad de definir la estructura completa del sistema antes de especificar cualquier otra cosa.

Se recomienda que todo los integrantes del equipo colaboren en el diseño de estructura completo, también que se identifiquen tareas que el resto de los integrantes puedan hacer, mientras uno o dos de los ingenieros definen esa estructura y especifican los componentes del sistema con suficiente detalle para poder obtener un diseño especificado completamente.

El problema más crítico en equipos de trabajo es la contribución de todos sus integrantes. Cuando se trabaja en equipo, en ocasiones existe un rechazo a ofrecer sugerencias e ideas; principalmente, en equipos de software este es un gran problema porque las decisiones de diseño mayores tienen que establecerse desde un principio.

Cada integrante debería contribuir con su experiencia y conocimiento. La persona que dirige la reunión en equipo debería hacer pequeñas pausas para preguntar si alguien tiene ideas o conocimientos sobre los temas discutidos. Los equipos que siguen estos consejos son más productivos.

Estándares de diseño

Existen varios estándares de diseño, de los cuales se consideraron importantes los siguientes:

❖ **Convenciones en nombramiento**

Es el primer estándar en el cual se especifica la estructura de nombramiento. El AA debe establecer un glosario del sistema. Se deben definir los nombres de tipos de programa jerárquicos (tales como sistema, producto, componente, módulo u objeto), las convenciones usadas en nombres de programas, archivos, variables y parámetros; y los procedimientos para establecer, controlar y cambiar nombres.

❖ **Formatos de interfaz**

Aquí se definen los formatos y el contenido de las interfaces de los componentes. Esto consiste en indicar parámetros para variables, códigos de error u otras condiciones. Cuando las interfaces están especificadas consistentemente se tendrán menos errores y se localizarán rápidamente durante las revisiones e inspecciones.

❖ **Mensajes del sistema y de error**

Es conveniente establecer formatos estándares y procedimientos para mensajes de error y del sistema. Un sistema útil debe tener consistencia y los mensajes deben ser comprendidos fácilmente.

❖ **Estándares de representación de diseño**

El estándar de representación del diseño define el producto que se generará a partir del diseño. Es importante definir y utilizar esos estándares porque una representación de diseño imprecisa o ambigua puede dirigir a problemas graves en la implementación y en las pruebas.

Diseñando para reuso

Para aprovechar el tiempo disponible de los integrantes del equipo durante el diseño de alto nivel se pueden identificar funciones comunes y proponer un conjunto inicial de partes de reuso.

El reuso es una fuente poderosa para incrementar la productividad del equipo. Permite ahorrar tiempo en el primer ciclo de desarrollo y en ciclos posteriores. En proyectos grandes se puede aplicar reuso a partir del desarrollo de requerimientos y de la estrategia. Los resultados principales que se obtienen son la definición en los estándares de interfaces y en las convenciones de nombramiento, para generar estándares de documentación, generar productos de alta calidad y proporcionar apoyo en la aplicación.

Los estándares de documentación marcan la diferencia entre las partes de reuso y las que no lo son. Un buen reuso permite que los ingenieros puedan utilizar un programa sin tener que analizar el código fuente. Por eso, las partes de reuso deben contener una especificación completa del comportamiento

externo, también es buena idea añadir una sección de comentarios al principio del programa fuente en cada parte.

1.1.6.6 Fase Implementación del proceso de software

Esta fase tiene como objetivo describir el proceso de implementación a través del criterio de diseño, estándares de implementación, estrategias de implementación, revisiones e inspecciones.

Estándares de implementación

Con unos cuantos minutos que el equipo dedique a la definición de estándares al inicio del proyecto ahorrará gran cantidad de tiempo. Primero, todos los integrantes deben estar de acuerdo en los estándares necesarios y en su contenido y pedir a uno o dos de los integrantes que realicen el desarrollo de los mismos. El ACP dirige las actividades de definición de estándares.

Revisión de estándares

Se debe revisar lo siguiente:

- ❖ Los estándares de nombres, de interfaces y de mensajes desarrollados durante la fase de diseño para asegurarse de que son apropiados y están siendo utilizados.
- ❖ La lista de rutinas de reuso para verificar que estén terminadas y que todos los integrantes del equipo las estén utilizando.
- ❖ El glosario de nombres para asegurarse que cada quien está usando el mismo nombre para el mismo producto y que todos los nombres nuevos se van agregando al glosario.
- ❖ Los nombres de componentes, los nombres de las variables compartidas, de los parámetros y de los archivos para verificar su consistencia.
- ❖ Las interfaces y los mensajes estándares para asegurarse de que ya han sido definidos, registrados en el glosario de nombres y que son conocidos y utilizados por los integrantes del equipo.

Estándares de codificación

Un estándar de codificación común asegura la consistencia en el código del equipo; eso facilita la realización de inspecciones, permite que se hagan de una manera más rápida y que sean efectivas. También facilita que los integrantes del equipo puedan compartir su código.

En un estándar de codificación bien estructurado se define una sección para comentarios, lo cual permite mayor flexibilidad al momento de realizar

incrementos en ciclos subsecuentes.

Con estos estándares existe la posibilidad de considerar el reuso en un sistema, lo cual permite ahorro de tiempo para el diseño y la implementación.

1.1.6.7 Fase Pruebas del proceso de software

En ésta fase se pretende cumplir el siguiente objetivo: integrar las pruebas y la documentación de las mismas. Indicar los objetivos, las estrategias y planeación de las pruebas.

En TSPi, el propósito de las pruebas consiste en evaluar el producto y no en modificarlo. Los defectos debieron haberse detectado y arreglado antes de la fase de pruebas. La calidad de un producto se determina cuando se desarrolla.

Estrategia de pruebas de TSPi

TSPi tiene como objetivo verificar que los productos a los que se les aplican las pruebas son de alta calidad. Las actividades principales son:

- ❖ Utilizar las partes desarrolladas y a las que se les ha aplicado pruebas unitarias para construir el sistema.
- ❖ Utilizar las pruebas de integración para verificar que el sistema está debidamente construido, que todas las partes se encuentran desarrolladas y que funcionan en conjunto.
- ❖ Validar con las pruebas del sistema que el producto realiza los requerimientos del sistema.

Paralelamente se deben:

- ❖ Identificar módulos o componentes de baja calidad y entregarlos al ACP para su evaluación y depuración.
- ❖ Identificar componentes de baja calidad que estén causando problemas, aún después de que se han depurado; regresarlos al ACP para que trabaje de nuevo con ellos o los reemplace.

Las pruebas de integración aseguran que todas las partes se hayan incluido en el sistema y que las interfaces trabajen en conjunto. Finalmente, las pruebas del sistema, validan las funciones y desempeño contra los requerimientos.

Planeación de las pruebas

Se realiza en diversos momentos de acuerdo a TSPi. El plan de pruebas describe las pruebas que se planean aplicar, el orden de las mismas y los materiales necesarios. Con el plan completo se debe mostrar la manera en la que cada requerimiento será probado y cómo cada escenario o guión cubre las áreas de los requerimientos. También se debe saber que áreas serán probadas y cuales tendrán un amplio cubrimiento. Además se deberá nombrar a cada prueba por anticipado, estimar los defectos encontrados en cada fase, el tiempo total en la detección de defectos y el tiempo de pruebas total. Después

se deberán estimar los materiales requeridos.

Al final de la planeación de las pruebas se deberá tener:

- ❖ Una lista de todos los pasos que se van a desarrollar en las pruebas.
- ❖ Los materiales requeridos para cada una.
- ❖ Los resultados que generarán.
- ❖ Estimación del tiempo de ejecución libre de defectos, de los defectos que serán encontrados y del tiempo total para cada prueba.
- ❖ Estimación del trabajo requerido en el plan de pruebas.

También será necesario tener:

1. Lista de los materiales requeridos y cuales se utilizarán en cada prueba.
2. Los objetivos de cada prueba.
3. Las características de los materiales y su tiempo de desarrollo.
4. El responsable de desarrollar cada material y la fecha final de terminación.

En la Forma Registro de Pruebas (REGPR) se deberá indicar:

1. La fecha de ejecución de la prueba.
2. El nombre de quien aplica la prueba.
3. Nombre o número de las pruebas.
4. El producto y configuración que se está probando.
5. La hora de inicio.
6. La hora de terminación.
7. El número de defectos encontrados, de acuerdo a los datos en REGD.
8. Los resultados de las pruebas.
9. La configuración del sistema que se está probando.
10. Herramientas o facilidades utilizadas.
11. Si se requirió la intervención de algún operador y cuánto tiempo.

Documentación

El número de personas asignadas para las pruebas y la documentación varían dependiendo del contenido funcional y de calidad del producto. Es buena idea, que durante el primer ciclo de desarrollo se asignen más ingenieros en las pruebas y en ciclos posteriores, se incremente el número de ingenieros que se asignen a la documentación. El desarrollo de pruebas debería reducirse en ciclos posteriores y la carga de trabajo de la documentación debería aumentar en cada funcionalidad que se incremente.

La importancia de la documentación

La documentación es parte esencial de cada producto de software. Es más importante que el código del programa. TSPi incluye el trabajo de generación de documentación en la fase de pruebas. En grandes sistemas, debería iniciar antes y continuar durante las pruebas.

1.1.6.8 Fase Postmortem del proceso de software

El Postmortem es el último paso en el proceso de TSPi. Aquí se revisa que el equipo de trabajo haya terminado todas las tareas y registrado todos los datos requeridos. Proporciona una manera estructurada de aprendizaje y mejora, porque se evalúa el desempeño personal y del equipo. Cada ciclo de desarrollo de TSPi finaliza con un postmortem.

En esta fase se genera el reporte de Postmortem, cada integrante describe las actividades del ciclo, las compara con lo que se planeó y con ciclos anteriores. Se evalúan los datos del proyecto y se realiza una revisión y crítica sobre roles.

Reporte del ciclo

En este reporte se describe lo que se produjo, el proceso utilizado y los roles desempeñados; lo que sí se pudo aplicar y lo que no fue conveniente, para ello se pueden hacer sugerencias que puedan aplicarse la siguiente vez. Se deberá describir el desempeño que cada integrante tuvo respecto al rol que desempeñó, de acuerdo a los indicados por TSPi, así como el rol correspondiente al área de desarrollo. Si es posible, se justificarán las conclusiones con datos actuales del equipo y se comparará el desempeño en ciclos anteriores. Tal reporte debe ser breve y objetivo.

El reporte del ciclo deberá tener el siguiente contenido:

1. Índice.
2. Resumen.
3. Reporte de roles, en cuanto a:
 - a) Liderazgo.
 - b) Desarrollo.
 - c) Planeación.
 - d) Proceso.
 - e) Calidad.
 - f) Apoyo o soporte.
4. Reporte de ingenieros.

Reportes de ingenieros

Cada ingeniero debe describir el desempeño personal de acuerdo a lo planeado y la calidad del trabajo que realizó. Es importante indicar cómo mejorar el trabajo para la siguiente vez.

1.2 Proceso Personal de Software (PSP)

“Esta metodología es una herramienta de gran ayuda para los ingenieros de software debido a que se puede planear la realización del software por medio de un programa de trabajo. El resultado de esto será un producto de calidad”⁵.

Es importante tener una lógica de la administración del tiempo.

- ❖ Cuanto tiempo se dedica a la semana en curso y cuanto se le dedicará a la siguiente semana.
- ❖ Verificar la exactitud del tiempo empleado y planes estimados durante la semana.
- ❖ Es necesario administrar el tiempo, planearlo y seguir el programa establecido.

Los siguientes pasos nos ayudan a tener una mejor administración del tiempo:

- ❖ Categorizar las actividades principales.
- ❖ Registrar el tiempo a cada una de estas actividades.
- ❖ Almacenar los datos recabados en formatos especiales para este propósito.

1.2.1 Cuaderno del ingeniero

Es de gran utilidad para llevar un registro completo de las actividades relacionadas con el desarrollo de un proyecto de software.

Es necesario clasificar las actividades dentro de las categorías principales, de una manera estándar, registrar el tiempo que se dedicó a cada actividad, guardando los datos en el cuaderno del ingeniero.

Partes que componen el Cuaderno del Ingeniero:

- ❖ Carátula.
- ❖ Índice.
- ❖ Registro de tiempo.
- ❖ Resumen semanal de actividades.
- ❖ Registro del número de trabajo.
- ❖ Registro de defectos.

A continuación se describen los formatos principales que conforman éste cuaderno:

⁵ Watts S. Humphrey Introduction to the Personal Software Process Pág.12

❖ Carátula

En la parte superior derecha se tiene el número del cuaderno.

El título del cuaderno es "Cuaderno del Ingeniero".

A continuación se anota el nombre de la compañía o la universidad correspondiente.

También se etiqueta con el nombre del ingeniero, teléfono y/o la dirección de correo electrónico.

En la parte inferior de la carátula se etiqueta con la fecha de inicio del libro y la fecha en la que se ha concluido el llenado del cuaderno.

En las dos primeras hojas del cuaderno se incluye un índice de contenido con los siguientes datos:

Número de la página, tema y fecha en que se inicia dicho tema.

A continuación se muestra la figura correspondiente a la carátula:

❖ Registro del tiempo

Se conforma por un encabezado el cual se compone por nombre del ingeniero, la fecha en que se inició el registro, nombre del instructor y el nombre y número de clase.

Las columnas de cuerpo de la forma de la manera siguiente:

Fecha.- Es la fecha en que da comienzo la actividad.

Inicio.- Es la hora en que se da comienzo la actividad.

Fin.- Es la hora en que finaliza la actividad.

Interrupción.- Es el tiempo que se toma a partir de que se genera una interrupción de la actividad en curso.

Tiempo delta.- Este tiempo es la diferencia en minutos entre la hora de inicio y la hora de terminación de la actividad; a esta diferencia se le resta el tiempo de la interrupción en caso de que ésta exista.

Actividad.- Se nombra la actividad a desarrollar.

Comentarios.- En esta sección se especifica con más detalle lo que ocurrió en cada actividad, el tipo de interrupción que se generó, etc.

C.- Se registra esta columna cuando se ha concluido la actividad.

U.- Se registra el número de unidades de una actividad que ya está concluida.

❖ Resumen semanal de actividades

Esta tabla está muy relacionada con el registro de tiempo antes mencionado. El propósito de este registro es dar seguimiento y analizar el tiempo dedicado al proyecto.

❖ Tabla de Registro del número de trabajo

El propósito de esta tabla es registrar el número de trabajo o tarea para cada proyecto, además de la información clave de cada trabajo.

Esta tabla cuenta con un encabezado que contiene el nombre del ingeniero y la fecha en que se comienza el registro.

Trabajo.- Se numera cada uno de los trabajos.

Fecha.- Se anota la fecha de inicio de cada trabajo.

Proceso.- Se introduce el tipo de tarea.

Tiempo estimado.- Se anota el tiempo en minutos que se estima para cada tarea.

Unidades estimadas.- Se anota el número de unidades que se estiman para cada tarea.

Tiempo actual.- Se anota el tiempo total final de la tarea.

Unidades actuales.- Se introduce el número actual final del total de unidades.

Promedio actual.- Se calcula dividiendo el tiempo actual por las unidades actuales.

Tiempo final.- Se suma el tiempo final de la tarea del mismo tipo con el tiempo actual por las unidades actuales.

Unidades finales.- Se anotan las unidades completas por cada tipo.

Promedio final.- Se divide el tiempo final por las unidades finales.

Máximo final.- Se hace un comparativo para todas las tareas del mismo tipo que ya están terminadas y se obtiene el valor máximo.

Mínimo final.- Se hace un comparativo para todas las tareas del mismo tipo que ya están terminadas y se obtiene el valor mínimo.

A continuación se muestra la forma correspondiente a la Tabla de Registro del número de trabajo:

❖ Registro de Defectos

Esta forma está compuesta por un encabezado que contiene el nombre del ingeniero, fecha de registro programa y nombre del instructor.

Y en su contenido posee los siguientes aspectos: fecha de registro del defecto, número de defecto, tipo, aportación, reparar, reposición de tiempo y reparación del defecto y una descripción breve del defecto encontrado.

Además se mencionan los tipos de defectos que pudiesen encontrar tales como: documentación, sintaxis, estructura del paquete, asignación, interfaz, comprobación, datos, función, sistema y ambiente.

2. PROGRAMACIÓN ORIENTADA A OBJETOS

2.1 Programación Orientada a Objetos

Es una técnica de programación que aumenta la velocidad de desarrollo de los programas y hace que su mantenimiento sea más fácil al volver a utilizar "objetos" que tienen comportamientos, características, y relaciones asociadas con el programa. Los objetos son organizados en grupos que están disponibles para la creación y mantenimiento de aplicaciones.

Tiene tres características básicas: debe estar basado en objetos, basado en clases y capaz de tener herencia de clases.

Definición de objeto

"Un objeto es una instancia de una clase. Puede ser identificado por medio de un nombre y un estado, el cual está definido por los valores de sus atributos en un momento en particular"¹.

Estructura de un objeto

Un objeto puede considerarse como una especie de cápsula dividida en tres partes:

- ❖ Relaciones
- ❖ Propiedades
- ❖ Métodos

Cada uno de estos componentes desempeña un papel totalmente independiente:

Las relaciones permiten que el objeto se inserte en la organización y están formadas esencialmente por punteros a otros objetos.

Las propiedades distinguen un objeto determinado de los restantes que forman parte de la misma organización y tiene valores que dependen de la propiedad de que se trate. Las propiedades de un objeto pueden ser heredadas a sus descendientes en la organización.

Los métodos son las operaciones que pueden realizarse sobre el objeto, que normalmente estarán incorporados en forma de programas (código) que el objeto es capaz de ejecutar y que también pone a disposición de sus descendientes a través de la herencia.

¹ <http://www.monografias.com/trabajos/objetos/objetos.shtml>

Encapsulamiento y ocultación

Cada objeto es una estructura compleja en cuyo interior hay datos y programas, todos ellos relacionados entre sí, como si estuvieran encerrados conjuntamente en una cápsula. Esta propiedad (encapsulamiento), es una de las características fundamentales en la Programación orientada a objetos.

Los objetos son inaccesibles, e impiden que otros objetos, los usuarios, o incluso los programadores conozcan cómo está distribuida la información o qué información hay disponible. Esta propiedad de los objetos se denomina ocultación de la información.

Esto no quiere decir, sin embargo, que sea imposible conocer lo necesario respecto a un objeto y a lo que contiene. Si así fuera no se podría hacer gran cosa con él. Lo que sucede es que las peticiones de información a un objeto, deben realizarse a través de mensajes dirigidos a él, con la orden de realizar la operación pertinente.

El hecho de que cada objeto sea una cápsula facilita enormemente que un objeto determinado pueda ser transportado a otro punto de la organización, o incluso a otra organización totalmente diferente que precise de él. Si el objeto ha sido bien construido, sus métodos seguirán funcionando en el nuevo entorno sin problemas. Esta cualidad hace que la Programación orientada a objetos sea muy apta para la reutilización de programas.

Organización de los objetos

Los objetos forman siempre una organización jerárquica, en el sentido de que ciertos objetos son superiores a otros de cierto modo.

Existen varios tipos de jerarquías: serán simples cuando su estructura pueda ser representada por medio de un "árbol". En otros casos puede ser más compleja.

En cualquier caso, sea la estructura simple o compleja, podrán distinguirse en ella tres niveles de objetos.

- ❖ La raíz de la jerarquía. Se trata de un objeto único y especial. Este se caracteriza por estar en el nivel más alto de la estructura y suele recibir un nombre muy genérico, que indica su categoría especial, como por ejemplo objeto madre, raíz o entidad.
- ❖ Los objetos intermedios. Son aquellos que descienden directamente de la raíz y que a su vez tienen descendientes. Representan conjuntos o clases de objetos, que pueden ser muy generales o muy especializados, según la aplicación. Normalmente reciben nombres genéricos que denotan al conjunto de objetos que representan, por ejemplo, ventana, cuenta, fichero. En un conjunto reciben el nombre de clases o tipos si descienden de otra clase o subclase.

- ❖ Los objetos terminales. Son todos aquellos que descienden de una clase o subclase y no tienen descendientes. Suelen llamarse casos particulares, instancias o ítems porque representan los elementos del conjunto representado por la clase o subclase a la que pertenecen.

Polimorfismo

Una de las características fundamentales de la programación orientada a objetos es el polimorfismo, que no es otra cosa que la posibilidad de construir varios métodos con el mismo nombre, pero con relación a la clase a la que pertenece cada uno, con comportamientos diferentes. Esto conlleva la habilidad de enviar un mismo mensaje a objetos de clases diferentes. Estos objetos recibirían el mismo mensaje global pero responderían a él de formas diferentes; por ejemplo, un mensaje "+" a un objeto entero significaría suma, mientras que para un objeto cadena significaría concatenación.

2.2 Lenguaje Unificado de Modelado (UML)

El Lenguaje Unificado de Modelado (UML - Unified Modeling Language) es "un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software"². UML entrega una forma de modelar cosas conceptuales como lo son procesos de negocio y funciones de sistema, además de cosas concretas como lo son escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reutilizables.

UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real.

2.2.1 Modelado de Clases

Un diagrama de clases sirve para visualizar las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, de herencia, de uso y de contenido.

Un diagrama de clases esta compuesto por los siguientes elementos:

- ❖ Clase: atributos y métodos
- ❖ Relaciones: Herencia, Composición, Agregación, Asociación y Uso.

² Ibid

2.2.1.1 Elementos del modelado de clases

Clase

"Es la unidad básica que encapsula toda la información de un Objeto (un objeto es una instancia de una clase)"³. A través de ella podemos modelar el entorno en estudio. En UML, una clase es representada por un rectángulo que posee tres divisiones, como se muestra en la siguiente figura:

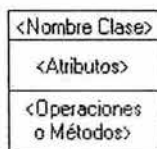


Figura 2.1 Clase


En donde:

- ❖ Superior: Contiene el nombre de la Clase
- ❖ Intermedio: Contiene los atributos (o variables de instancia) que caracterizan a la Clase (pueden ser private, protected o public).
- ❖ Inferior: Contiene los métodos u operaciones, los cuales son la forma como interactúa el objeto con su entorno (dependiendo de la visibilidad: private, protected o public).

Atributos y Métodos

Atributos




Los atributos o características de una Clase pueden ser de tres tipos, los que definen el grado de comunicación y visibilidad de ellos con el entorno, estos son:

- ❖ public (+, 

³ <http://www.dcc.uchile.cl/~psalinas/uml/introduccion.html>

Métodos

Los métodos u operaciones de una clase son la forma en como ésta interactúa con su entorno, éstos pueden tener las características:

- ❖ **public (+, **): Indica que el método será visible tanto dentro como fuera de la clase, es decir, es accesible desde todos lados.
- ❖ **private (-, **): Indica que el método sólo será accesible desde dentro de la clase (sólo otros métodos de la clase lo pueden acceder).
- ❖ **protected (#, **): Indica que el método no será accesible desde fuera de la clase, pero sí podrá ser accesado por métodos de la clase además de métodos de las subclases que se deriven (ver herencia).

2.2.1.2 Relaciones entre Clases

Ahora ya definido el concepto de Clase, es necesario explicar como se pueden interrelacionar dos o más clases (cada uno con características y objetivos diferentes).

Antes es necesario explicar el concepto de cardinalidad de relaciones: En UML, la cardinalidad de las relaciones indica el grado y nivel de dependencia, se anotan en cada extremo de la relación y éstas pueden ser:

- ❖ uno o muchos: 1..* (1..n)
- ❖ 0 o muchos: 0..* (0..n)
- ❖ número fijo: m (m denota el número).

Herencia (Especialización/Generalización)

Indica que una subclase hereda los métodos y atributos especificados por una Super Clase, por ende la Subclase además de poseer sus propios métodos y atributos, poseerá las características y atributos visibles de la Super Clase (public y protected).



Figura 2.2 Notación Herencia

Agregación

Para modelar objetos complejos, no bastan los tipos de datos básicos que proveen los lenguajes: enteros, reales y secuencias de caracteres. Cuando se requiere componer objetos que son instancias de clases definidas por el desarrollador de la aplicación, tenemos dos posibilidades:

- ❖ **Por Valor:** Es un tipo de relación estática, en donde el tiempo de vida del objeto incluido esta condicionado por el tiempo de vida del que lo incluye. Este tipo de relación es comúnmente llamada Composición (el Objeto base se construye a partir del objeto incluido, es decir, es "parte/todo").
- ❖ **Por Referencia:** Es un tipo de relación dinámica, en donde el tiempo de vida del objeto incluido es independiente del que lo incluye. Este tipo de relación es comúnmente llamada Agregación (el objeto base utiliza al incluido para su funcionamiento).



Figura 2.3 Notación Agregación

Dependencia o Instanciación (uso)

Representa un tipo de relación muy particular, en la que una clase es instanciada (su instanciación es dependiente de otro objeto/clase). Se denota por una flecha punteada.

El uso más particular de este tipo de relación es para denotar la dependencia que tiene una clase de otra.



Figura 2.4 Notación Dependencia

2.2.2 Diagramas principales de UML

Se dispone de dos tipos diferentes de diagramas los que dan una vista estática del sistema y los que dan una visión dinámica.

Los diagramas estáticos son:

- ❖ **Diagrama de clases:** muestra las clases, interfaces, colaboraciones y sus relaciones. Son los más comunes y dan una vista estática del proyecto.
- ❖ **Diagrama de objetos:** Es un diagrama de instancias de las clases mostradas en el diagrama de clases. Muestra las instancias y como se relacionan entre ellas. Se da una visión de casos reales.
- ❖ **Diagrama de componentes:** Muestran la organización de los componentes del sistema. Un componente se corresponde con una o varias clases, interfaces o colaboraciones.

- ❖ Diagrama de despliegue: Muestra los nodos y sus relaciones. Un nodo es un conjunto de componentes. Se utiliza para reducir la complejidad de los diagramas de clases y componentes de un gran sistema. Sirve como resumen e índice.
- ❖ Diagrama de casos de uso: Muestran los casos de uso, actores y sus relaciones. Muestra quien puede hacer que y relaciones existen entre acciones (casos de uso). Son muy importantes para modelar y organizar el comportamiento del sistema.

Lo diagramas dinámicos son:

- ❖ Diagrama de secuencia o diagrama de colaboración: muestran a los diferentes objetos y las relaciones que pueden tener entre ellos, los mensajes que se envían entre ellos. Son dos diagramas diferentes, que se puede pasar de uno a otro sin pérdida de información, pero que nos dan puntos de vista diferentes del sistema. En resumen, cualquiera de los dos es un Diagrama de Interacción.
- ❖ Diagrama de estados: muestra los estados, eventos, transiciones y actividades de los diferentes objetos. Son útiles en sistemas que reaccionen a eventos.
- ❖ Diagrama de actividades: Es un caso especial del diagrama de estados. Muestra el flujo entre los objetos. Se utilizan para modelar el funcionamiento del sistema y el flujo de control entre objetos.

2.2.2.1 Diagrama de Interacción

El diagrama de interacción, representa la forma en como un Cliente (Actor) u Objetos (Clases) se comunican entre si en petición a un evento. Esto implica recorrer toda la secuencia de llamadas, de donde se obtienen las responsabilidades claramente.

Dicho diagrama puede ser obtenido de dos partes, desde el Diagrama Estático de Clases o el de Casos de Uso.

Elementos del diagrama de interacción

- ❖ Objeto/Actor: El rectángulo representa una instancia de un Objeto en particular, y la línea punteada representa las llamadas a métodos del objeto. A continuación se muestra en la siguiente figura:

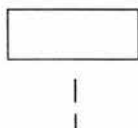


Figura 2.5 Objeto/Actor

- ❖ Mensaje a Otro Objeto: Se representa por una flecha entre un objeto y otro, representa la llamada de un método (operación) de un objeto en particular.

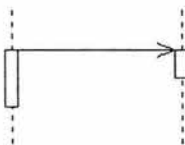


Figura 2.6 Mensaje a Otro Objeto

- ❖ Mensaje al Mismo Objeto: No solo llamadas a métodos de objetos externos pueden realizarse, también es posible visualizar llamadas a métodos desde el mismo objeto en estudio.

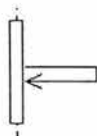


Figura 2.7 Mensaje al Mismo Objeto

2.2.2.2 Diagrama de Actividad

Los diagramas de actividad describen cómo se coordinan las actividades. Por ejemplo un diagrama de actividad puede utilizarse al igual que un diagrama de interacción) para indicar como podría implementarse una operación. Un diagrama de actividad es particularmente útil cuando se sabe que una operación tiene que alcanzar un número de cosas distintas y se quiere modelar cuales son las dependencias fundamentales entre ellas, antes hay que decidir en que orden se realizarán.

Los diagramas de actividad son también útiles para describir cómo se exponen los casos de uso individuales y puede depender de otros casos de uso. En ocasiones, los casos de uso que se descubren no suceden en un orden arbitrario sino como parte del flujo general de un área de actividades de un cliente.

Elementos del diagrama de actividad

- ❖ Actividad que aparece como una caja con nombre, y esquinas redondeadas. Técnicamente es un tipo de estado que se abona como respuesta a algún evento que llega desde afuera, sino cuando termina la actividad que presenta. La actividad puede implicar varios pasos, incluyendo la espera a la llegada de eventos, aunque esta actividad detallada normalmente no se muestra.



Figura 2.8 Actividad

- ❖ Transición que aparece como una flecha, igual que en el diagrama de estado. Las transiciones en un diagrama de actividad normalmente no se etiquetan, porque la transición es provocada por la finalización de la actividad propia y normalmente es más conveniente incluir las acciones en las actividades.



Figura 2.9 Transición

- ❖ Barra de sincronización es una barra gruesa horizontal que describe la coordinación entre actividades. Una vez que todas las transiciones dirigidas a la barra han terminado, pueden pasar la barra. En este punto, todas las actividades a la barra se han activado, por lo que a las actividades a las que se dirigen estas transiciones han empezado en paralelo. Es decir la barra de sincronización proporciona una manera de expresar varias tareas en paralelo.



Figura 2.10 Barra de sincronización

- ❖ Diamante de decisión se utiliza para representar las decisiones, como alternativa a las rutas a seguir de la transición de un estado a otro.

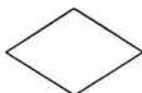


Figura 2.11 Diamante de decisión

2.2.2.3 Diagrama de Casos de Uso

“Los Casos de Uso documentan el comportamiento del sistema desde el punto de vista del usuario. En este caso por usuario que cualquier cosa ajena al sistema se desarrolla y que interactúa con el mismo”⁴. Un usuario podría ser una persona, otro sistema de información, un dispositivo hardware, etc.

La utilidad de los casos de uso es para realizar la captura de los requisitos, proporcionando una forma estructurada de abordarlos:

1. Identificar los actores.
2. Para cada actor, averiguar:
 - ❖ Lo que necesita del sistema, es decir, qué casos de uso hay que tienen valor para ellos.
 - ❖ Cualquier otra interacción que espera tener con el sistema, esto es, en que casos de uso podría formar parte para el beneficio del otro.

El diagrama de casos de uso representa la forma en como Cliente (Actor) opera con el sistema en desarrollo, además de la forma tipo y orden en como los elementos interactúan (operaciones o casos de uso).

Un diagrama de casos de uso consta de los siguientes elementos:

- ❖ Actor.
- ❖ Casos de Uso.
- ❖ Relaciones.
- ❖ Escenario.

⁴ Stevens Perditá Utilización de UML en Ingeniería del Software con objetos y componentes Pág. 150

Elementos del diagrama de casos de uso

- ❖ Actor: Un actor en un diagrama de casos de uso representa un rol que alguien tiene que cumplir. Se representa por la siguiente figura:



Figura 2.13 Actor

- ❖ Caso de Uso: es una operación/tarea específica que se realiza tras una orden de algún agente externo, sea desde una petición de un actor o bien desde la invocación desde otro caso de uso. A continuación se observa en la figura su representación:



Figura 2.14 Caso de Uso

Relaciones

La relación de comunicación entre un actor y un caso no significa que necesariamente que alguien de ese rol tenga que estar implicado en la ejecución de una tarea, simplemente significa que puede ser dependiendo de las circunstancias.

- ❖ Asociación

Es el tipo de relación más básica que indica la invocación desde un actor o caso de uso a otra operación (caso de uso). Dicha relación se denota con una línea simple.



Figura 2.15 Notación Asociación

❖ Dependencia o instanciación

Es una forma muy particular de relación entre clases, en la cual una clase depende de otra, es decir, se instancia (se crea). Dicha relación se denota con una flecha punteada.



Figura 2.16 Notación Dependencia

❖ Generalización

Este tipo de relación es uno de los más utilizados, cumple una doble función dependiendo de su estereotipo, que puede ser de Uso (<<uses>>) o de Herencia (<<extends>>).



Figura 2.17 Notación Dependencia

Este tipo de relación esta orientado exclusivamente para casos de uso (y no para actores).

extends: Se recomienda utilizar cuando un caso de uso es similar a otro (características).

uses: Se recomienda utilizar cuando se tiene un conjunto de características que son similares en más de un caso de uso y no se desea mantener copiada la descripción de la característica.

De lo anterior cabe mencionar que tiene el mismo paradigma en diseño y modelado de clases, en donde esta la duda clásica de usar o heredar.

Escenario

"Un escenario es una posible interacción entre el sistema y algunas personas o sistemas/dispositivos (en diversos roles)"⁵. La interacción puede describirse como una secuencia de mensajes.

A continuación se representa en la figura de una comunicación sencilla entre un actor y un caso de uso. Esta representación es denominada escenario.

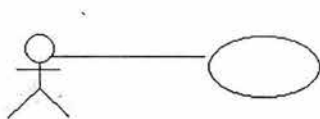


Figura 2.18 Escenario

2.2.2.4 Diagrama de Estados

Muestra el conjunto de estados por los cuales pasa un objeto durante su vida en una aplicación, junto con los cambios que permiten pasar de un estado a otro.

Elementos del diagrama de estados

❖ Estado

"Identifica un periodo de tiempo del objeto (no instantáneo) en el cual el objeto está esperando alguna operación, tiene cierto estado característico o puede recibir cierto tipo de estímulos"⁶.

Se representa mediante un rectángulo con los bordes redondeados, que puede tener tres compartimientos: uno para el nombre, otro para el valor característico de los atributos del objeto en ese estado y otro para las acciones que se realizan al entrar, salir o estar en un estado. Dicha representación se muestra en la siguiente figura.



Figura 2.19 Estado

⁵ Stevens Perdita Utilización de UML en Ingeniería del Software con objetos y componentes Pág. 159

⁶ Stevens Perdita Utilización de UML en Ingeniería del Software con objetos y componentes Pág. 162

❖ Marcas de creación y destrucción

Aparecen como un punto negro con una flecha (sin etiquetar) que apunta al estado inicial del diagrama. Y su representación se muestra en la siguiente figura.



Figura 2.20 Marca de creación

Un diagrama de estado puede mostrar también una marca de destrucción. Esto es un punto negro con un anillo alrededor, y significa que el objeto ha alcanzado el final de su vida y será destruido. Pueden existir varias marcas de destrucción en un diagrama o ninguna. El símbolo que representa a este tipo de marca se observa en la siguiente figura.



Figura 2.21 Marca de destrucción

❖ Eventos

Es una ocurrencia que puede causar la transición de un estado a otro de un objeto. Esta ocurrencia puede ser una de varias cosas:

- ❖ Condición que toma el valor de verdadero o falso.
- ❖ Recepción de una señal de otro objeto en el modelo.
- ❖ Recepción de un mensaje.
- ❖ Paso de cierto período de tiempo, después de entrar al estado o de cierta hora y fecha particular.

El nombre de un evento tiene alcance dentro del paquete en el cual está definido, no es local a la clase que lo nombre.

❖ Envío de mensajes

Puede representarse el momento en el cual se envían mensajes a otros objetos. Esto se realiza mediante una línea punteada dirigida al diagrama de estados del objeto receptor del mensaje. Se representa por la siguiente figura.



Figura 2.22 Envío de mensajes

❖ Transición simple

Una transición simple es una relación entre dos estados que indica que un objeto en el primer estado puede entrar al segundo estado y ejecutar ciertas operaciones, cuando un evento ocurre y si ciertas condiciones son satisfechas. Se representa como una línea sólida entre dos estados.

❖ Transición interna

Es una transición que permanece en el mismo estado, en vez de involucrar dos estados distintos. Representa un evento que no causa cambio de estado. Se denota como una cadena adicional en el compartimiento de acciones del estado.

2.2.2.5 Diagrama de Componentes

Componente: Es una parte física reemplazable de un sistema que empaqueta su implementación y es conforme a un conjunto de interfaces a las que proporciona su realización.

Algunos componentes tienen identidad y pueden poseer entidades físicas, que incluyen objetos en tiempo de ejecución, documentos, bases de datos, etc. Los componentes existentes en el dominio de la implementación son unidades físicas en los computadores que se pueden conectar con otros componentes, sustituir, trasladar, archivar, etc.

Características de los componentes:

- ❖ Empaquetan el código que implementa la funcionalidad de un sistema.
- ❖ Poseen sus propias instancias de objetos que constituyen el estado del sistema.

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas.

Un diagrama de componentes muestra clasificadores de componentes, las clases definidas en ellos, y las relaciones entre ellas. Los clasificadores de componentes también se pueden anidar dentro de otros clasificadores de componentes para mostrar relaciones de definición.

El diagrama de componentes hace parte de la vista física de un sistema, la cual modela la estructura de implementación de la aplicación por sí misma, su organización en componentes y su despliegue en nodos de ejecución. Esta vista proporciona la oportunidad de establecer correspondencias entre las clases y los componentes de implementación y nodos. La vista de implementación se representa con los diagramas de componentes.

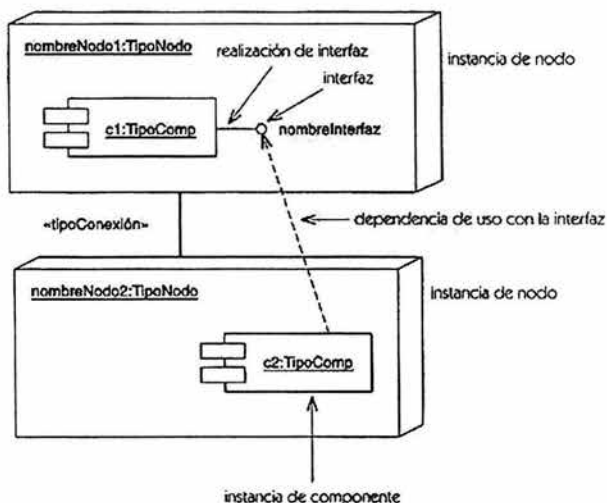


Figura 2.23 Diagrama global de componentes

3. SISTEMA MULTIMEDIA

3.1 Definición de sistema multimedia

Sistema multimedia: es una "colección de tecnologías basadas en la utilización de la computadora que da al usuario la capacidad de acceder y procesar información en por los menos tres de las siguientes formas; texto, gráficas, imagen fija, imagen con movimiento y audio"¹.

3.2 Principios para crear un sistema multimedia

1. Principio de múltiple entrada:

- ❖ Necesidad de varios canales para transmitir información como vídeo, audio, texto, etc.
- ❖ Deben estar sincronizados.

2. Principio de interactividad:

- ❖ Debe ser cuidadosa la interacción entre usuario e información y seguir las siguientes recomendaciones:
 - a) Tener un buen diseño para lograr que el mensaje se transmita.
 - b) Que el usuario siempre este activo.
 - c) El texto largo se tiene que dividir para que el usuario no se distraiga.
 - d) No poner componentes de más o inactivos en el sistema.
 - e) Que el sistema sea multiusuario.

3. Principio de libertad:

- ❖ Que el usuario pueda navegar de un sitio a otro.

4. Principio de retroalimentación:

- ❖ Para tener en cuenta este principio el que diseñe el sistema debe hacerse estas preguntas:
 - a) ¿Qué información se recoge?
 - b) ¿Cómo se presenta?
 - c) ¿A quien se dirige?
 - d) ¿Cómo se procesa?

5. Principio de vitalidad:

- ❖ Qué el usuario perciba la aplicación como si funcionara sola.

¹ <http://iteso.mx/~carlosc/pagina/documentos/multidef.html#concepto>

- ❖ La aplicación debe ser agradable para los usuarios (por ejemplo que tenga respuestas al movimiento del ratón)

6. Principio de la necesidad:

- ❖ La aplicación siempre tiene que servir para algo.
- ❖ La aplicación tiene que resolver algún problema.
- ❖ Cómodo.
- ❖ Accesible.

7. Principio de atención:

- ❖ Mantener al usuario pendiente de las actividades del sistema.

3.3 Producción multimedia

Los aspectos más importantes que se deben tomar en cuenta para la producción multimedia de un sistema se mencionan a continuación:

Guión: La historia y el flujo

Lo primero que hay que tener claro a la hora de desarrollar es que es lo que se quiere hacer.

Habrá que decidir el tipo de aplicación que se quiere hacer.

Se fijarán las partes y se decidirá en que modo se va a expresar la información de cada una de estas partes (mediante menús, ventanas, cajas de opciones, un espacio virtual, etc.)

Es conveniente realizar un diagrama para el flujo de la aplicación, es decir, los enlaces que van a existir entre las partes principales y cómo se va a discurrir la navegación y los accesos entre ellos.

Escenarios

Los escenarios o decorados son los lugares físicos donde va a transcurrir la aplicación.

Los escenarios no siempre tiene que ser realizados de una forma estrictamente lógica, sino que es conveniente jugar con el factor sorpresa, como por ejemplo, intentar diseñar escenarios divertidos para una aplicación de contenido serio, para descargar un poco de rigurosidad.

Personajes

Son los elementos que participan de una manera activa o pasiva en la historia.

Sonido

Para el sonido se tiene los siguientes tipos: el sonido digitalizado y el MIDI. Para la banda sonora es conveniente que el sonido sea uno o varios archivos MIDI porque ocupan mucho menos espacio. Pero para las voces, es

aconsejable utilizar sonido digitalizado. Por último en lo que respecta a los efectos sonoros, se puede optar por alguno de los dos indistintamente.

Video

El video es un aspecto muy importante de las aplicaciones multimedia, debido a que aporta mucha realidad y espectacularidad. Es importante tomar en cuenta los siguientes parámetros para este aspecto; el número de fotogramas, colores, velocidad, etc.

Animaciones

"Las animaciones son secuencias de fotogramas encadenados (en dos o tres dimensiones) que dan la sensación de movimiento"². Esto sirve para producir determinados efectos, para animar, alegrar y enriquecer un determinado evento, etc.

Interfaz

La interfaz debe ser la parte más importante de la aplicación multimedia. Debe seguir las siguientes pautas para su diseño:

- ❖ Es importante que existan zonas de pantalla que se mantengan fijas, para que el usuario las identifique fácilmente.
- ❖ Los botones deben estar agrupados conforme a acciones comunes, deben aparecer siempre en los mismos puntos de la pantalla y tener siempre el mismo comportamiento ante acciones similares.
- ❖ Es conveniente identificar las pantallas con algún tipo de rango o categoría.
- ❖ Siempre deben existir botones o zonas que nos permitan salir directamente de la aplicación, avanzar o retroceder hasta un punto anterior.

Diseño de enlaces

En cada pantalla deben definirse que enlaces van a existir en otras pantallas o zonas de la aplicación y cómo responderán las distintas partes de la interfaz ante distintos eventos.

Localización de zonas calientes

Las zonas calientes de una pantalla o de una interfaz son los puntos en los que va a ocurrir una determinada acción cuando el usuario realice un determinado evento.

Existen diferentes formas de marcar las zonas calientes, los más utilizados son:

- ❖ Un cambio de color en el texto (hipertexto), como en Internet.

² Multimedia aplicada / Centro de Computación Profesional de México Pág.196

- ❖ Un cambio de color o de intensidad en un determinado objeto.
- ❖ Un sonido identificativo.

El cambio de forma del cursor del ratón, adoptando la forma de una mano o una flecha.

Una vez especificadas cuáles van a ser las zonas calientes, es necesario definir cómo van a comportarse y que acciones van a realizar cuándo se les apliquen una serie de eventos.

Es importante señalar que las zonas calientes marcan los puntos en los que se tiene un enlace.

Diseño de la interfaz

- Utilizar una herramienta de dibujo (Adobe Photoshop).
- Dibujar la interfaz a mano y luego digitalizar.

- ❖ Parámetros generales del diseño

Para realizar el diseño es conveniente tener en cuenta los siguientes parámetros:

El tamaño que la aplicación va a tener en pantalla.

El tamaño máximo de visualización del entorno en que se va a ver para la pantalla (320X240, 640X480, 800X600) el de 640X480 suele ser el estándar.

El número de colores: 256, 16 bits o 24 bits.

Dibujar Interfaz

- ❖ Para dibujar la interfaz la recomendación es tener imaginación y creatividad.
- ❖ Diseñar e incluir las zonas de pantalla.

Una vez que la interfaz genérica ha sido dibujada es conveniente marcar una serie de zonas en la pantalla en donde quedaran fijados los botones y las principales partes de la aplicación.

Diseñar y añadir botones

Una vez que las zonas de pantalla han sido definidas hay que dibujar y fijar los distintos botones los pasos a seguir son los siguientes:

Siempre teniendo en cuenta el guión y las posibilidades y alternativas que vamos a tener en cada pantalla en cuestión, se fijará el número y utilidad de los botones que se van a utilizar.

Los botones tienen que ser identificativos.

Para ser identificativos se pueden utilizar iconos o símbolos que representen lo que quiere expresar.

Otra alternativa es añadir un texto informativo al botón indicando para qué sirve. Es conveniente que los botones estén agrupados por categorías de acciones similares.

Botones alternativos

Indican al usuario una zona caliente, es decir, el botón original al pasar el ratón sobre él este podría cambiar de apariencia y este nuevo icono será nuestro botón alternativo.

Diseñar y añadir escenarios

Decidir que tipo de diseño que se va a utilizar, ver que tipo de fondos en los que irán apareciendo el video, animaciones, texto.

También otra forma de diseñar el escenario es dibujar como va transcurriendo la historia.

Diseñar y añadir personajes

Se diseñan como elemento alternativo de la interfaz, es decir, no van vinculados con ella.

Añadir texto

El proceso de añadir texto puede realizarse en distintos momentos del desarrollo de la aplicación.

Los rótulos que aparecen en la interfaz, asociados con los botones o zonas de la pantalla. Este texto se incluirá en el proceso de diseño de las pantallas y siempre permanecerá fijo.

Textos que pueden aparecer y desaparecer de la interfaz, éstos deberán guardarse aparte como en los personajes y se debe guardar como un archivo de imagen.

El texto que representa la información puede ser muy extenso, por lo que es conveniente que se guarden

❖ Las fuentes

Las fuentes representan la forma que van adquirir las letras a la hora de representar la aplicación. Se deben de seguir una serie de criterios que son los siguientes:

1. Es conveniente elegir las fuentes de una manera adecuada, con juntándolas siempre con el resto del diseño, como por ejemplo utilizar fuentes góticas o barrocas en una historia de terror.
2. La legibilidad y facilidad de lectura es otro factor importante que vendrá dado por el tamaño y color de las fuentes.
3. Es conveniente pensar si la computadora en el que se va a instalar la aplicación van a estar presentes las fuentes elegidas, por lo que siempre es conveniente incluirlas con la aplicación.

Guardar archivos

Una vez que todos los elementos gráficos han sido diseñados, hay que guardarlos en un formato adecuado. Es importante que el formato elegido sea el correcto, porque habrá que tener en cuenta el número de colores y el tamaño de los gráficos, así como su factor de compresión, entonces debemos tomar en cuenta los siguiente:

Se debe tener en cuenta en que tipo de aplicación vamos a incluir los gráficos, y saber que formato soporta, para grabarlo en uno que pueda ser utilizado por ésta.

El factor y tipo de compresión es importante. Siempre hay que elegir el mayor factor posible, atendiendo a la calidad y rapidez de visualización.

Si se van a emplear 256 colores es necesario que todas las imágenes se guarden bajo la misma paleta.

Para guardar imágenes se puede tener en cuenta una serie de normas útiles, las cuales son mostradas en la siguiente tabla:

Tipo de aplicación	No. de colores	Formato de archivos	Paleta
Ejecutable para Windows	256 colores	.BMP O .PIC	Del sistema
	RGB	.BMP O .PIC	Ninguna
Para Internet (opaco)	256 colores	.GIF	De Internet
	RGB	.JPG(comprimido)	Ninguna
Para Internet (transparente)		.GIF89a	Ajustable
Soporte Impreso		.TIF	Ninguna

Tabla 3.1 Formato de archivos para imágenes

❖ Formato y tipo de animación

Existen dos tipos básicos de gráficos:

Los gráficos planos que se pueden generar animaciones.

Los gráficos en 3D.

Las técnicas y métodos de trabajo empleados de trabajo empleados en cada uno de los tipos de animaciones son muy diferentes, el diseñador de la aplicación debe sopesar cuál es la que puede encajar mejor en cada momento.

❖ Diseño de animaciones clásicas

Las animaciones clásicas se generan a partir de una serie de fotogramas continuos, algunas posibilidades para generar una animación más habitual son:

1. Una sola secuencia animada del tipo comienzo-fin. Una vez terminada la animación se guarda en un solo archivo de secuencia o video-clip.

2. Un conjunto de movimientos animados y repetitivos (por ejemplo un personaje de videojuego).

❖ Trazo de movimientos

Los personajes, una vez animados, deben poderse mover por la pantalla, siguiendo una serie de pautas y limitaciones y pueden ser:

- ❖ Movimientos de personajes realizados por voluntad del usuario, utilizando para ello el teclado, el ratón, etc.
- ❖ Movimientos controlados por la computadora.

Dentro de estos dos tipos existen otros que son los siguientes:

Tipo de movimiento	Acción	Controlado por
Mediante un dispositivo	El usuario indica al personaje si debe subir, bajar.	El usuario
Mediante el teclado	Permite mover al personaje en cualquier dirección utilizando las teclas.	El usuario
Juego de plataformas	El personaje avanza, retrocede, sube, baja, etc.	El usuario

Tabla 3.2 Trazo de movimientos

3.4 La navegación multimedia

Existen diferentes métodos de navegación y acceso a la información, y unas de las más importantes pueden ser:

- ❖ Acceso lineal: Se accede desde la primera pantalla hasta la última de una forma secuencial. Este tipo de navegación se representa en la siguiente figura

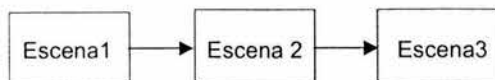


Figura 3.1 Acceso lineal

- ❖ Acceso indexado: Se accede de una pantalla a otra utilizando las entradas de uno o varios índices. Es como un acceso secuencial a saltos.

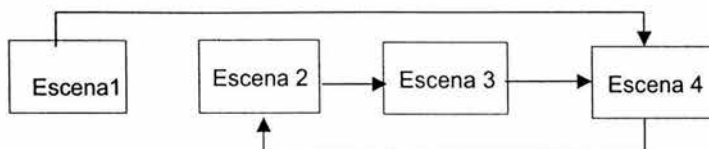


Figura 3.2 Acceso indexado

- ❖ Acceso circular: Al llegar a la última pantalla, ya sea mediante un acceso secuencial o indexado, se accede de nuevo a la primera de una manera cíclica.

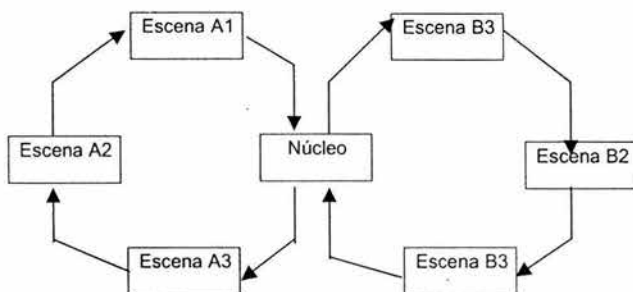


Figura 3.3 Acceso circular

- ❖ Acceso Estrella: Desde cualquier pantalla se puede acceder a cualquier otra. Se puede establecer limitaciones (lo que daría lugar a una estructura en forma de árbol).

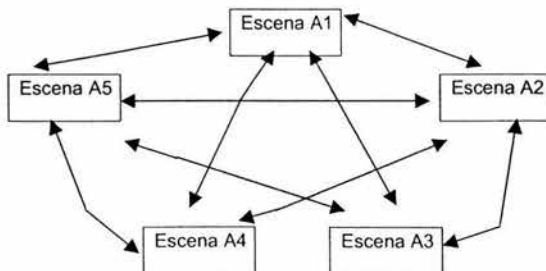


Figura 3.4 Acceso estrella

- ❖ Acceso múltiple: Se combinan algunos o todos los accesos anteriores.

3.5 El sistema multimedia en el proceso pedagógico

3.5.1 Funciones pedagógicas de los sistemas multimedia

El sistema multimedia responde a las siguientes funciones: cognoscitiva, comunicativa, motivadora, informativa, integradora, sistematizadora, y de control.

- ❖ **Función cognoscitiva:** "señala la forma de estructurar el proceso de aprendizaje como un proceso del conocimiento"³, requiere el empleo de medios de enseñanza, y por supuesto el sistema multimedia es uno de ellos. Este sistema actúa cumpliendo con el principio del carácter audiovisual de la enseñanza, y de esta manera permite establecer el camino entre las representaciones de la realidad objetiva en forma de medios y los conocimientos que asimilarán los estudiantes.

La justificación de lo anterior se basa en la teoría Leninista del conocimiento al llevar la dialéctica a este campo. El papel de los medios está en proporcionar verdaderamente el puente o vínculo entre las percepciones concretas y el proceso lógico del pensamiento. El sistema multimedia al actuar en su carácter sistémico, aprovecha al máximo las posibilidades de los canales sensoriales a través de los cuales se manifiestan los distintos medios que lo forman. De esta manera explota con gran fuerza la memoria visual y auditiva, logrando una mayor apropiación de los conocimientos. Por supuesto que la adquisición de estos conocimientos requiere ante todo de la relación objetivo-contenido-método.

- ❖ **Función comunicativa:** está apoyada en el papel que los medios de enseñanza cumplen en el proceso de la comunicación. En el mismo ocupan el lugar del canal que es a su vez soporte de la información, es vínculo portador del mensaje que se transmite a los estudiantes. Por tal razón "el sistema multimedia actúa como soporte a partir del cual se desarrolla el proceso comunicativo entre los realizadores del mismo y los estudiantes que lo emplean"⁴. Es en ese momento donde el sistema multimedia manifiesta la interactividad con el estudiante. El puede seleccionar la información, el camino, y el sistema multimedia le puede sugerir otras vías y otras fuentes alternativas o no a las que pretende tomar. La interacción es parte de la función comunicativa pues con ella se logra la verdadera comunicación con el sistema. Este proceso no debe verse solamente entre el sistema y los estudiantes sino que se extiende a las posibilidades de comunicación telemática con otros profesores, estudiantes, o centros remotos, situación que no es cumplida por otros medios hasta el presente.

³ <http://usuarios.lycos.es/leodata/docencia/GDS/unidad%203/UNIDAD03.pdf>

⁴ Ibid

Esto último es poco explotado aún, pero a medida que las comunicaciones informáticas incrementen su presencia física e Internet se expanda, esta comunicación bilateral dejará de ser una novedad para ser una constante.

- ❖ Función motivadora: "aumenta la motivación por la enseñanza al presentar estímulos que facilitan la auto actividad del alumno, la seguridad en el proceso de aprendizaje y el cambio de actividad"⁵. Este sistema muestra desde el primer momento una manera novedosa de presentar los conocimientos, apoyada en su forma, en la integración de medios y en las estructuras de navegación. Cada uno de ellos contribuye de forma efectiva a facilitar e incrementar el autoaprendizaje del estudiante en este sistema educacional.

Las estructuras de navegación elevan considerablemente la motivación del estudiante hacia la materia o el curso que estudia, pues lo pone cada vez en contacto con un nuevo conocimiento, un nuevo medio, un camino distinto cada vez. El propio sistema multimedia en su conjunto se constituye en un medio novedoso en parte por su escasa utilización en los sistemas a distancia y por el hecho de integrar a otros medios.

- ❖ Función informativa: "maneja el empleo de los medios permitiendo brindar una información más amplia, completa y exacta, ampliando los límites de la transmisión de los conocimientos. La aplicación del sistema multimedia enriquece el proceso de transmisión de la información que es necesario en la educación, debido a la integración de medios, a las posibilidades de búsquedas de información fuera del propio sistema, a las consultas con el profesor y otros alumnos, así como a la interactividad entre el sistema y el estudiante"⁶.

La integración de medios como el video, animaciones, sonidos, fotografías y textos diversifican las fuentes de adquisición de información, donde una añade una nueva información a la otra para completar un conjunto superior de informaciones sobre un mismo tema. De esta manera el estudiante encontrará una amplitud de informaciones sobre el tema que estudia o las recomendaciones para buscarlas en otros sitios.

- ❖ Función integradora: es aquella en donde "la integración de medios no significa la sustitución de ellos, ni la sobre valoración de este medio por encima de otros"⁷. Pero en la enseñanza es importante facilitar al estudiante el acceso a la información, el ahorro de tiempo y la disminución del esfuerzo en el aprendizaje. Estas necesidades las cumple el sistema multimedia al permitir la integración de numerosos medios, de esta manera el estudiante no tiene que buscar en el libro la

⁵ Ibid

⁶ Ibid

⁷ Ibid

tarea, en el casete de audio escuchar la grabación o ver la animación en el video, pues todos ellos estarán integrados en el propio sistema. Pero esta función además se extiende a la integración de los contenidos.

- ❖ **Función sistematizada:** obedece a la planificación del trabajo con el sistema multimedia, la que se cumple desde la etapa de elaboración del mismo. Aun cuando este medio se caracteriza por la navegación no lineal, ello no significa en modo alguno que el aprendizaje sea improvisado. "La sistematización garantiza que el estudiante pueda ir ampliando sus conocimientos a medida que avanza en el trabajo con el multimedia y a su vez va comprobando lo aprendido"⁸. Se manifiesta en esta función la articulación didáctica de los componentes del multimedia en particular y del proceso pedagógico en general. De igual manera la repetición de tareas cuando estas no han sido solucionadas de manera satisfactoria por el estudiante o aquellas necesarias durante todo el proceso, la ejercitación y la aplicación de lo aprendido se manifiestan en el sistema multimedia. Este medio permite además el desarrollo integrado del pensamiento del estudiante al permitir la manifestación de las operaciones lógicas de análisis, síntesis, abstracción, inducción y deducción.

El sistema multimedia manifiesta su función de control a partir de la posibilidad que tiene el estudiante de comprobar su aprendizaje, y el profesor de conocer este. La retroalimentación que el obtiene mediante su auto evaluación le permite además corregir los métodos que emplea, su eficiencia y trazarse nuevas formas de auto enseñanza.

3.5.2 Posibilidades didácticas

Las posibilidades didácticas son el resultado integrado del funcionamiento de sus componentes. De esta manera el sistema multimedia reúne un grupo de posibilidades didácticas extraídas al analizar las funciones del sistema en el proceso pedagógico.

Los sistemas multimedia tienen la posibilidad de ofrecer la sensación de variar la estructura de la organización de la información al facilitar la búsqueda no secuencial. La búsqueda no lineal significa que el estudiante sigue un patrón que no se corresponde con la lectura lineal aplicada a los medios. Hasta el presente los medios de enseñanza empleados no tenían la posibilidad de estructurar sus mensajes de manera no lineal.

El estudiante para observar un fragmento de un video, o escuchar un sonido grabado debe rebobinar toda la cinta hasta el sitio deseado. Pero las características del hipertexto en la que no se tiene que recurrir a una secuencia fija y previamente determinada para la búsqueda de la información rompe con la linealidad y ayuda al estudiante a seleccionar el camino que seguirá y que guarda relación con sus características personales, gustos y aspiraciones.

⁸ Ibid

La sensación de la no linealidad se manifiesta en el hecho que existe un orden de la información, esta no se expone de manera indiscriminada, desorganizada, todo lo contrario, solo que esta información es seleccionada por el equipo realizador del sistema multimedia y debe disponerla de manera tal que el estudiante seleccione dentro de la organización el camino que considera mas adaptado a sus características, dando la sensación de una ruta variable.

Lo anterior es importante para comprender que el sistema multimedia favorece el aprendizaje individual de los estudiantes, ya que se adapta a sus características, se vuelven protagonistas de su aprendizaje, favoreciendo su participación y actividad e incrementando su grado de responsabilidad. Los estudiantes se encuentran en situaciones que le exigen la toma de decisiones y la búsqueda de respuestas adecuadas.

El sistema multimedia libera al estudiante de la participación directa en la selección de la información en otras fuentes, por ello el centro de gravedad se desplaza de los materiales impresos y los medios audiovisuales entre otros medios dispersos, no solo en el plano físico sino en el enfoque sistémico al multimedia. Con ello además se contribuye a ahorrar tiempo al trabajo del estudiante a distancia.

Otra posibilidad didáctica es la interactividad del estudiante con el sistema multimedia pero bajo el control de éste. El estudiante puede manejar la información como lo desee; puede imprimirla, copiarla a otro archivo, modificarla, realizar búsquedas en otros sitios, organizar el ritmo de lectura, detener en el momento que desee su estudio y comenzar en el mismo sitio que lo dejó. La interactividad es sinónimo de calidad del sistema multimedia, pero por sí sola no garantiza la función de reforzar el mensaje.

Los sistemas futuros deberán combinar estas dos potentes características en busca de satisfacer los más variados intereses de los receptores, dejando atrás el criterio de aislamiento de los estudiantes a distancia.

Los sistemas multimedia estimulan la creatividad de los estudiantes, pues no sólo reaccionan ante el multimedia expuesto sino que están en posibilidades de crear otros, o incorporar nuevas situaciones al que está analizando.

Otra posibilidad viene dada por la complementariedad de los mensajes, los contenidos o las informaciones útiles al estudiante. No quiere decir que en el diseño del sistema multimedia un mismo mensaje se deba obligatoriamente dar mediante diferentes medios, sino todo lo contrario. Es decir un medio de los que componen el sistema es encargado de transmitir la información mediante su lenguaje, otro complementa la misma, mientras que un tercer medio acaba de dar la idea completa.

Las cualidades que integran el sistema y la simultaneidad de acciones posibles entre cada componente son las causas directas de la complementariedad de las informaciones.

Uso variado durante un curso, la capacidad del sistema multimedia es variable, por lo que pueden ser empleados para tratar más de un asunto en un curso, o en varios. En este caso las enciclopedias y atlas multimedia son un ejemplo del empleo en varios cursos y por materias diferentes. Pero a ello se une también que los daños físicos que puede sufrir el sistema son mínimos si está soportado en un disco compacto y casi nulos si está copiado al disco duro de la computadora, si se compara con los que sufren los casetes de video y diapositivas entre otros medios.

Facilidad en su empleo, provocado por lo sencillo de los medios utilizados los que no requieren de locales especiales, condiciones específicas, horarios o habilidades muy específicas. Un mismo sistema multimedia puede ser empleado por diferentes estudiantes. Integran a los restantes medios de enseñanza. Es justamente el sistema multimedia el encargado de integrar a todos los medios de enseñanza, tal y como lo hizo anteriormente el video. El cumplimiento del principio didáctico de la enseñanza audiovisual no implica en modo alguno que la cantidad de medios sea sinónimo de calidad en la enseñanza, pues esta se basa en la selección adecuada de los medios.

Pero es indiscutible que los profesores y estudiantes deben utilizar una variedad de medios acorde con los métodos empleados en el proceso de adquisición del conocimiento y los que a su vez requieren algunos recursos para su visualización.

El sistema multimedia viene a agrupar, integrar estos medios, de manera que tanto estudiantes como profesores pueden adentrarse en el texto escrito y reforzar la idea con una foto de lo que leen, con un video o una animación del fenómeno que estudian.

La integración solo debe analizarse desde la posición de incluir en un multimedia todos los medios que el profesor debe manejar en la clase, sobre todos aquellos audiovisuales, a lo que se añade la capacidad de interconexión con sitios lejanos al aula docente.

3.5.3 Limitaciones didácticas

Las limitaciones didácticas son aquellos aspectos que el sistema multimedia no puede lograr o en los cuáles se encuentra en desventaja con otros medios. Este medio tiene muy pocas desventajas, lo que no significa que sea el medio ideal, o el más importante. Todo lo contrario cada medio tiene sus ventajas y desventajas, cada uno tiene su propio lenguaje y por ello empleará su propio canal para transmitir el mensaje.

Una limitante de este sistema es que está concebido para ser empleado por pocos estudiantes a la misma vez. Pero en ningún momento puede pensarse en que el sistema multimedia individualiza la enseñanza pues el estudiante tiene la posibilidad de estar en contacto directo o virtual con el profesor y otros estudiantes.

El vínculo de este medio con la computadora requiere de una educación informática tanto de profesores como de estudiantes. Las facilidades de manejo de estos sistemas no determinan tanto su expansión sino el hecho del conocimiento informático requerido para producir otros multimedia.

El sistema multimedia puede estar al alcance tanto de profesores como de estudiantes, pero para ello es necesaria la presencia de conocimientos mínimos de informática, lo que viene a constituirse en una limitante al sistema. No obstante lo anterior se conoce que numerosos sistemas educacionales están introduciendo aceleradamente programas de estudio que permiten poner a profesores y estudiantes en contacto con las modernas técnicas informáticas. De igual manera la presentación de numerosos sistemas multimedia determinan que el estudiante y el profesor deban manejar mayor cantidad de imágenes simbólicas, para las que deben estar preparados.

Otra limitante didáctica de este medio es que los estudiantes pueden perderse en la navegación sin saber por donde proseguir su itinerario de acceso a dicha información. El problema puede reducirse al dotar al sistema de ayudas, generando estructuras de tipo asociativo para aumentar la orientación.

3.6 Aplicación de los sistemas multimedia en diversas áreas

La multimedia es una tecnología que está encontrando aplicaciones, rápidamente, en diversos campos, por la utilidad social que se le encuentra. Comenzó por aplicaciones en la diversión y el entretenimiento a través de los juegos de video. De allí se pasó a las aplicaciones en la información y la educación, para pasar al campo de la capacitación y la instrucción, a la publicidad y marketing hasta llegar a las presentaciones de negocios, a la oferta de servicios y productos y a la administración. Inicialmente, lo que se aprovecha de este recurso es su enorme capacidad de ofrecer información atractiva. En México, aparte de la aplicación de los juegos de video y de los programas de cómputo empleados para el autoaprendizaje de software, el desarrollo de la multimedia se impulsa gracias a las aplicaciones en las presentaciones de negocios, la industria, la capacitación y los módulos de información.

Diversión y entretenimiento

Multimedia es la base de los juegos de video, pero también tiene aplicaciones en pasatiempos de tipo cultural como cuentos infantiles interactivos, exploración de museos y ciudades a manera de visitas digitales interactivas.

Negocios

Las principales aplicaciones se dan en: la inducción, capacitación y adiestramiento de personal, la disposición rápida, accesible y procesamiento de altos volúmenes de información, los módulos de información, las presentaciones, intercambio y circulación de información. El trabajo en grupo o de equipo para elaborar proyectos.

En los negocios de destacan los siguientes beneficios: el incremento del rendimiento del usuario, la reducción de costos en el entrenamiento, la reducción del retraso de la productividad de los programadores, al acortar la curva de aprendizaje; lo que permite tomar ventajas e incrementar la utilización del equipo.

Señala el problema de la administración del cambio de un sistema viejo a uno nuevo, cuando éste es sustancial, puesto que exige reaprender secuencias; sin embargo. Otro problema, que señala, lo constituyen los errores de requerimiento del recurso, cuando no se conoce la herramienta o la estructura de la aplicación: redundancia en pérdida de tiempo para gente de soporte o desarrollo y representa un alto costo.

Publicidad y mercadeo

Las principales aplicaciones son: la presentación multimedia de negocios, de productos y servicios, la oferta y difusión de los productos y servicios a través de los módulos de información.

Los módulos de información son máquinas multimedia situadas en espacios públicos estratégicos, con determinado tipo de dispositivos que, mediante una aplicación, accedan datos y permiten al usuario interactuar con ellos, obteniendo, así, información.

El módulo proporciona información de forma atractiva, sirviendo de apoyo a museos, centros comerciales, salas de espera de bancos, restaurantes, hospitales, consultorios, etc. La función del módulo es transmitir información cultural, comercial o de trámite de servicios y proporcionar acceso a la información para involucrar en el adiestramiento o el aprendizaje. Para cumplir tales funciones, se requiere evaluar periódicamente la información que proporciona, actualizarla y presentarla permanentemente con cambios esporádicos.

Difusión del saber y conocimiento

La característica de la interactividad de multimedia, que permite navegar por el programa y buscar la información sin tener que recorrerlo todo, logra que la tecnología se aplique en los nuevos medios de dos modos diferentes y se use de tres formas alternativas. A continuación se presenta la tabla:

SISTEMA MULTIMEDIA EN LOS NUEVOS MEDIOS	a) Como medio aprendizaje	<ul style="list-style-type: none"> • Por interacción, al ritmo personal, simulando situaciones reales. • Con juegos que agilizan habilidades.
	b) Como medio informativo	<ul style="list-style-type: none"> • Conectado a bibliotecas electrónicas. • Acceso a información, desde casa, por correo electrónico.
LOS USOS DE LOS SISTEMAS MULTIMEDIA	<ol style="list-style-type: none"> 1) Medio de orientación. Presentaciones multimedia de índices de orientación en bancos y museos. Por módulos o kioscos de información. 2) Medio didáctico. Capacitación (interactividad y simulaciones). Dominio teórico previo a práctica. Posibilita conjugar actitudes y creatividad. 3) Libro electrónico. Mediante el CD-ROM se puede tener acceso a libros y bibliotecas. 	

Tabla 3.3 Formas de aplicación de los sistemas multimedia.

Administración

Los sistemas multimedia permiten tener a la vista los acostumbrados inventarios de productos, más que por columnas de números, por registros e inspecciones de cámaras de video de los estantes de almacén, realizados por el administrador de éste. Igualmente permite revisar y analizar reportes de clientes realizados por video, de manera más rápida y efectiva. La realización del trabajo en colaboración es, así mismo, posible, aún con personas que están en lugares distantes o diferentes.

4. ANÁLISIS DEL SISTEMA

4.1 Planteamiento del problema

En el presente capítulo se hace un análisis del sistema a desarrollar, el cual incluye las etapas de Lanzamiento, Estrategia, Planeación, Requerimientos y Diseño de la Metodología TSPi aplicadas al producto que se desea, es decir al "Tutorial del Microcontrolador HC11".

El sistema "Tutorial del Microcontrolador HC11" surge como una propuesta del Laboratorio de Multimedia e Internet de la Facultad de Ingeniería para que los alumnos de la materia del Laboratorio de Microcomputadoras cuenten con un material didáctico, el cual contiene la teoría básica del microcontrolador MC68HC11, los manuales de software para editar, simular y programar el microcontrolador, los manuales del MC68HC11 de la serie A, E y F del microcontrolador, el software necesario para la realización de las prácticas y una sección para buscar conceptos relacionados con la teoría básica del tutorial.

La principal razón por la cual se ha elegido la metodología TSPi es porque su estructura permite desarrollar proyectos de software en equipo, siguiendo un proceso bien definido con el cual se adquiere un aprendizaje más efectivo y una retroalimentación rápida. Esto es posible por que esta metodología posee un marco de trabajo definido, que se puede medir y además se repite en cada ciclo del proyecto en que éste se divida. Además por que permite dividir el proyecto en ciclos el cual hace más fácil el trabajo.

El seguimiento de ésta metodología deberá ser reflejado en la calidad del producto final de acuerdo a las necesidades del cliente.

Algo muy importante que debe destacarse en el desarrollo de productos de software es que debe contarse con una herramienta más, que complementa la metodología TSPi, ésta herramienta es el Proceso Personal de Software (PSP), el cual tiene como principal objetivo controlar las actividades en forma individual para administrar un proyecto y generar un producto de alta calidad, mediante la disciplina en administración y registro de tiempo y conteo de defectos.

Básicamente consiste en el registro de la siguiente información: estimación del tiempo de actividades similares, planeación y administración de tiempo, generación de planes y asignación de tiempo del proyecto, estimación del tamaño, calidad de los productos, establecimiento y evaluación de mejoras para las actividades.

El número de ciclos de desarrollo del producto de software son dos porque la distribución del trabajo es más equitativa además que la metodología proporciona esa gran ventaja. En el primer ciclo de desarrollo del tutorial se realizaron las secciones correspondientes a la Teoría Básica, Prácticas, Manuales, Software y Acerca de; dejando para el segundo ciclo el desarrollo de la sección Buscar.

A continuación se presentan las etapas mencionadas anteriormente de la metodología durante los dos ciclos de desarrollo.

4.2 Lanzamiento

Esta etapa contiene el establecimiento de los objetivos tanto personales como de equipo, los cuales al final del desarrollo del sistema deberán ser revisados para verificar que se cumplieron. Además estos objetivos permiten proporcionar la base para la construcción del sistema.

Los objetivos debieron basarse en los siguientes puntos principales:

- ❖ Generar un producto de calidad.
- ❖ Realizar un proyecto bien administrado y que sea productivo.
- ❖ Además que objetivos deben ser medibles.

A continuación se presentan los objetivos que se plantearon al inicio del desarrollo del Tutorial del Microcontrolador HC11:

Objetivos personales:

- ❖ Aprender y seguir la metodología TSPi en todas sus etapas.
- ❖ Terminar el proyecto en el tiempo establecido con el 100% de eficiencia.
- ❖ Realizar todas las tareas o actividades en el tiempo establecido.

Tomando en cuenta el alcance de los objetivos durante el primer ciclo, para la segunda etapa (ciclo 2) se espera cumplir con los siguientes objetivos de equipo:

1. La terminación completa del Tutorial de Microcontrolador HC11.
Para el ciclo2 se realizará la ventana de búsqueda, así como los gráficos faltantes en la parte del material. Y se completará la creación del sistema multimedia (imágenes y diseño gráfico de la interfaz)
2. El manejo de la metodología TSPi y PSP.
Al terminar este ciclo se espera manejar por completo la metodología de TSPi y PSP, para poder aplicarla en cualquier momento.
3. El conocimiento de programación orientada a objetos utilizando Visual Basic.
De igual forma terminando este ciclo tendremos más conocimientos en la programación orientada a objetos utilizando Visual Basic.

Aplicando estos tres puntos quedará cubierto por completo el objetivo del proyecto, para llegar a tener un buen producto final con el mínimo de defectos. También con estos puntos quedará cubierto el objetivo del Laboratorio de Multimedia e Internet.

Cabe señalar que desde el momento en que se inicia esta etapa se dio inicio al seguimiento del PSP, por lo que se lleva el registro de las formas que se mencionan en el capítulo 1 de este trabajo.

4.3 Estrategia

En esta etapa de desarrollo se presenta la estrategia para realizar el proyecto, además de crear el diseño conceptual, realizar las estimaciones de tamaño del producto y de tiempo de desarrollo del mismo.

En ésta misma etapa se tomó la decisión de establecer dos ciclos de desarrollo para el sistema.

Para elaborar la estrategia de desarrollo del sistema debe contarse con una serie de necesidades que deben ser satisfechas al finalizarse el producto.

4.3.1 Lista de necesidades del Sistema

La descripción que a continuación se muestra corresponde a las necesidades del sistema:

Se requiere desarrollar un sistema que ayude a los alumnos de la materia de Microcomputadoras de la Facultad de Ingeniería a desarrollar sus prácticas del laboratorio correspondiente.

Las necesidades se denotan con una "N" para indicar que se refiere a una necesidad, seguida de un número consecutivo que podrá distinguir a cada una.

Las necesidades del tutorial son las siguientes:

- N1.** El usuario necesita acceder al Documento de teoría básica del HC11.
- N2.** El usuario necesita acceder a las prácticas del laboratorio del HC11.
- N3.** El usuario necesita acceder a los manuales del HC11.
- N4.** El usuario necesita utilizar el software del tutorial.

4.3.2 Gráfica de dependencias del sistema

Estas mismas necesidades pueden ser representadas en conjunto por una gráfica de dependencias, la cual como su nombre lo indica, muestra en forma gráfica cada una de las necesidades del sistema y su dependencia entre ellas; si es que ésta existe.

Cada una de las necesidades es representada por un rectángulo y en su interior se muestra la notación definida en la sección anterior.

Esto se muestra en la siguiente figura que representa la gráfica de dependencias del tutorial.

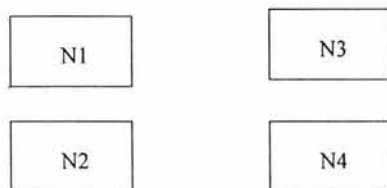


Figura 4.1. Gráfica de dependencias del sistema

Nota: Entre las necesidades del sistema no existe ninguna dependencia.

4.3.3 Estimaciones

Las estimaciones que se realizan están enfocadas a la manera en que se distribuye el desarrollo del sistema, tomando como base la lista de necesidades y además clasificándolas de acuerdo al ciclo en el que serán implementadas.

A continuación se pueden observar las tablas que representan las estimaciones durante los ciclos 1 y 2 de éste sistema.

Estimaciones del sistema en el ciclo 1

Pantallas a realizar:	Ciclo 1	Ciclo 2
El usuario consulta el menú principal.	X	
El usuario puede consultar el documento de la teoría básica del HC11.	X	
El usuario puede consultar la selección de las prácticas.	X	
El usuario puede tener acceso a cada una de las prácticas del HC11.	X	
El alumno puede ir al selector de manuales.	X	
El alumno puede ir a cada uno de los manuales.	X	
El alumno puede acceder a Acerca de.	X	
El alumno puede ir a la selección de software.	X	
El alumno puede acceder a los manuales de software.	X	
El alumno puede acceder a la sección Buscar.		X

Tabla 4.1. Estimaciones del sistema ciclo 1

Estimaciones del sistema en el ciclo 2

Pantallas a realizar:	Ciclo 1	Ciclo 2
El usuario consulta el menú principal.	X	
El usuario puede consultar el documento de la teoría básica del HC11.	X	
El usuario puede consultar la selección de las prácticas.	X	
El usuario puede tener acceso a cada una de las prácticas del HC11.	X	
El alumno puede ir al selector de manuales.	X	
El alumno puede ir a cada uno de los manuales.	X	
El alumno puede acceder a Acerca de.	X	
El alumno puede ir a la selección de software.	X	
El alumno puede acceder a los manuales de software.	X	
El alumno puede acceder a la Búsqueda de palabras y Manual de usuario.		X
El alumno puede ver las imágenes de cada una de las prácticas de laboratorio.		X
El alumno puede ver la presentación del Tutorial		X

Tabla 4.2. Estimaciones del sistema ciclo 2

NOTA: Todas las pantallas se realizarán por los dos miembros del equipo.

4.3.4 Forma Registro de Riesgos (Forma RR)

En la siguiente forma se detectan los riesgos posibles cuando algo puede o no suceder; para evitarlos o controlarlos, es conveniente tomarlos en cuenta desde el principio y determinar las precauciones adecuadas para ello.

Al establecer una estrategia se debe definir el criterio de la misma; identificar diferentes estrategias que se podrían utilizar; identificar riesgos y beneficios en cada una; hacer una evaluación comparativa de las mismas; decidir cuál utilizar y documentarla.

Las siguientes tablas muestran las Formas RR que corresponden a los ciclos de desarrollo.

Forma RR ciclo 1

Fecha	23 de enero de 2003
Riesgo o asunto	Software
Descripción	Posiblemente el software que se utiliza para simular programar y grabar del microcontrolador HC11 no se ejecute directamente del CD
Gravedad	Mediana
Asignado	Al equipo del proyecto del tutorial del microcontrolador HC11
Fecha de resolución	
Fecha	23 de enero de 2003
Riesgo o asunto	Individuales
Descripción	Que no tengamos los integrantes del equipo las bases suficientes (en cuanto a la programación) para el desarrollo del proyecto.
Gravedad	Mediana
Asignado	Al equipo del proyecto del tutorial del microcontrolador HC11
Fecha de resolución	
Fecha	23 de enero de 2003
Riesgo o asunto	Software y Interfaz de comunicación
Descripción	Que el software que se utiliza para la comunicación con el microcontrolador (PCBUG11) no sea compatible para versiones recientes de Windows utilizando la interfaz de la tarjeta del microcontrolador HC11
Gravedad	Mediana
Asignado	Al equipo del proyecto del tutorial del microcontrolador HC11
Fecha de resolución	

Tabla 4.3. Forma RR del ciclo 1

Forma RR ciclo 2

Fecha	26 de junio de 2003
Riesgo o asunto	Software
Descripción	Que el sistema no se pueda ejecutar aún en Windows 98.
Gravedad	Mediana
Asignado	Al equipo del proyecto del tutorial del microcontrolador HC11
Fecha de resolución	
Riesgo o asunto	Equipo del Laboratorio de Multimedia e Internet
Descripción	No contar con los recursos suficientes para el desarrollo del sistema, en cuanto a las prácticas del laboratorio (para la captura de imágenes).
Gravedad	Mediana
Asignado	Al equipo del proyecto del tutorial del microcontrolador HC11
Fecha de resolución	

Tabla 4.4. Forma RR del ciclo 2

4.3.5 Forma ESTRA

En la forma ESTRA se realiza la documentación correspondiente a la estrategia de desarrollo y está formada por:

- ❖ Nombre del integrante del equipo que la realizó.
- ❖ Fecha de realización.
- ❖ Nombre del equipo.
- ❖ Etapa del proyecto.
- ❖ Nombre del instructor.
- ❖ Número de ciclo de desarrollo al que corresponde.
- ❖ Número de referencia.
- ❖ Funcionalidades del sistema.
- ❖ Ciclo en que se desarrollará cada una de las funcionalidades.

Las Formas ESTRA se muestran a continuación:

Nombre	Miriam Sánchez Flores.	Fecha	26 de junio de 2003	
Equipo		Instructor	Honorato Saavedra	Hernández.
Parte	ESTRA	Ciclo		

Referencia	Funcionalidades	Horas por ciclo	
		1	2
1	El usuario puede acceder al documento de teoría básica del HC11.	X	
2	El usuario puede acceder a las prácticas del laboratorio del HC11.	X	
3	El usuario puede acceder a los manuales del HC11.	X	
4	El usuario puede utilizar el software del tutorial.	X	
5	El usuario puede acceder a la Búsqueda de palabras y Manual de usuario.		X

Tabla 4.5. Forma ESTRA ciclo 1 y 2

4.4 Planeación

En la tercera etapa de la metodología TSPi se realiza principalmente el plan de desarrollo del proyecto y el plan de calidad que se aplicará para establecer los criterios de estándares de calidad con la finalidad de generar un producto de calidad.

4.4.1 Plan de calidad

El principal objetivo que tiene el plan de calidad para este proyecto es construir un producto con un mínimo de defectos del 10% en cada ciclo y con base en este porcentaje obtener una mejora en el sistema.

A continuación se mencionan los estándares de calidad que aplicarán al sistema a desarrollar.

Medida	Objetivo	Comentarios
Porcentaje libre de defectos		
Pruebas unitarias	> 50%	
Pruebas de integración	> 70%	
Pruebas del sistema	> 90%	
Defectos/KLOC		
Defectos totales inyectados	75-150	
Pruebas del sistema	< 0.2	Sólo defectos mayores
Tasas en el tiempo de desarrollo		
Inspección de requerimientos/Tiempo en requerimientos	> 0.25	Sólo tiempo utilizado en requerimientos, no se asigna tiempo dedicado al estudio
Revisión de código/Tiempo en codificación	> 0.5	
Tasas de revisión e inspección		
Páginas de los requerimientos/hora	< 2	Páginas de texto con interlineado sencillo
LOC/hora	< 200	LOC lógicas.

Tabla 4.6. Plan de Calidad

Donde:

- ❖ Porcentaje libre de defectos en esta parte podemos medir la compilación, las pruebas unitarias y de integración y las pruebas del sistema. En donde se verá que el sistema que el sistema no tenga errores a la hora de compilación probando individualmente y en conjunto.
- ❖ Defectos/KLOC

En esta parte haremos pruebas para ver los defectos tanto individualmente como en conjunto, además que se verán los errores en la compilación.

- ❖ Tasa de defectos

En esta parte se verán los defectos de revisión en DBN/defectos de pruebas unitarias y defectos de revisión de código/defectos de compilación.

Es decir que en esta etapa vamos a ver los defectos unitarios de nuestro sistema comparándolos con el diseño que hayamos establecido de una forma más concreta. Además de ver los posibles errores en el código.

- ❖ Tasa en tiempo de desarrollo:

Evaluaremos en sistema de la siguiente forma:

1. Inspección de requerimientos/tiempo en requerimientos.
2. DBN/tiempo en codificación.
3. Inspección en DAN/tiempo en DAN.
4. Revisión de código/tiempo en codificación.

En esta etapa revisaremos el diseño respecto al tiempo que establecimos, de la misma forma también veremos que los requerimientos se hayan cumplido.

- ❖ Tasas de revisión de inspección

Evaluaremos viendo las páginas de los requerimientos/hora, páginas DBN/hora, y LOC/hora.

4.4.2 Forma Resumen del Tamaño (RESTM)

Esta forma resume los datos para el tamaño del producto, en la cual se registra el nombre de quién realiza la forma, fecha, nombre del equipo, nombre del instructor, nombre de la etapa de desarrollo y número de ciclo; todo esto se muestra en el encabezado de la forma.

En cuanto al contenido, se puede observar el nombre del productos y sus partes, número de ciclo y medida del tamaño; que en este caso se tomo la unidad días para su realización.

Esta forma se representa en la siguiente tabla:

Nombre	Miriam Sánchez Flores.	Fecha	11 de julio de 2003
Equipo	HC11	Instructor	Honorato Saavedra H.
Parte	ESTRA2	Ciclo	2

Nombre del producto y partes	No. de ciclo	Medida del tamaño en días
PROGRAMAR LAS SECCIONES DEL TUTORIAL: TEORÍA, PRÁCTICAS, MANUALES Y SOFTWARE	1	60
INSERTAR IMÁGENES		
Recopilar imágenes para cada una de las prácticas	2	2
Adquirir en forma digital las imágenes.	2	1
Insertar las imágenes en el Tutorial.	2	3
ACCESAR A LA SECCIÓN BUSCAR		
Investigar cómo ejecutar la búsqueda de palabras en un documento.	2	2
Programar la sección Buscar	2	3
INSERTAR LA PRESENTACIÓN DEL TUTORIAL		
Programar	2	3

Tabla 4.7. Forma RESTM

4.4.3 Forma Tarea Semanal

Es importante mencionar que ésta forma surge como iniciativa del equipo de trabajo debido a que en ella se registraron las tareas semanales con el objetivo de contar con un plan semanal de actividades.

Se asignan valores estimados para cada actividad a realizar, al finalizar dicha actividad se registraron los valores reales de cada actividad para después calcular el tiempo total real semanal correspondiente a cada actividad, así como el tiempo total acumulado.

La Forma Tarea Semanal conformada por los siguientes puntos:

- ❖ Nombre de la actividad a desarrollar.
- ❖ Periodo correspondiente a cada semana.
- ❖ Número del ciclo de trabajo.
- ❖ Valor Estimado (VE) asignado a cada tarea [horas].
- ❖ Valor Real (VR) asignado a cada tarea [horas].
- ❖ Total del tiempo real correspondiente a cada actividad.
- ❖ Total acumulado de cada actividad en la semana.

ACTIVIDAD	LUNES		MARTES		MIÉRCOLES		JUEVES		VIERNES		TOTAL DEL TIEMPO REAL	TOTAL ACUMULADO	
	VE	VR	VE	VR	VE	VR	VE	VR	VE	VR			
Periodo: 13 octubre al 17de octubre 2003						Ciclo 2							
Programar la sección del material	0.5	0.5									0.5	11.5	
Probar sistema			2	0							0		
Actualizar CD			0.5	1							1		
Revisar etapa de Pruebas 2			1	0	1	0.45					0.45		
Realizar Pruebas ciclo 2					2	0	1	0			0		
Realizar etapa 2 de Pruebas							2	0	2	0	0	0	

Tabla 4.8. Forma Tarea Semanal

4.4.4 Diagrama de Gannt

El diagrama de Gannt es una herramienta muy útil en las técnicas de planificación del proyecto.

Las técnicas de planificación se ocupan de estructurar las tareas a realizar dentro del proyecto, definiendo la duración y el orden de ejecución.

Es una herramienta que posee Microsoft Project.

Una gráfica de Gannt es un diagrama bidimensional en el que se representan las diferentes actividades o tareas del proyecto (eje vertical) frente a eje de los tiempos necesarios para realizar las mismas (eje horizontal).

Cada una de las actividades del proyecto se muestra en el diagrama de Gannt mediante una barra horizontal, cuyo extremo izquierdo se corresponde con su inicio, viniendo la duración de la misma indicada por la longitud de la misma.

Los diagramas de Gannt correspondientes al sistema a desarrollar, los cuales se fueron actualizando dependiendo del ciclo de desarrollo en el que se realizó.

Estos diagramas se muestran en las siguientes figuras:

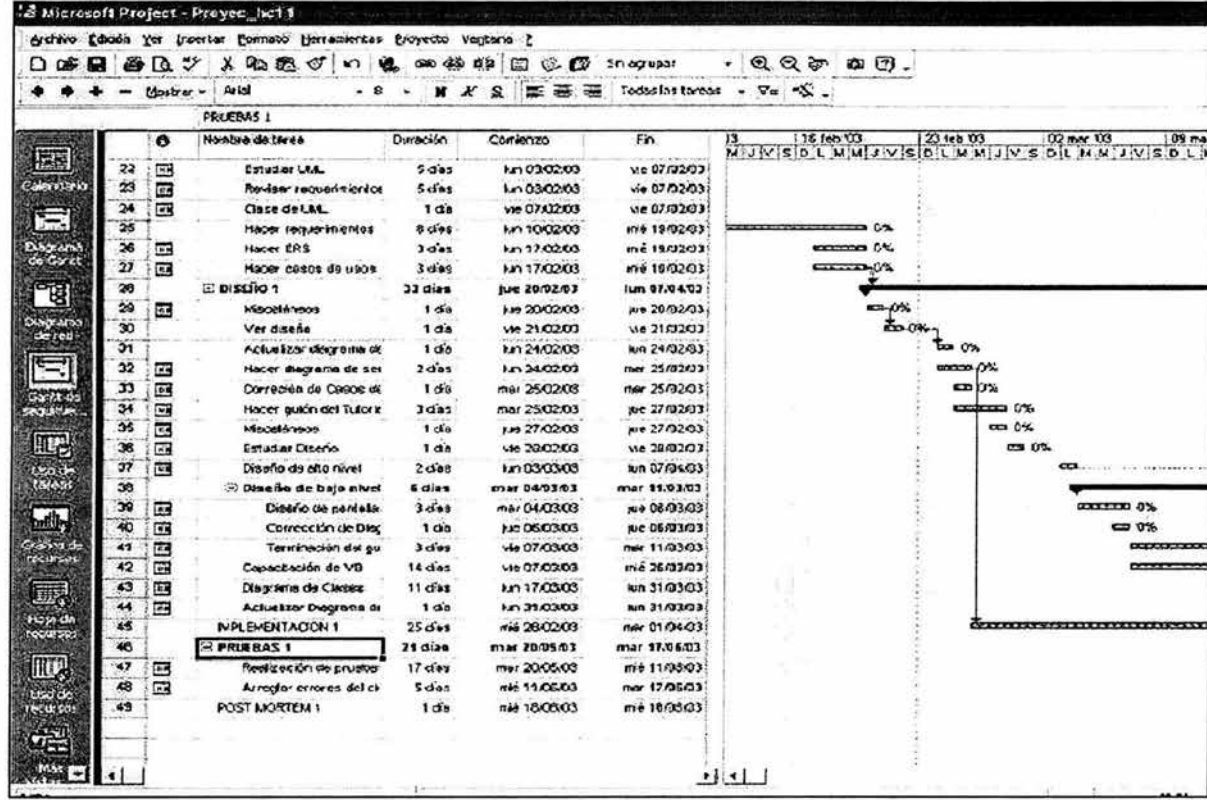


Figura 4.3 Diagrama de Gantt ciclo 2

4.5 Requerimientos

Durante la etapa Requerimientos se revisaron las necesidades del cliente y se formularon preguntas acerca de las funciones del producto. A través de este proceso se logró un acuerdo común entre el equipo acerca de lo que se va a construir; por esta razón, la participación de cada integrante fue importante en la definición de los requerimientos.

Existen diferentes tipos de requerimientos en un sistema a desarrollar:

- ❖ Requerimientos del sistema.
- ❖ Requerimientos del cliente.
- ❖ Requerimientos de interfaz.

4.5.1 Requerimientos del sistema

Desarrollar un sistema que ayude a los alumnos de la materia de microcomputadoras de la Facultad de Ingeniería a desarrollar sus prácticas de laboratorio.

4.5.1.1 Alcances del proyecto

El proyecto abarcará los siguientes aspectos:

1. Tutorial introductorio de microcontroladores.
2. Prácticas del laboratorio.
3. Manuales del microcontrolador y de software.
4. Software necesario par programar el microcontrolador.

Limitantes:

- ❖ El tutorial únicamente proporciona ayuda para la correcta realización de las prácticas sin proporcionar los resultados de éstas.
- ❖ El sistema no proporcionará el hardware necesario para la realización de las prácticas, solamente mencionará cual es el requerido.

4.5.1.2 Resultado final esperado

Un sistema multimedia en presentación de CD-ROM para computadoras con sistema operativo Windows XP y 2000.

4.5.1.3 Justificación y Beneficios

Se decidió desarrollar este software con base en los siguientes puntos:

- ❖ Por la dificultad de encontrar las prácticas y el software necesario, en condiciones óptimas de funcionamiento.
- ❖ Para contribuir con el personal docente en la tarea de llevar el conocimiento al alumnado mediante el uso de un nuevo material de trabajo.

Es importante el desarrollo de este sistema por la gran utilidad que tendrá para los alumnos y a su vez, por el hecho de que no existe un antecedente similar y además de que brinda a los desarrolladores nuevos conocimientos y mayor práctica en el área de los sistemas multimedia.

4.5.1.4 Identificación de los objetivos del Laboratorio de Multimedia e Internet que se relacionan con el proyecto

El proyecto cumple con el objetivo de formar recursos humanos, al poner en práctica los conocimientos adquiridos, para la realización del sistema multimedia, además de proporcionar una nueva ayuda a la comunidad, especialmente para el laboratorio de Microcomputadoras de la Facultad de Ingeniería.

4.5.1.5 Revisión de la situación actual

Hasta el momento se han encontrado varios sistemas o software que permite la programación de microcontroladores de los cuales podemos encontrar diferencias como son:

1. Versión del software.
2. Utilidad para un solo tipo de serie de microcontroladores.
3. Algunos con menor potencial que otro.

Sin embargo este tipo de software toma un papel complementario en el proyecto del tutorial y es éste último el que se ha buscado en otras escuelas o lugares de trabajo con microcontroladores y se llegó a la conclusión de que no existe software similar al que se desea diseñar.

En el aspecto de los procesos o metodología que se siguen actualmente en el laboratorio de microcomputadoras para la realización de las prácticas se encontró que cada profesor tiene una forma de llevar a cabo éstas y en ocasiones algunos no siguen ninguna y sólo se concretan a entregar el material para que el alumno desarrolle la práctica como crea que es correcto. Por lo tanto fue necesario crear un proceso general o global para guiar al alumno.

4.5.1.6 Definición de Requerimientos

Es necesario que el alumno tenga acceso a la siguiente información para la correcta realización de sus prácticas:

- ❖ Tutorial introductorio de cada práctica.
- ❖ Listado del material necesario para las prácticas del laboratorio.
- ❖ Conocer el uso de los simuladores y compiladores IASM11, HCSIM y JBUG11.
- ❖ Desarrollo de la práctica.
- ❖ Resultado final de la práctica al cual debe llegar el alumno.

Todo lo anterior se le facilitará al alumno mediante video, sonido, imágenes o material escrito que será proporcionado dependiendo de las consultas o peticiones que el usuario le solicite al tutorial.

En cuanto al desarrollo del sistema se tomaran en cuenta los siguientes aspectos importantes del software requerido:

1. El software para programar el microcontrolador HC11 están desarrollados para MS-DOS.
2. Windows nos permite correr los programas de MS-DOS.
3. Windows es el sistema operativo más común en el mercado y el más utilizado.
4. Al utilizar el sistema operativo Windows se decidió utilizar un lenguaje de programación el cuál nos facilite el manejo de multimedia y dicho lenguaje es Visual Basic.

En lo que se refiere a la elaboración del diseño conceptual se tomaron en consideración las siguientes funciones principales que el sistema debe realizar:

1. La función del tutorial para el usuario es proporcionarle la ayuda para la realización de sus prácticas.
2. Las prácticas estarán en archivos comprimidos y será posible copiarlos para su edición.
3. El tutorial incluirá otros paquetes que pueden ayudar en el desarrollo de las prácticas.

Otro punto importante en la definición de los requerimientos es la Identificación de los requisitos del cliente. Este proceso debe determinar:

a) La totalidad de los requisitos del producto y/o servicio del cliente:

- ❖ El cliente requiere el software necesario para la programación, compilación y simulación del microcontrolador HC11.
- ❖ El cliente requiere integrar todas las definiciones básicas de los elementos que componen el microcontrolador HC11.
- ❖ El cliente requiere los manuales para el manejo del microcontrolador HC11 y el software.

b) Los requisitos no especificados por el cliente pero necesarios para la adecuación a su propósito:

- ❖ Se recomienda una interfaz cómoda, accesible y fácil de entender.

c) Las obligaciones relacionadas con el producto y/o servicio, incluyendo los requisitos reglamentarios y legales:

- ❖ El software que se proporcionará para la programación, simulación y compilación del microcontrolador HC11 es software libre.

d) Los requisitos del cliente para la disponibilidad, entrega y el soporte del producto y/o servicio:

- ❖ Se requiere que todo el sistema se encuentre en CD o se pueda bajar en formato ISO de algún servidor FTP o Web.

4.5.2 Requerimientos del Cliente

En ésta sección se definen de una manera más amplia los requerimientos del cliente con los cuales el sistema debe cumplir:

En el Tutorial del Microcontrolador del HC11 el usuario podrá acceder a los diferentes módulos como son: teoría básica del microcontrolador HC11, las prácticas del laboratorio de Microcomputadoras, los manuales técnicos del microcontrolador HC11, el software de microcontrolador HC11.

A continuación se describen brevemente las secciones del sistema:

1. Teoría básica del microcontrolador HC11: el usuario podrá visualizar la información relacionada con el microcontrolador HC11, es decir, conceptos básicos, programación del CPU, lista de comandos, etc. Ver Anexo A.

2. Manuales del software del microcontrolador HC11:

- ❖ **Ensamblador IASM11:**
Donde encontrará información de como editar, ensamblar, guardar y cargar programas referentes a microcontrolador HC11.
- ❖ **Simulador HCSIM:**
Podrá encontrar información de como simular los programas realizados para los microcontroladores de la familia 68HC11; así como algunos comandos básicos donde podrá hacer las ejecuciones más rápidas sin ir al menú principal.
- ❖ **JBUG11:**
En este programa el usuario va a poder hacer la comunicación entre el microcontrolador y la PC, donde encontrará los comandos básicos para hacer dicha comunicación, además se le dará la información de como grabar los programas en la memoria del microcontrolador.

3. Prácticas del laboratorio de Microcomputadoras donde el alumno podrá ver cada una de ellas, consta de 10 prácticas abarcando el temario correspondiente. Ver Anexo B.

4. Manuales técnicos del MC68HC11 solamente los de la serie A, E y F los cuales van estar contenidos en tres bloques para que el usuario dependiendo el tipo del microcontrolador los pueda ocupar. Ver Anexo C.

5. Software que se utiliza para programar, simular, compilar y cargar el MC68HC11, el sistema contará con tres de los cuales son:

- ❖ **Ensamblador IASM11**

Donde va a poder editar, ensamblar, guardar y cargar programas referentes a microcontrolador HC11.
- ❖ **Simulador HCSIM**

Con el programa HCSIM se pueden simular los programas realizados para los microcontroladores de la familia 68HC11 en sus versiones; se pueden simular con otras versiones del MC68HC11, pero considerando algunas restricciones, como son la calidad de puertos que se tienen disponibles, así como las direcciones asignadas para ellos en el mapa de memoria del microcontrolador.

❖ JBUG11

En este programa el usuario va a poder hacer la comunicación entre el microcontrolador y la PC, donde podrá grabar los programas en la memoria del microcontrolador.

4.5.2.1 Diagramas de Casos de Uso

Una herramienta útil para la representación de la fase de requerimientos son los diagramas de caso de uso, que en el capítulo 2 se describieron con detalle.

A continuación se describen brevemente los cuatro casos de uso y se representan en el diagrama correspondiente:

1. Caso de uso: Acceso a la teoría básica del microcontrolador HC11.

El usuario al acceder a la teoría básica podrá ver y descargar la información del microcontrolador MC68HC11, en el cual también puede elegir algún tema con respecto a este microcontrolador.

2. Caso de uso: Acceso a las prácticas del Laboratorio de Microcomputadoras.

El usuario en esta sección podrá consultar cada una de las prácticas correspondientes al Laboratorio de Microcomputadoras, el material necesario para cada una de éstas; así como la teoría previa necesaria para la realización de cada una de las prácticas. Y si así lo desea será posible descargar la información de éstas.

3. Caso de uso: Acceso a los manuales del microcontrolador HC11.

El usuario al ingresar en esta parte podrá tener acceso a cada uno de los manuales del microcontrolador MC68HC11, y del Software a utilizar; además que podrá descargarlos.

4. Caso de uso: Acceso al software del microcontrolador HC11.

El usuario al ingresar a la sección de Software, podrá descargar cada programa que dispone el tutorial; esto le permitirá simular, compilar, programar y cargar el MC68HC11. Además que podrá ejecutarlos sin necesidad de descargarlos. Tendrá la opción de consultar el manual del software si así lo requiere.

El diagrama general de caso de uso se muestra a continuación:

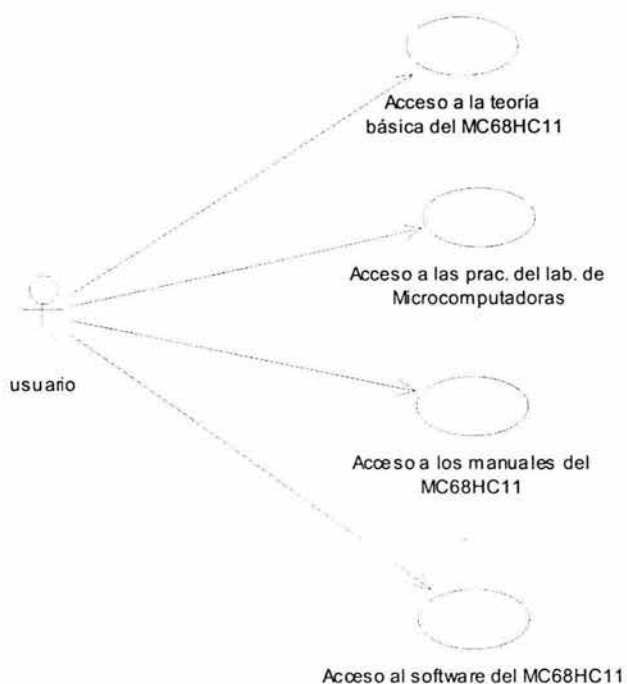


Figura 4.4 Diagrama general de caso de uso.

De igual manera cada uno de los casos de uso principales se subdivide:

1. Caso de uso: Acceso a la Teoría básica del microcontrolador HC11.

- ❖ El usuario puede ver toda la información de la teoría básica del microcontrolador HC11.
- ❖ El usuario puede hacer una búsqueda por tema en la teoría básica del HC11.
- ❖ El usuario podrá descargar archivos (teoría básica del microcontrolador HC11).

La representación del diagrama se muestra en la siguiente figura:

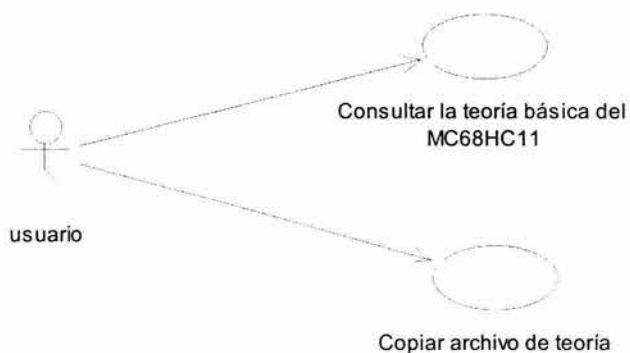


Figura 4.5 Diagrama del caso de uso Teoría.

2. Caso de uso: Acceso a las Prácticas del Laboratorio de Microcomputadoras.

- ❖ El usuario puede consultar cada una de las prácticas en forma individual, así como el material, el desarrollo y la teoría previa para cada una de estas.
- ❖ El usuario puede ir a la teoría previa para cada una de las prácticas.
- ❖ El usuario puede descargar archivos (prácticas del Laboratorio de Microcomputadoras).

Su representación se muestra en la siguiente figura:

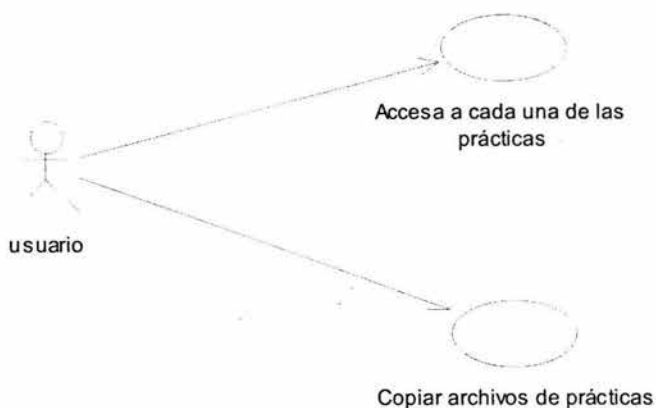


Figura 4.6 Diagrama del caso de uso Prácticas.

3. Caso de uso: Acceso a los Manuales del microcontrolador HC11.

- ❖ El usuario puede consultar cada uno de los manuales técnicos del Microcontrolador HC11.
- ❖ El usuario puede consultar cada uno de los manuales del software del Microcontrolador HC11.
- ❖ El usuario puede descargar archivos (manuales del microcontrolador HC11).

A continuación se muestra el diagrama correspondiente en la figura siguiente:

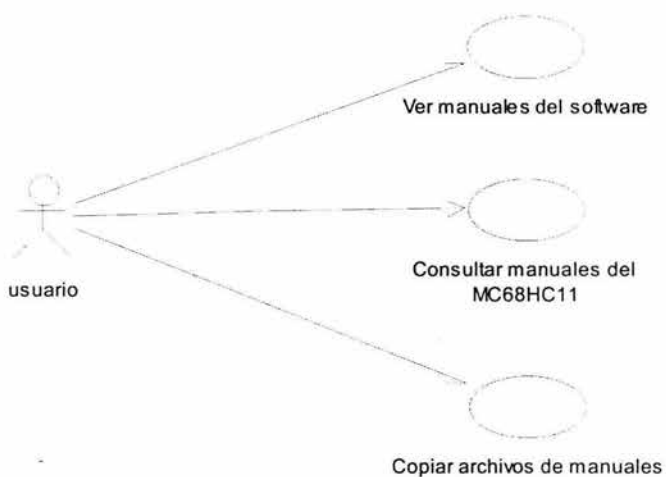


Figura 4.7 Diagrama de caso de uso Manuales.

4. Caso de uso: Acceso al Software del Microcontrolador HC11.

- ❖ El usuario puede descargar archivos (software del HC11).
- ❖ El usuario puede ejecutar de forma individual el software del HC11.
- ❖ El usuario puede consultar los manuales del software del HC11.

La siguiente figura representa el diagrama correspondiente a este caso de uso:

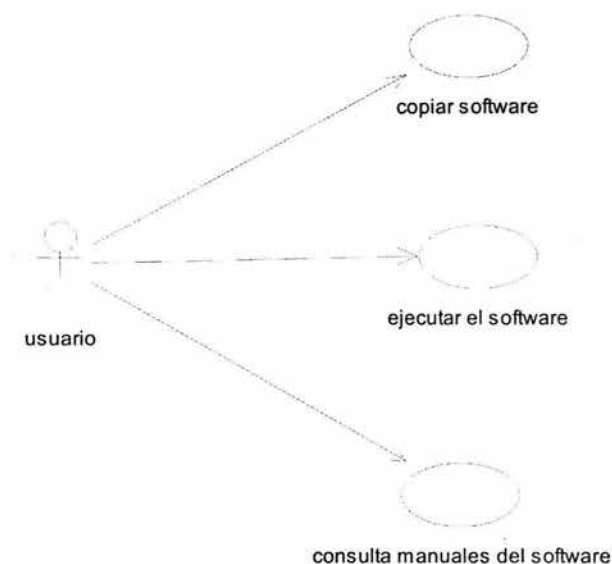


Figura 4.8 Diagrama de caso de uso Software.

4.5.2.2 Diagramas de Actividad

Dentro de los diagramas que se utilizan dentro del Lenguaje Unificado de Modelado, se encuentran los diagramas de actividad, los cuales representan la coordinación de cada una de las actividades que se realizarán en cada uno de los casos de uso que componen el sistema a desarrollar.

En esta sección se observarán los diagramas de actividades correspondientes a la Teoría, Prácticas, Manuales, Software, Buscar y Copiar Archivos.

a) Diagrama de actividad: Teoría.

El diagrama de actividades correspondiente a Teoría está constituido por una etiqueta que identifica al diagrama. Existe un botón que indica el inicio de la actividad, a continuación se realiza la selección del botón Teoría y posteriormente se despliega la información correspondiente al botón seleccionado.

Lo anterior se representa en la siguiente figura:



Figura 4.9 Diagrama de actividades Teoría.

b) Diagrama de actividad: Prácticas.

Este diagrama representa las actividades que se realizan para la consulta de una práctica: se selecciona el botón Prácticas, a continuación se despliega una pantalla para elegir una práctica y después se desplegará la información relacionada.

A continuación se muestra en la figura estas actividades:



Figura 4.10 Diagrama de actividades Prácticas.

c) Diagrama de actividad: Manuales.

En cuanto al diagrama de Manuales se tienen las siguientes actividades: seleccionar el botón Manuales, seleccionar un manual y desplegar la información respectiva.

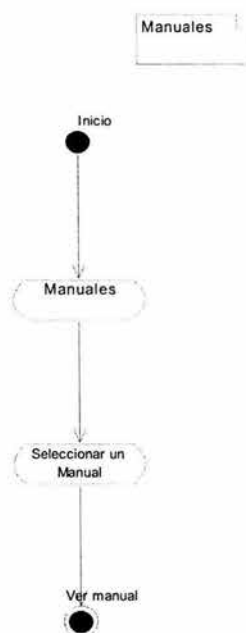


Figura 4.11 Diagrama de actividades Manuales.

d) Diagrama de actividad: Software.

Para el diagrama de actividades del software se tiene lo siguiente: seleccionar el botón software, a continuación seleccionar una opción de software, después se toma la decisión para elegir ver el manual correspondiente al software seleccionado o ejecutar el software seleccionado; ya tomada la decisión se visualiza la información.

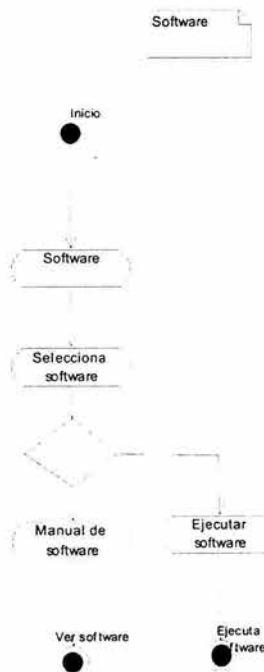


Figura 4.12 Diagrama de actividades Software

e) Diagrama de actividad: Buscar.

Las actividades a desarrollar en el diagrama Buscar son: elegir el botón Buscar, introducir un tema o palabra que se desea buscar, se selecciona una opción de búsqueda y finalmente se toma la decisión para visualizar la información o se cancela dicha búsqueda.



Figura 4.13 Diagrama de actividades Buscar.

f) Diagrama de actividad: Copiar archivos

Para realizar la copia de archivos es necesario seleccionar el botón Copiar, a continuación seleccionar el archivo que se desea copiar y posteriormente se elige la ruta destino para copiar el archivo, esto se realiza por medio del botón Examinar y a continuación se realiza la copia haciendo click en el botón copiar. Si no se desea copiar el archivo después de seleccionarlo, se elegirá el botón Cancelar.



Figura 4.14 Diagrama de actividades Copiar archivos.

4.5.2.3 Diagramas de Secuencia

El diagrama de Secuencia es uno de los diagramas más efectivos para modelar interacción entre objetos en un sistema. Un diagrama de secuencia se modela para cada caso de uso.

Un diagrama de secuencia muestra los objetos que intervienen en el escenario con líneas discontinuas verticales, y los mensajes pasados entre los objetos como vectores horizontales. Los mensajes se dibujan cronológicamente desde la parte superior del diagrama a la parte inferior; la distribución horizontal de los objetos es arbitraria.

a) Diagrama de secuencia: Teoría.

El usuario ingresa al Menú, donde elige la opción teoría básica, se muestra la teoría básica, el usuario se encuentra dentro de la teoría básica, el usuario puede visualizar la información o copiar el archivo correspondiente. Si el usuario decide copiar el archivo correspondiente elegirá el botón para copiar y el sistema responderá al mostrar una ventana.

El usuario también puede elegir salir de ésta sección o cerrar por completo la aplicación.

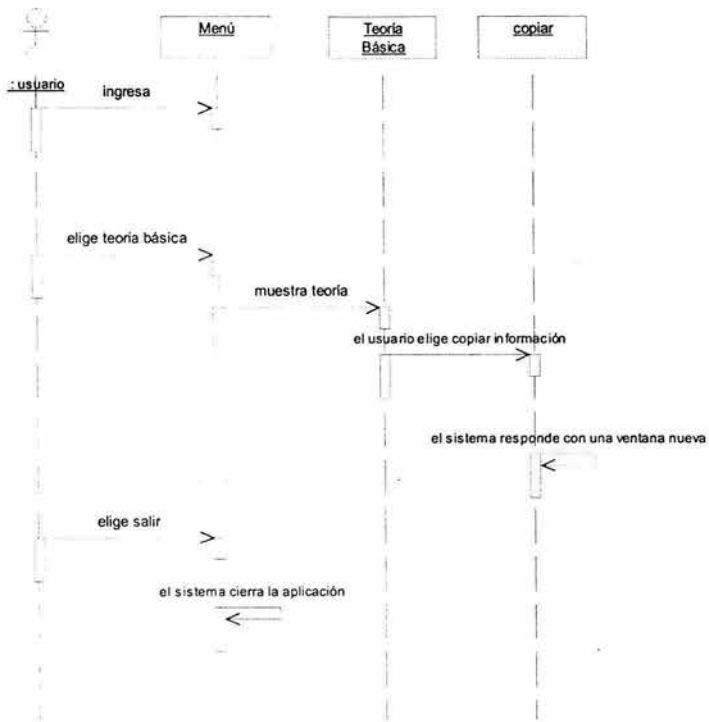


Figura 4.15 Diagrama de secuencia Teoría

b) Diagrama de secuencia: Prácticas.

El usuario se encuentra en el Menú donde elige la opción Prácticas, el sistema responde mostrando las distintas opciones, donde el usuario podrá elegir una práctica, el sistema mostrará la información correspondiente.

El usuario también podrá elegir ver el material correspondiente estando en la ventana que se despliega la información, el sistema mostrará el material correspondiente.

El usuario elegirá escuchar el resultado de la práctica, dentro de la misma ventana y a continuación el sistema mostrará el resultado.

El usuario podrá elegir ejecutar una opción de software y el sistema mostrará la ventana correspondiente.

El usuario podrá elegir copiar el archivo correspondiente a ésta práctica por lo que el sistema mostrará la ventana Copiar.

El usuario también puede elegir salir de ésta sección o cerrar por completo la aplicación.

La figura 4.16 que se muestra a continuación corresponde al diagrama de secuencia Prácticas.

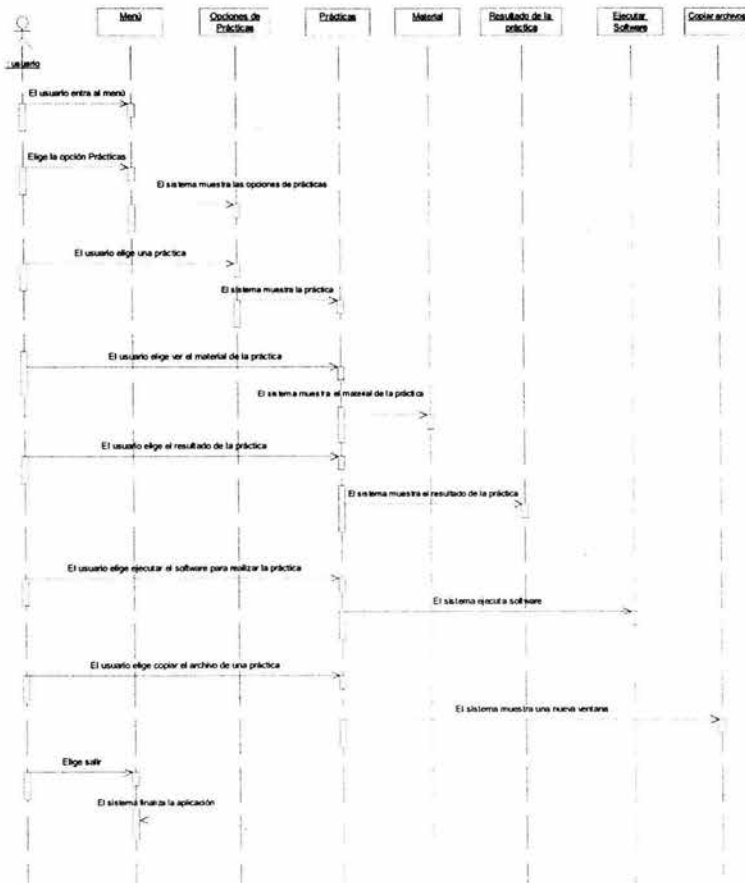


Figura 4.16 Diagrama de secuencia Prácticas

c) Diagrama de secuencia: Manuales.

El usuario tiene acceso al Menú donde podrá elegir la opción Manuales, el sistema mostrará las opciones que posee de manuales.

El usuario elegirá una opción de manuales, ya sea del microcontrolador o de software, el sistema mostrará la información de dicho manual.

El usuario elegirá copiar un archivo correspondiente a un manual, ya sea del microcontrolador o un manual de software, por lo que el sistema responderá al mostrar la ventana Copiar.

El usuario también puede elegir salir de ésta sección o cerrar por completo la aplicación.

Este diagrama se encuentra representado en la figura siguiente:

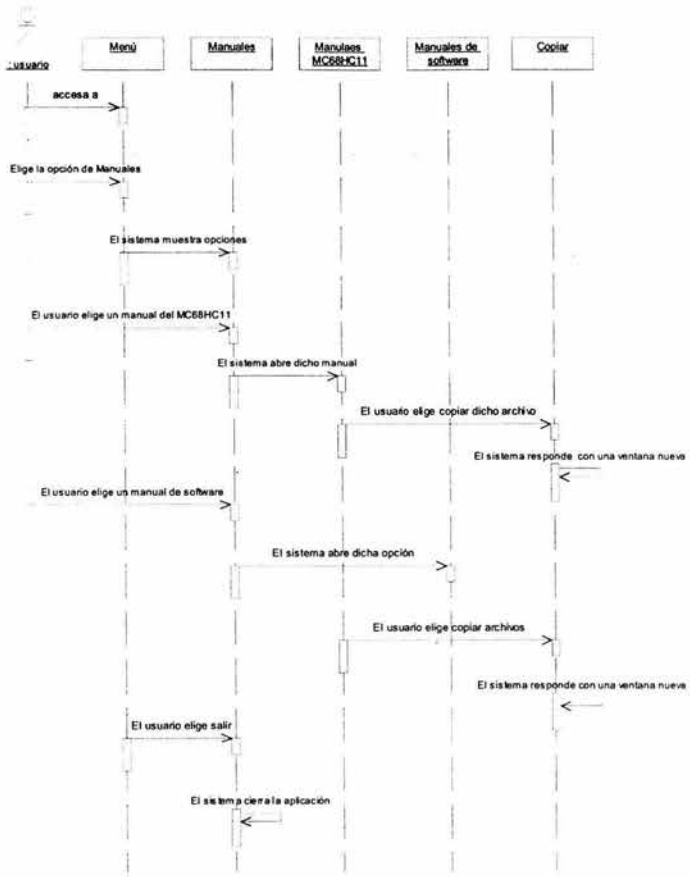


Figura 4.17 Diagrama de secuencia Manuales

d) Diagrama de secuencia: Software.

El usuario se encuentra en el Menú principal donde elige la opción Software, el sistema muestra las distintas opciones de software que posee.

El usuario selecciona una opción de software y elegirá ver el manual correspondiente por lo que el sistema mostrará su contenido.

El usuario selecciona una opción de software y elegirá ejecutar el software por lo que el sistema mostrará la ventana correspondiente.

El usuario elige copiar el archivo correspondiente al software y el sistema mostrará una ventana para realizar ésta acción.

El usuario también puede elegir salir de ésta sección o cerrar por completo la aplicación.

Este diagrama se muestra a continuación:

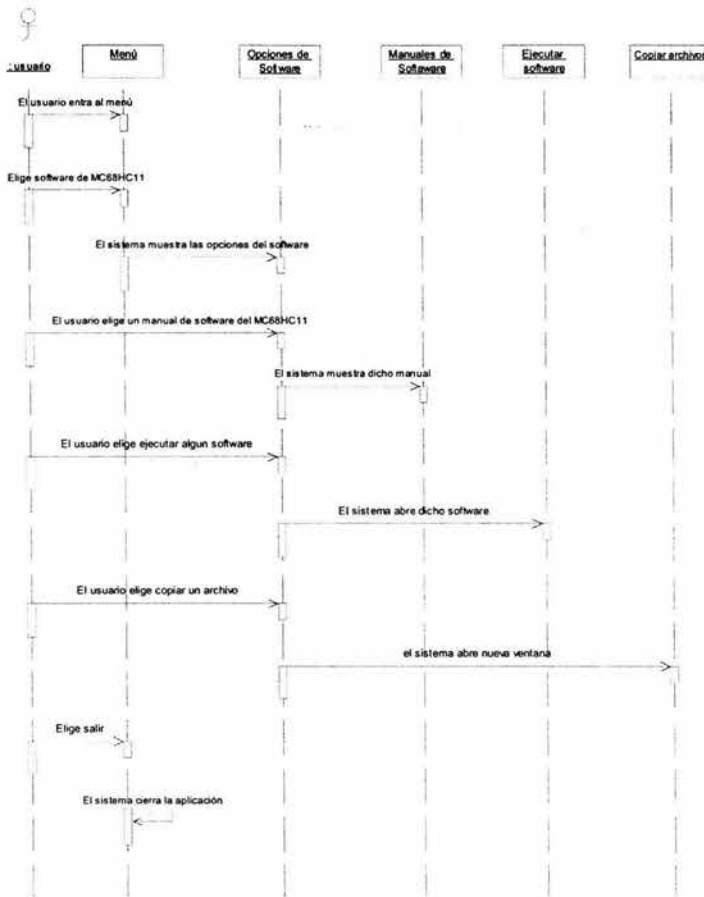


Figura 4.18 Diagrama de secuencia Software

e) Diagrama de secuencia: Buscar.

El usuario se encuentra en el Menú, donde selecciona la opción Buscar, el sistema muestra la ventana correspondiente.

El usuario escribe un tema o palabra a buscar, el usuario acciona la búsqueda y el sistema desplegará la información.

El usuario también puede elegir salir de ésta sección o cerrar por completo la aplicación.



Figura 4.19 Diagrama de secuencia Buscar

f) Diagrama de secuencia: Copiar archivos.

El usuario elige copiar un archivo, el sistema abre una nueva ventana.

El usuario elige el archivo a copiar.

El usuario elige la opción examinar para elegir la ruta destino de la copia.

El usuario elige la opción copiar para que se realice la copia del archivo elegido.

El usuario también puede elegir salir de ésta sección o cerrar por completo la aplicación.

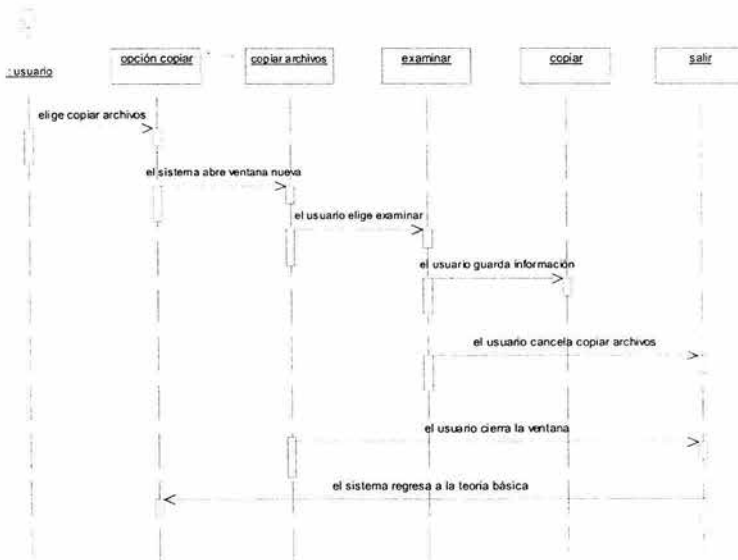


Figura 4.20 Diagrama de secuencia Copiar archivos

4.5.3 Requerimientos de Interfaz

El usuario deberá instalar el Tutorial del microcontrolador del HC11 y para hacer uso de él en sesiones subsecuentes sólo será necesario insertar el CD debido a que es fácil de utilizar, es portátil y además posee una gran capacidad de almacenamiento.

El usuario al insertar el CD podrá acceder al Menú principal donde podrá navegar a diferentes secciones del Tutorial como son:

- ❖ La documentación de la teoría básica del microcontrolador HC11.
- ❖ Las prácticas del Laboratorio de Microcomputadoras.
- ❖ Los manuales del microcontrolador HC11.
- ❖ El software del microcontrolador HC11.
- ❖ La búsqueda de un tema específico en la teoría básica.
- ❖ La información acerca del sistema.

En cuanto al formato en pantalla se usará una ventana de 700 X 500 pixeles, la cual se podrá minimizar para que el usuario tenga una mejor apreciación de la aplicación independientemente del tipo de resolución en el monitor que tenga.

4.5.3.1 Requisitos del sistema para el desarrollo de la interfaz

Los requisitos del sistema para la utilización de la aplicación son los siguientes:

- ❖ Computadora:
 - Microprocesador Pentium 166 Mhz o superior.
 - Pantalla VGA de 800 X 600 o de resolución superior compatible con Microsoft Windows.
 - 24 MB de RAM para Windows 2000.
 - Sistema operativo Windows 2000 o Windows XP.
 - Requisitos de espacio en disco duro:
 - a) Edición estándar: instalación típica 48 MB, instalación completa 80 MB.
 - b) Edición Profesional: instalación típica 48 MB, instalación completa 80 MB.
 - Lectora de CD ROM.
 - Tarjeta de sonido.
 - Ratón.
 - Teclado.

4.5.3.2 Restricciones en diseño/implementación

Las restricciones que se tienen en cuanto al diseño de la interfaz y su implementación se encuentran relacionadas con los acuerdos estándares siguientes:

- ❖ La interfaz de usuario se realizará en Visual Basic.
- ❖ Todo el sistema a desarrollar estará contenido en un solo CD.
- ❖ Se utilizará programación orientada a objetos.
- ❖ El usuario sólo podrá ejecutar la aplicación por medio de Windows XP o Windows 2000.

La interfaz de usuario se realizará en el lenguaje de programación Visual Basic porque nos permite crear una interfaz gráfica; y además de poseer las siguientes características para su programación:

Producir un código legible, detección rápida de posibles errores.

En cuanto a la programación orientada a objetos nos permite modelar elementos del mundo real, además de reutilizar código y encapsulamiento.

El usuario sólo podrá ejecutar la aplicación en Windows XP o 2000 debido a que el tutorial está diseñado para estos sistemas operativos y por la facilidad que proporcionan para instalar aplicaciones de este tipo.

Características del equipo para el desarrollo del sistema:

❖ Computadora 1:

- Sistema operativo Windows 2000.
- Procesador Pentium IV 1.4 GHz.
- 128 MB en RAM.
- 40 GB en disco duro.
- Tarjeta de video ATI ALL IN WONDER 32 Mb.
- Acelerador gráfico.
- CDRW 12X8X32.
- DVD 16X.
- Tarjeta de red integrada.
- Monitor a color 17".
- Teclado estilo multimedia.
- Ratón.
- Tarjeta de sonido integrada.
- Bocinas.

❖ Computadora 2:

- Sistema operativo Windows XP.
- Procesador: AMD Athlon(TM) 1.73 Ghz.
- 256 MB en RAM.
- 40 GB en disco duro.
- Tarjeta de video integrada.
- Tarjeta de sonido integrada
- Tarjeta de red integrada.
- Monitor a color 15".
- Teclado.
- Ratón.
- Bocinas Acteck.

Se utilizará este equipo debido a que cuenta con las características antes mencionadas y el software necesario para desarrollar el tutorial y además nos permite darle al sistema una mejor resolución de audio y video.

Cabe mencionar que es el equipo con el que cuenta el Laboratorio de Multimedia e Internet para el desarrollo del sistema.

Otros aspectos importantes relacionados con las restricciones de diseño e implementación son los estándares de documentación para llevar un orden y una estructura en los documentos a desarrollar. Además de apegarse a un estándar de codificación.

4.5.3.3 Criterios de medición

Los criterios de medición tienen por objetivo establecer los criterios mediante los cuales se logra alcanzar la buena calidad del proyecto.

Se puede alcanzar el objetivo mediante una encuesta realizada después de seis meses de haberlo puesto a disposición de alumnos y maestros

Además que el objetivo del laboratorio de Multimedia e Internet se verá cumplido si la metodología para el desarrollo de éste tutorial, para que sirva de base para sistemas posteriores.

Los objetivos tangibles son:

- ❖ Tener en forma física el CD-ROM cuyo contenido sea el tutorial con todos sus componentes.
- ❖ Tener el procedimiento necesario para el desarrollo de un sistema multimedia incluyendo la documentación necesaria.

Y el objetivo no tangible es:

- ❖ Adquirir experiencia en el desarrollo de sistemas multimedia.

La meta a la cual se desea llegar es:

- ❖ El tutorial podrá ser utilizado por toda aquella persona que quiera información sobre el microcontrolador HC11 para realizar las prácticas de Laboratorio de Microcomputadoras.

Además se establecieron criterios de adicionales de evaluación para los siguientes puntos:

- ❖ Facilidad de uso.
- ❖ Confiabilidad.
- ❖ Capacidad de mantenimiento: no aplica porque el sistema no recibe mantenimiento.
- ❖ Extensibilidad y flexibilidad.
- ❖ Seguridad y eficiencia.
- ❖ Actualidad de información.
- ❖ Habilidad para comunicarse con otros sistemas: esta opción no aplica ya que no la requiere.

El tutorial será probado por 10 personas que tengan poca experiencia en el manejo de computadoras, después de 30 minutos se les aplicará el cuestionario de evaluación que se muestra a continuación:

Cuestionario de evaluación

1.- ¿La información que busca en el tutorial la encontró en forma rápida?

a) SI__ b) NO__

Conteste lo siguiente sólo si respondió NO

¿Porqué?_____

2.- ¿Está de acuerdo con la distribución de la información?

a) SI__ b) NO__

Conteste lo siguiente sólo si respondió NO

¿Porqué?_____

3.- ¿Los botones simbolizan la información que contiene?

a) SI__ b) NO__

Conteste lo siguiente sólo si respondió NO

¿Porqué?_____

4.- ¿Tuvo algún problema cuando estaba ejecutando el tutorial?

a) SI__ b) NO__

Conteste lo siguiente sólo si respondió SI

¿Porqué?_____

5.- Podría decir, ¿qué le fue de utilidad el Tutorial para la asignatura de Microcomputadoras?

a) SI__ b) NO__

Conteste lo siguiente sólo si respondió NO

¿Porqué?_____

6.- ¿Le pareció interesante el Tutorial?

a) SI__ b) NO__

Conteste lo siguiente sólo si respondió NO

¿Porqué? _____

7.- ¿La información se presenta en forma clara?

a) SI__ b) NO__

Conteste lo siguiente sólo si respondió NO

¿Porqué? _____

8.- ¿La información que se despliega cuando solicitó la sección Buscar era la adecuada?

a) SI__ b) NO__

Conteste lo siguiente sólo si respondió NO

¿Porqué? _____

9.- ¿Le gustaría que se difundiera el tutorial?

a) SI__ b) NO__

Conteste lo siguiente sólo si respondió SI

¿De qué manera? _____

10.- Sugerencias:

4.6 Diseño

En esta etapa de la metodología TSPi se desarrolla el guión multimedia del sistema, de acuerdo a los puntos que se establecieron en el capítulo 3 correspondiente a los sistemas multimedia.

4.6.1 Guión del Tutorial del Microcontrolador HC11

Al dar inicio la aplicación se presentará dos áreas fijas que permanecerán de esta forma durante toda la aplicación.

Una de estas secciones se encontrará en la parte izquierda de la pantalla, que estará asignada al menú principal.

En la parte superior de la ventana también se encontrará una sección que se destinará para la barra de título.

Existirá una sección llamada área dinámica, la cual es nombrada de esta manera debido a que se modificará de acuerdo a la opción del menú principal que se haya elegido.

La sección de Menú principal contendrá los botones correspondientes a las diferentes secciones del tutorial además de un botón "Salir".

Las opciones que estarán representadas en este menú son las siguientes: Teoría básica del Microcontrolador HC11, Prácticas del Laboratorio, Manuales, Software, Acerca de, Buscar y Salir.

En la sección dinámica estará presente el slogan "Bienvenido al Tutorial del Microcontrolador HC11. Elige una opción del menú".

Para elegir la opción deseada será necesario dar un click en el botón de Menú principal que lo llevará a cada una de las secciones de dicho menú.

En la barra de título de la ventana aparecerá el nombre del sistema Tutorial del Microcontrolador HC11, donde también aparecerán dos iconos en la parte derecha para indicar que se puede minimizar y cerrar la ventana.

4.6.1.1 Ventana Teoría Básica

Si el usuario se posiciona sobre el botón Teoría; que se encuentra en la sección de menú principal, este mostrará un mensaje que se desplegará en forma de recuadro con una descripción breve de lo que el usuario puede encontrar en esta sección.

La descripción es la siguiente: "Encontrarás toda la información teórica del microcontrolador HC11".

En el momento que el usuario de clic en el botón Teoría básica, este resaltará más que los otros, para indicar que opción ha sido seleccionada y en el área dinámica, inmediatamente aparecerá la información correspondiente a la teoría básica de microcontrolador HC11.

En la barra de título de la ventana aparecerá el nombre del sistema Tutorial del Microcontrolador HC11, donde también aparecerán dos iconos en la parte derecha para indicar que se puede minimizar y cerrar la ventana.

El usuario podrá tener acceso a un submenú, el cual aparecerá en la parte inferior del área dinámica y contará con la opción para copiar el archivo correspondiente a la teoría básica del microcontrolador. En el momento que el usuario haga clic en la opción de copiar, se abrirá una nueva ventana, para especificar la ruta para copiar dicho archivo.

Al desplegarse la información en la pantalla dinámica de la teoría básica, ésta ventana tendrá una barra de desplazamiento para poder visualizar el contenido de la información. Esta información estará presentada en forma de ligas para tener acceso más rápido a la información.

4.6.1.2 Ventana Copiar

Contendrá una zona de título con la palabra "Copiar" en la parte superior de esta nueva ventana y dos iconos para minimizar y cerrar esta ventana.

En el interior se podrá seleccionar el archivo que se desea copiar, es decir que si está en teoría básica, solamente podrá copiar información relacionada con la teoría básica; pero si se encuentra en Prácticas de laboratorio, el usuario podrá seleccionar una o las que desee de manera individual.

Cabe mencionar que existen otras opciones para copiar como son: Software y Manuales, por lo que cuando se encuentre en la opción de Software sólo podrá copiar JBUG11, HCSIM y IASM11. En cambio para la sección de Manuales, se encontrará disponible la información relacionada con la serie A, E y F, además de los manuales correspondiente al software.

Se presentarán tres opciones al usuario para hacer uso de ellas, es decir, un botón de "Examinar" para elegir la ruta en donde se desea guardar el archivo, un botón de "Copiar" para que empiece a guardar la información y finalmente un botón "Cancelar" para cerrar la aplicación.

4.6.1.3 Ventana Buscar

Si el usuario se posiciona sobre el botón Buscar; que se encuentra en la sección de menú principal, este mostrará un mensaje que se desplegará en forma de recuadro con una descripción breve de lo que el usuario puede encontrar en esta sección.

La descripción es la siguiente: "Aquí podrás ingresar a la búsqueda de un tema relacionado con la teoría básica".

En el menú principal existe la opción de Buscar, la cual al ser accionada, se activará una nueva ventana no importando en que sección de la aplicación se

encuentre, ésta ventana tendrá en la barra de título la palabra Ayuda HC11 y dos botones para cerrar y minimizar.

La ventana de Buscar estará formada por distintas secciones en las que se podrá realizar la búsqueda de información.

Estas secciones son: Búsqueda por contenido, desplegándose así las distintas secciones; Búsqueda por medio del índice de temas y finalmente Búsqueda por palabras específicas.

Además se encuentra dentro de ésta sección el manual de usuario.

4.6.1.4 Ventana Prácticas

Si el usuario se posiciona sobre el botón Prácticas; que se encuentra en la sección de menú principal, este mostrará un mensaje que se desplegará en forma de recuadro con una descripción breve de lo que el usuario puede encontrar en esta sección.

La descripción es la siguiente: "Encontrarás las prácticas del laboratorio, así como toda la información necesaria para la realización de éstas".

En el menú principal existirá la opción Prácticas, que al darle clic en el botón correspondiente; en el área dinámica aparecerá un índice de prácticas a las que podrá tener acceso el usuario, por medio de un submenú que aparecerá en la parte inferior de esta sección dinámica.

Al seleccionar una práctica podrá copiar la información correspondiente, ya que contará con esa opción en el submenú y para ir al contenido de la práctica sólo será necesario darle clic en el submenú que se presenta en la parte inferior del área dinámica.

En la barra de título de la ventana aparecerá el nombre del sistema Tutorial del Microcontrolador HC11, donde también aparecerán dos iconos en la parte derecha para indicar que se puede minimizar y cerrar la ventana.

4.6.1.5 Ventana 2 Prácticas

En el momento en que el usuario haya elegido una práctica verá la información correspondiente.

En la parte inferior aparecerá un submenú con las siguientes opciones: copiar, resultado esperado, material y software para la práctica.

Al accionar la opción de "Copiar" podrá elegir una práctica o varias para guardarlas y sí así lo desea.

Al elegir la opción "Material" se presentará al usuario el contenido del material que necesita para la realización de su práctica. Esta información será presentada en forma de imágenes, las cuales representarán a cada uno de los componentes del material y cada una de estas imágenes contará con un nombre para identificarlos.

En la opción resultado esperado, el usuario podrá escuchar una descripción del resultado esperado de la práctica.

En la opción "Software" aparecerán las tres opciones para programar, simular y cargar el microcontrolador. Se abrirá una nueva ventana para cada una de las opciones anteriores que corresponderá al software seleccionado.

En la barra de título de la ventana aparecerá el nombre del sistema Tutorial del Microcontrolador HC11, donde también aparecerán dos iconos en la parte derecha para indicar que se puede minimizar y cerrar la ventana.

4.6.1.6 Ventana Software

Si el usuario se posiciona sobre el botón Software; que se encuentra en la sección de menú principal, este mostrará un mensaje que se desplegará en forma de recuadro con una descripción breve de lo que el usuario puede encontrar en esta sección.

La descripción es la siguiente: "En esta sección encontrarás el software necesario para la realización de tus prácticas".

Desde el menú principal el usuario podrá tener acceso a la opción "Software" donde aparecerán opciones en un submenú, en la parte inferior. Las opciones que podrá elegir son: copiar, ejecutar y ver manual de software.

En la opción "Copiar" podrá elegir el software: JBUG11, HCSIM y IASM11.

En la opción "Ejecutar" el usuario podrá ejecutar alguno de estos programas.

Y en la opción de "Manual" el usuario podrá visualizar a continuación en el área dinámica el contenido del manual de software

En la barra de título de la ventana aparecerá el nombre del sistema Tutorial del Microcontrolador HC11, donde también aparecerán dos iconos en la parte derecha para indicar que se puede minimizar y cerrar la ventana.

4.6.1.7 Ventana Manuales

Si el usuario se posiciona sobre el botón Manuales; que se encuentra en la sección de menú principal, este mostrará un mensaje que se desplegará en forma de recuadro con una descripción breve de lo que el usuario puede encontrar en esta sección.

La descripción es la siguiente: "En esta sección encontrarás los manuales del MC68HC11 serie A, E y F; así como los manuales del software que necesitas para la realización de tus prácticas".

El usuario al seleccionar la opción "Manuales" desde el menú principal, se mostrará en el área dinámica los diferentes tipos de manuales que podrán ser consultados (Manuales técnicos del microcontrolador HC11, serie A, E y F; además de los manuales de Software).

En la barra de título de la ventana aparecerá el nombre del sistema Tutorial del Microcontrolador HC11, donde también aparecerán dos iconos en la parte derecha para indicar que se puede minimizar y cerrar la ventana.

4.6.1.8 Ventana 2 Manuales

En el momento que el usuario haya elegido una opción de manual y accionado el botón correspondiente, se desplegará el contenido del manual en el área dinámica, además que en la parte inferior de la ventana dinámica (submenú) aparecerá la opción para copiar el archivo de cualesquiera de los manuales.

4.6.1.9 Ventana Acerca de

Si el usuario se posiciona sobre el botón "Acerca de"; que se encuentra en la sección de menú principal, este mostrará un mensaje que se desplegará en forma de recuadro con una descripción breve de lo que el usuario puede encontrar en esta sección.

La descripción es la siguiente: "Información relacionada con la producción del sistema".

Desde el menú principal el usuario podrá tener acceso a "Acerca de", en esta sección se mostrará la información relacionada con el sistema (producción, créditos, etc.)

Finalmente, el usuario desde el menú principal podrá seleccionar la opción "Salir", y al darle clic podrá salir automáticamente de la aplicación.

En la barra de título de la ventana aparecerá el nombre del sistema Tutorial del Microcontrolador HC11, donde también aparecerán dos iconos en la parte derecha para indicar que se puede minimizar y cerrar la ventana.

4.6.2 Especificaciones de la Interfaz del Tutorial del Microcontrolador HC11

Para el diseño de la interfaz se contemplaron las siguientes especificaciones:

La ventana de la aplicación será de 700 pixeles de ancho y 500 de alto pixeles, teniendo las siguientes secciones fijas, como se muestra en la figura:

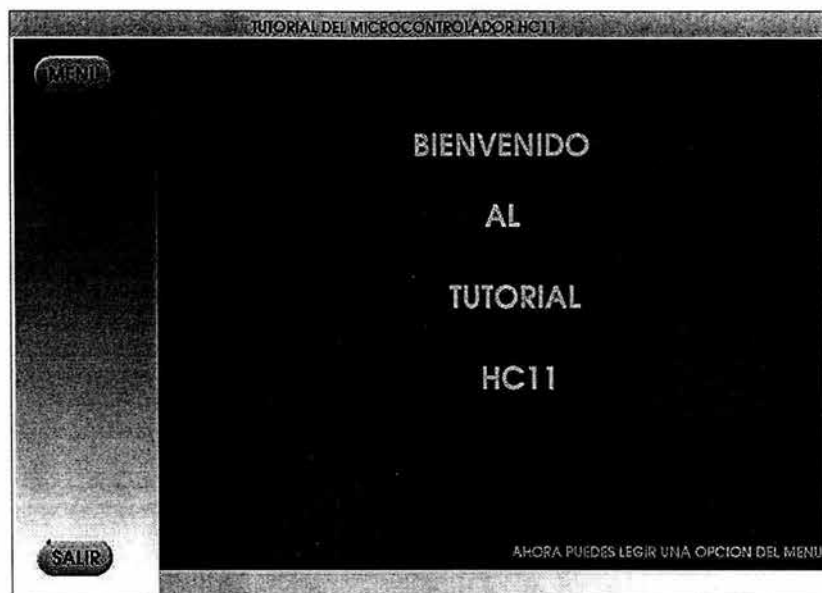


Figura 4.21 Ventana de la aplicación.

Las secciones serán las siguientes:

1. Área de título de un tamaño de 20 pixeles de ancho y 700 pixeles de alto.
2. Área de menú principal con un tamaño de 100 pixeles de ancho y 680 pixeles de alto.
3. Área dinámica con un tamaño de 450 pixeles de ancho y 600 pixeles de alto.
4. Área de submenú con un tamaño de 30 pixeles de ancho y 600 pixeles de alto.

Todos los botones de la interfaz del menú principal tendrán el mismo tamaño, es decir, 50 pixeles de alto y 80 pixeles de ancho, como se muestra en la figura:



Figura. 4.22 Botón.

En el menú principal habrán 7 botones del mismo tamaño y del mismo color que van hacer: Teoría, Prácticas, Manuales, Software, Buscar, Acerca de y Salir.

El submenú se encontrará en la parte inferior del área dinámica con un alto de 30 pixeles por 600 pixeles de ancho con un color diferente a los del menú principal. Y tendrán una acción al activarlos, los botones serán: Copiar, Ver, Material, Resultado, Ejecutar, Software. El tamaño de cada botón será de 30 pixeles de alto por 80 pixeles de ancho como se muestra en la figura:



Figura 4.23 Submenú

Los botones que aparecerán en el submenú dependerán de la sección en que se encuentre el usuario.

En la ventana Buscar tendrá un tamaño de 500 pixeles de alto por 500 pixeles de ancho como se muestra en la siguiente figura.

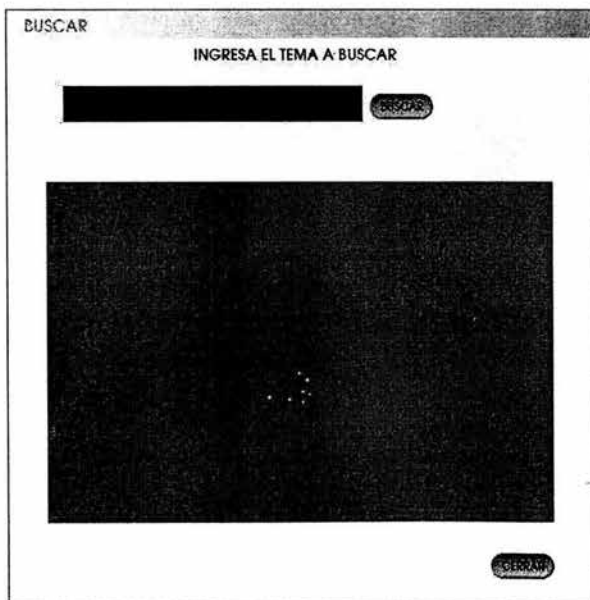


Figura 4.24 Ventana Buscar

Las secciones que tendrá esta ventana serán las siguientes:

- Área de título de un tamaño de 20 pixeles de ancho y 500 pixeles de alto.
- Área de búsqueda aparecerá en el centro de la ventana con un tamaño de 20 pixeles de ancho y 80 pixeles de alto esta será una caja de texto.
- Área donde aparecerá la información tendrá un tamaño de 400 pixeles de ancho y 300 pixeles de alto.

En la parte inferior derecha aparecerá el botón de cerrar y al lado de la caja de búsqueda el botón de buscar.

En la ventana de copiar tendrá una dimensión de 300 pixeles de alto por 200 pixeles de ancho, como se muestra en la figura:

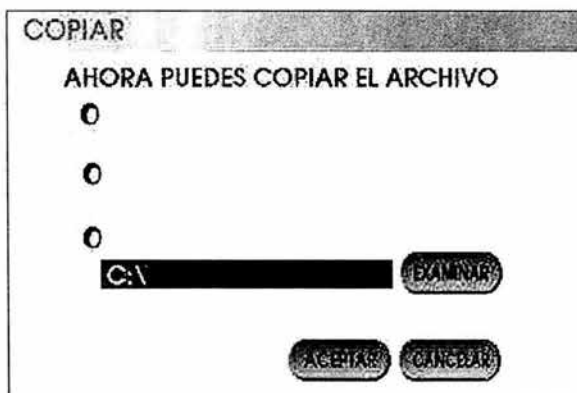


Figura 4.24 Ventana Copiar

Esta ventana tendrá:

- ❖ Una barra de título de un tamaño de 20 pixeles alto por 300 pixeles largo.
- ❖ Una caja de texto con una dimensión de 20 pixeles de alto y 150 de largo, esta nos servirá para indicar la ruta donde se copiará la información. Al lado de esta caja aparecerá el botón de examinar y en un nivel inferior se encontrarán los botones aceptar y cancelar.
- ❖ Los botones de la ventana Copiar tendrán la misma dimensión que los anteriores, es decir, de 30 pixeles de alto por 80 pixeles de ancho, estos botones serán: Examinar, Copiar y Salir.

Las especificaciones respecto al texto son las siguientes:

1. El tipo de letra que se utilizará será de tipo Arial con un tamaño de 12 puntos, para los títulos y 10 puntos para el resto de la información; de color negro.
2. La información de la teoría básica será en formato HTML.
3. Los manuales en formato PDF.
4. Aparecerá en cada ventana una imagen de fondo con el título de la sección en donde se encuentre.
5. Para elegir una opción en cualquier ventana se tendrán un botón de opción.

En cuanto a los estándares de archivos se consideraron las siguientes especificaciones:

1. El tipo de archivos utilizado en la Teoría básica se tendrán archivos de tipo HTML debido a que nos permitirá navegar de una manera más rápida; esto se hará por medio de ligas.
2. Para los manuales de software del HC11, el formato utilizado será PDF, porque es un formato estándar fácil de utilizar.
3. Para las prácticas el tipo de archivos para esta parte es HTML debido a que se podrá tener acceso más rápido.
4. En cuanto a los manuales del microcontrolador serán del tipo PDF porque los archivos de esa manera estarán protegidos contra modificaciones en su contenido.

Todos los archivos se encontrarán comprimidos para su descarga.

5. IMPLEMENTACIÓN DEL SISTEMA

En éste capítulo abarcarán las últimas etapas de la metodología TSPI, es decir, Implementación, Pruebas y finalmente Postmortem.

La fase de Implementación se realizó con base en el diseño que se describió en el capítulo cuatro, con el cual se podrá desarrollar las partes que finalmente contendrá el sistema.

En cuanto a la fase de Pruebas, se realizaron en dos etapas las pruebas al sistema realizadas por el equipo de desarrollo y las pruebas al sistema que se realizaron por usuarios de la facultad.

Finalmente en la fase de Postmortem se realizó una evaluación final del proyecto.

5.1 Implementación

Esta fase fue desarrollada de acuerdo a los requerimientos del cliente, es decir, contar con un Tutorial del HC11 para el Laboratorio de Microcomputadoras de la Facultad de Ingeniería.

Se tomó en cuenta que el sistema a desarrollar constaría de las siguientes secciones principales:

- ❖ Sección TEORÍA.
- ❖ Sección PRÁCTICAS.
- ❖ Sección MANUALES.
- ❖ Sección SOFTWARE.
- ❖ Sección BUSCAR.
- ❖ Sección COPIAR.
- ❖ Sección ACERCA DE.

Por lo que a continuación se realiza una descripción más detallada de cada una de ellas.

Como se ha mencionado anteriormente la implementación del sistema se realizó en el lenguaje de programación Visual Basic versión 6.0.

Presentación del sistema

En lo que se refiere a la presentación propia del sistema consiste en un formulario denominado formulario principal.

Se consideran dos secciones importantes dentro del diseño: el área de menú y el área de dinámica.

La primera se compone de las siguientes secciones:

a) Área Menú Principal

Esta sección de la interfaz está relacionado estrechamente con el área dinámica y el área de submenú, ya que será el vínculo principal para explorar el tutorial.

b) Área dinámica

La característica de la ventana dinámica es como su nombre lo indica, actualizarse de acuerdo a la sección que en el Menú principal se solicite. Además que se encuentra otra sección llamada Área Submenú, que posteriormente se presentará.

Por lo que se crea la clase cAreadinamica para abrir la información en el área dinámica y mostrar los botones que se encuentran en el menú principal.

A continuación se muestra la clase cAreadinamica con sus atributos y métodos correspondientes:

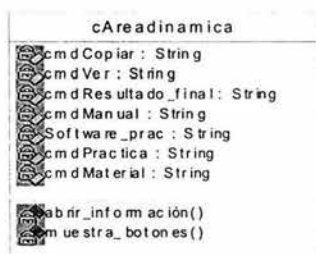


Figura 5.1 Clase cAreadinamica

En cuanto a la presentación del sistema se cuentan con los siguientes elementos:

Esta primera ventana mostrará el área Menú principal, una barra de título, y en el área dinámica la bienvenida al tutorial.

La clase `cCargar_frmPrincipal` se creó para establecer las características de posición, apariencia y dimensiones de la interfaz.

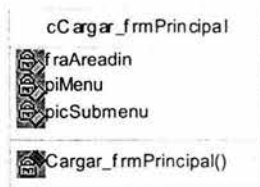


Figura 5.2 Clase `cCargar_frmPrincipal`

A partir de ésta ventana se podrá acceder automáticamente a cada una de las secciones del sistema, mediante los botones que se encuentran dentro del Menú principal.

La función que tiene en éste caso el área dinámica es mostrar la presentación de bienvenida formada por imágenes. Por lo que se crea la clase `cPicbienvenido`.

Su representación se muestra en la figura siguiente:



Figura 5.3 Clase `cPicbienvenido`

En cuanto a los botones correspondientes al menú principal, se mostrarán mensajes dinámicos relacionados con el contenido de cada sección.

Por lo que se crea la clase `cpicMensajes`, encargada de mostrar dichos mensajes dinámicos y se representa a continuación:



Figura 5.4 Clase cpicMensajes

Área Submenú.

El área submenú es muy importante en el funcionamiento del sistema, ya que es la encargada de mostrar las diferentes funciones que el área dinámica puede mostrar de acuerdo a la sección en que el usuario se encuentre.

A continuación se muestran las funciones que se tiene en el área submenú:

- ❖ Copiar archivos.
- ❖ Visualizar una práctica seleccionada.
- ❖ Mostrar el material de una práctica seleccionada.
- ❖ Escuchar el resultado correspondiente a una práctica que se encuentra en el área dinámica.
- ❖ Ejecutar un software estando activa una práctica.
- ❖ Mostrar el contenido de un manual seleccionado, ya sea de software o del MC68HC11.
- ❖ Ejecutar el software seleccionado por el usuario.

Es por eso que se crea clase correspondiente a ésta sección es csubmenu.

Y se representa de la siguiente manera:

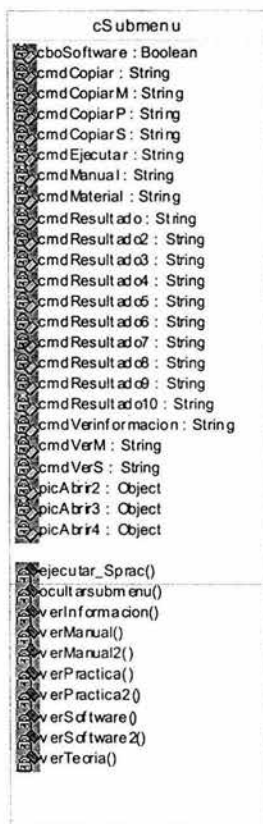


Figura 5.5 Clase cSubmenu

Es importante señalar que las clases creadas y mencionadas anteriormente se relaciona entre sí (tipo de relación agregación) por lo que a continuación se muestra el diagrama de clases, en el cual se puede observar dichas relaciones:

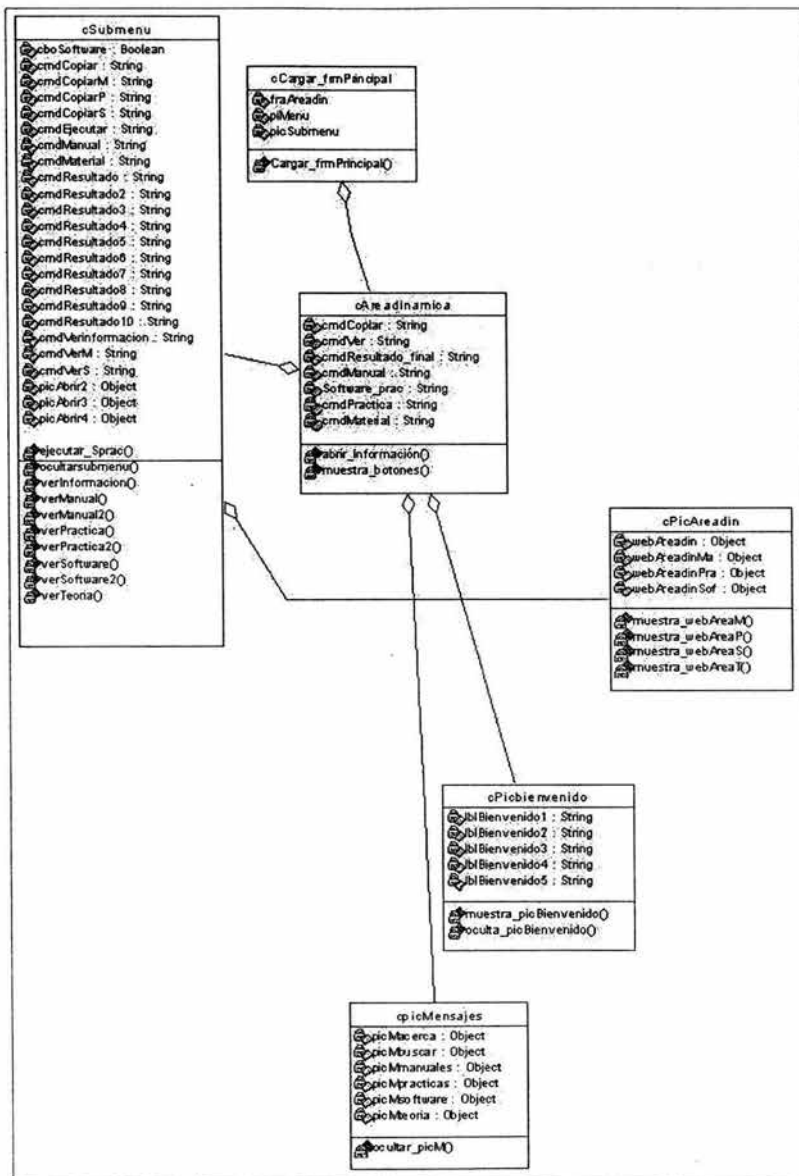


Figura 5.6 Diagrama de clases

La siguiente figura representa la ventana final de la presentación del sistema.

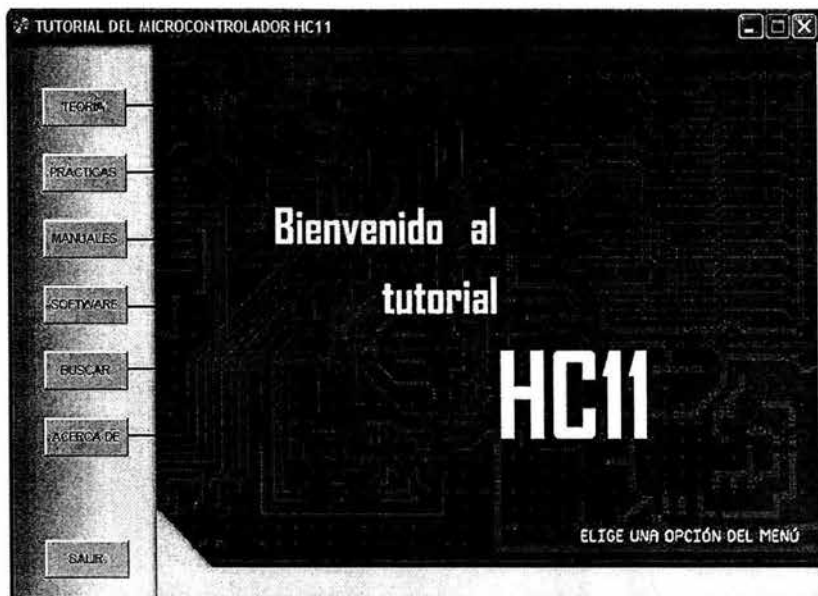


Figura 5.7 Presentación del sistema

5.1.1 Sección TEORÍA

Esta sección es la encargada de mostrar la teoría básica del microcontrolador MC68HC11.

Se activa al dar clic en el botón **TEORÍA** que se encuentra en el Menú principal de la figura anterior, desplegándose así en el área dinámica el archivo en formato HTML que permitirá su consulta por medio de un índice de temas relacionados con la teoría básica.

La ventana correspondiente se muestra a continuación:

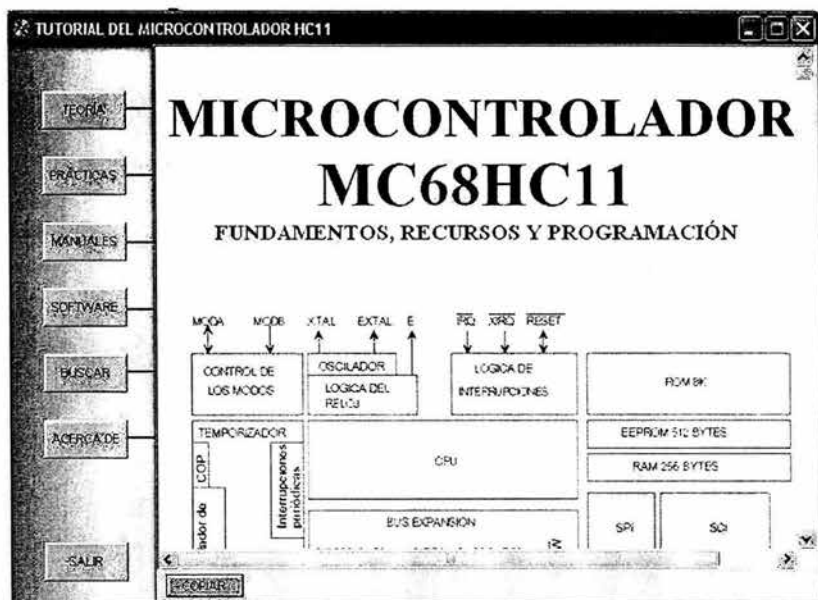


Figura 5.8 Ventana información TEORÍA

En el momento en que se visualiza la información requerida, el submenú es actualizado de acuerdo a las funciones que se pueden realizar en ésta sección, es decir se cuenta con la opción para copiar el archivo completo de la teoría básica, el cual se encuentra comprimido.

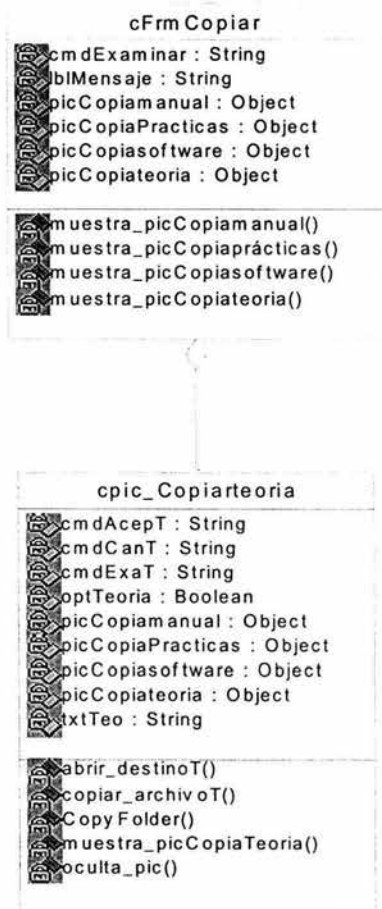


Figura 5.9 Clases `cpic_Copiateoria` y `cFrmCopiar`

Como se muestra en la figura 5.9, el diagrama de clases para la sección de copiar teoría se desarrollaron varios métodos que sirvieron para hacer la acción de copiar dicho archivo como: `abrir_destinoT`, `copiar_archivoT`, etc.

Por lo que al accionar el botón copiar de ésta sección el sistema mostrará una ventana llamada Copiar, para que el usuario sólo elija la ruta destino en la que se copiará dicho archivo, posteriormente se da click en el botón copiar y el sistema mostrará un mensaje con el que el usuario podrá saber que el archivo fue copiado exitosamente en la ruta destino que anteriormente se especificó. Ver las figuras 5.10 y 5.11.

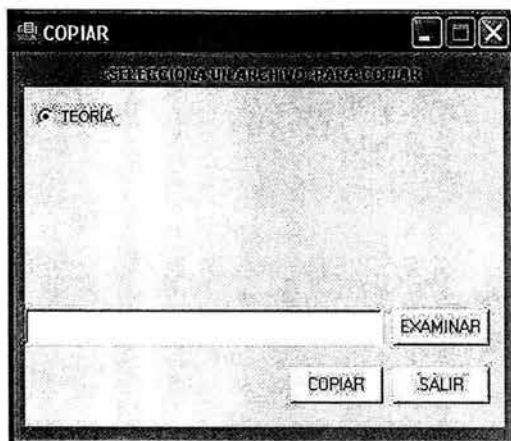


Figura 5.10 Copiar Teoría



Figura 5.11 Copia exitosa archivo TEORÍA

5.1.2 Sección PRÁCTICAS

En la sección prácticas se crearon las siguientes clases `cPicPrac`, `cMaterial`, la primera clase contienen los siguientes métodos con los cuales se visualizará en el área dinámica el contenido de las prácticas y el resultado final de cada una de las prácticas.

Las relaciones que tiene esta clase se muestran en el diagrama siguiente que se representa en la figura 5.12.

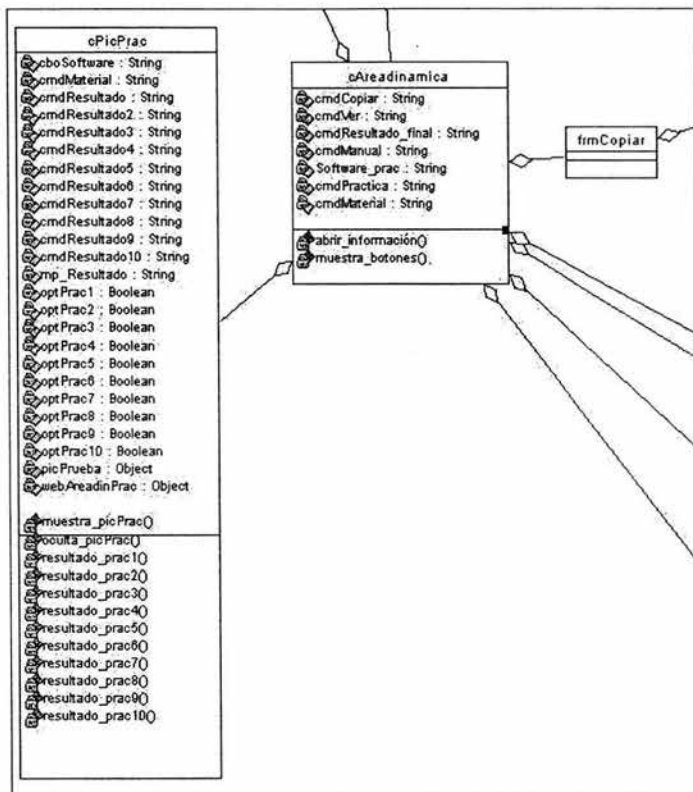


Figura 5.12 Diagrama de clases Prácticas

Para la sección Prácticas se realizaron dos clases, una para mostrar al usuario la información de cada una de las prácticas (cPicPrac) y otra para copiar el archivo (cpicCopiapracticass), además como se muestra en la figura anterior; estas clases están relacionadas con la clase cAreadinamica.

La figura 5.13 muestra la clase cpicCopiapracticass y sus relaciones con otras clases como cFrmCopiar.

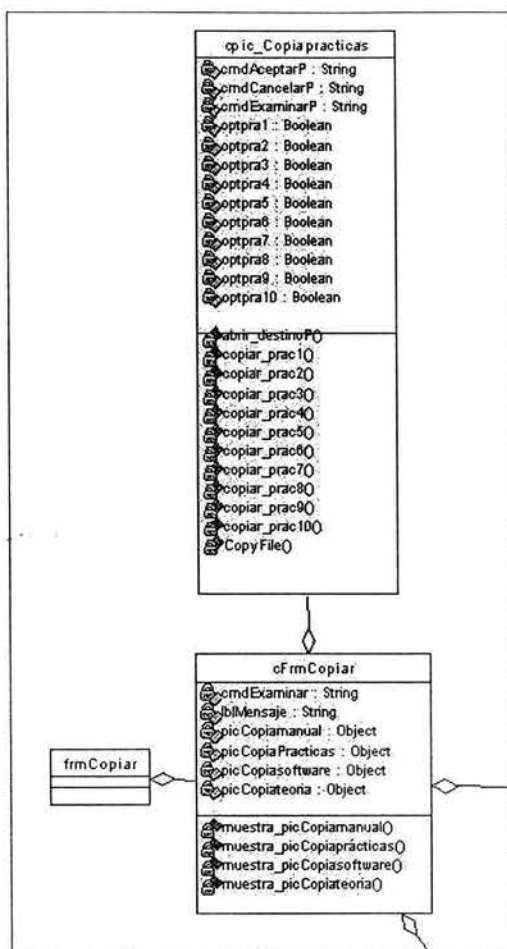


Figura 5.13 Diagrama de clases cpicCopia practicas

El usuario podrá tener acceso a la sección Prácticas por medio del Menú principal, el cual muestra el botón correspondiente a ésta sección.

Al accionar el botón Prácticas, el sistema mostrará el área dinámica las diferentes prácticas de las que está compuesta, es decir mostrará botones de selección con los cuales el usuario podrá elegir sólo una práctica para visualizar su contenido.

Para indicarle al sistema que se desea ver el contenido de dicha práctica seleccionada, sólo será necesario observar el submenú que se encuentra en la parte inferior de la ventana y elegir la opción ver.

Esta nueva ventana se muestra en la figura 5.14

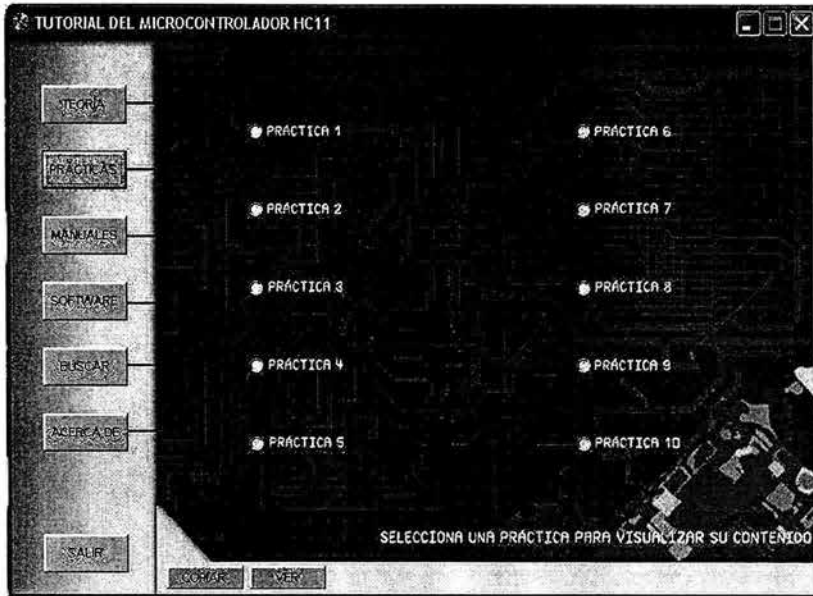


Figura 5.14 Ventana PRÁCTICAS

El submenú de ésta sección ha cambiado en el momento que el sistema despliega la información de la práctica.

Ahora se cuentan con las opciones que se pueden ejecutar dentro de ésta sección, las cuales se mencionan a continuación:

- ❖ Copiar los archivos de las prácticas (COPIAR).
- ❖ Ver el material correspondiente a la práctica activa (MATERIAL)
- ❖ Escuchar el resultado de la práctica activa (RESULTADO)
- ❖ Ejecutar los diferentes programas con los que cuenta el tutorial (SOFTWARE)

Por lo que la ventana que cuenta con estas características se muestra en la figura siguiente:

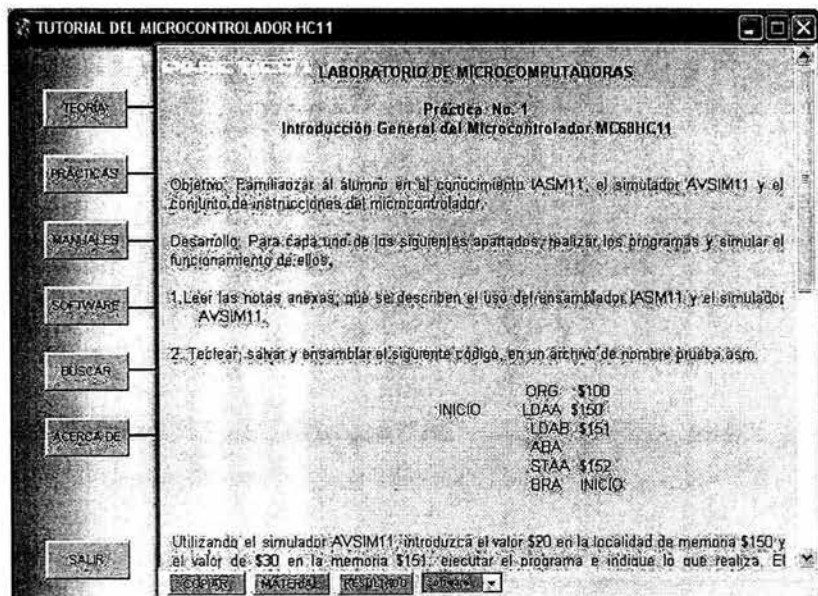


Figura 5.15 Información desplegada de una práctica seleccionada

Descripción de cada una de las secciones con las que cuenta el submenú para esta sección.

COPIAR

El usuario podrá tener acceso a esta opción al darle click en el botón COPIAR que se encuentra en el submenú, el sistema desplegará la ventana correspondiente para que el usuario elija la práctica que desea copiar.

La representación de dicha ventana se muestra en la figura 5.16. Es importante señalar que dicha ventana se actualiza de acuerdo a la sección en que el usuario se encuentre, por lo que su contenido cambiará.



Figura 5.16 Ventana Copiar Prácticas

A continuación el usuario elige una opción de la ventana y posteriormente al darle clic en el botón EXAMINAR se podrá elegir la ruta destino para guardar el archivo de la práctica. Para finalizar la copia el usuario dará click en el botón COPIAR y el sistema mostrará un mensaje indicándole que su archivo fue copiado exitosamente.

Si el usuario lo desea podrá cerrar la venta o cancelar la copia con el botón SALIR.

MATERIAL

La clase creada para mostrar al usuario las imágenes que contiene ésta sección es denominada cMaterial.

La sección Material es importante debido a que en ella se muestra por medio se imágenes, el material correspondiente a la práctica que el usuario está observando. Sólo será necesario que el usuario elija la opción MATERIAL del submenú. Y se activará una sub-ventana dentro del área dinámica, la cual posee un botón independiente para cerrarla.

El usuario podrá identificar el nombre de los componentes del material al posicionarse sobre la imagen correspondiente.

Esto se muestra en la siguiente figura:

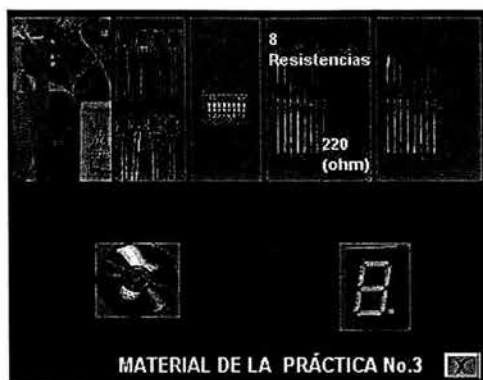


Figura 5.17 Ventana MATERIAL

RESULTADO

En cuanto al resultado de la práctica que se encuentra activa, el usuario podrá elegir la opción RESULTADO que se encuentra localizada en el submenú.

Al activarse dicha opción el usuario podrá escuchar el resultado final de la práctica en la que se encuentre. En el momento en que se activa el sonido, en la parte derecha del submenú se observará tres opciones para manipular el sonido, es decir, existe una opción para activar nuevamente el sonido, otra para pausar y la última para detenerlo.

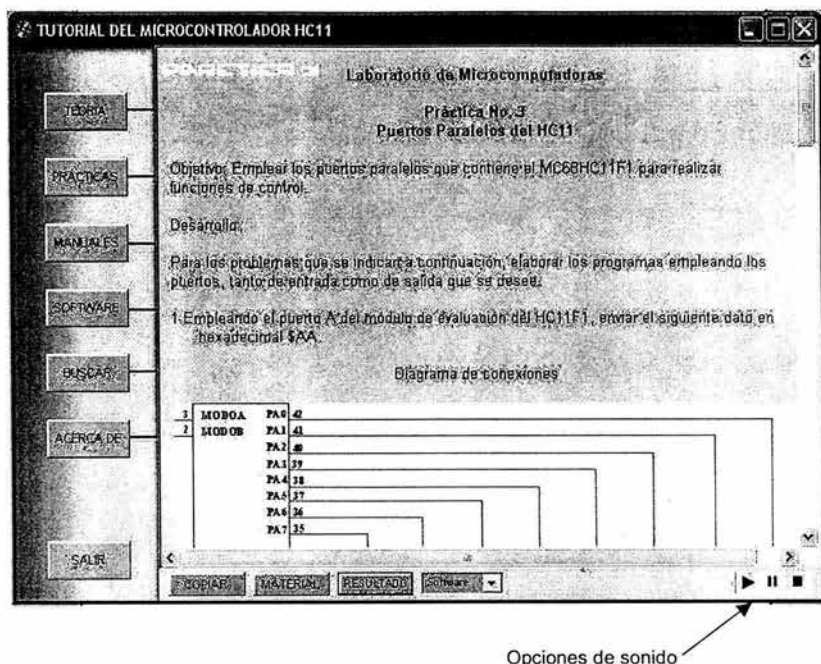


Figura 5.18 Ventana RESULTADO

SOFTWARE

En cuanto a esta sección del submenú, se pueden ejecutar desde la sección PRÁCTICAS los diferentes programas que posee el tutorial.

Se desplegará una ventana independiente relacionada con la opción de software que el usuario haya elegido de la lista que se despliega.

5.1.3 Sección MANUALES

Para la implementación de ésta sección se crearon las clases cPicMan, cpic_Copiamaual.

La primera tiene métodos que permiten desarrollar las funciones que despliega la información en el área dinámica, correspondiente a un manual seleccionado. Y la segunda permite establecer funciones que realizan la copia de cada uno de los archivos correspondientes a los manuales que posee el tutorial.

Estas clases se muestran en las figuras 5.19 y 5.20 respectivamente.

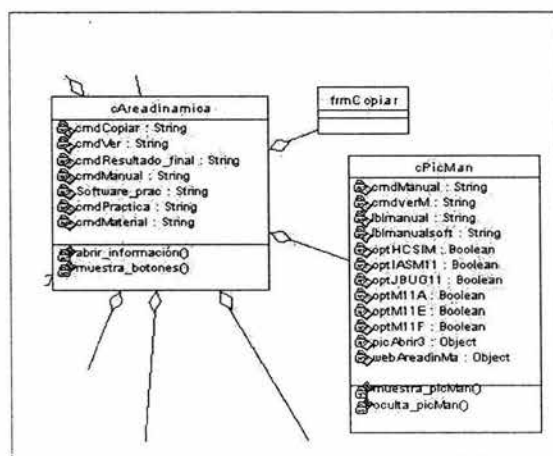


Figura 5.19 Clase cPicMan y sus relaciones con otras clases

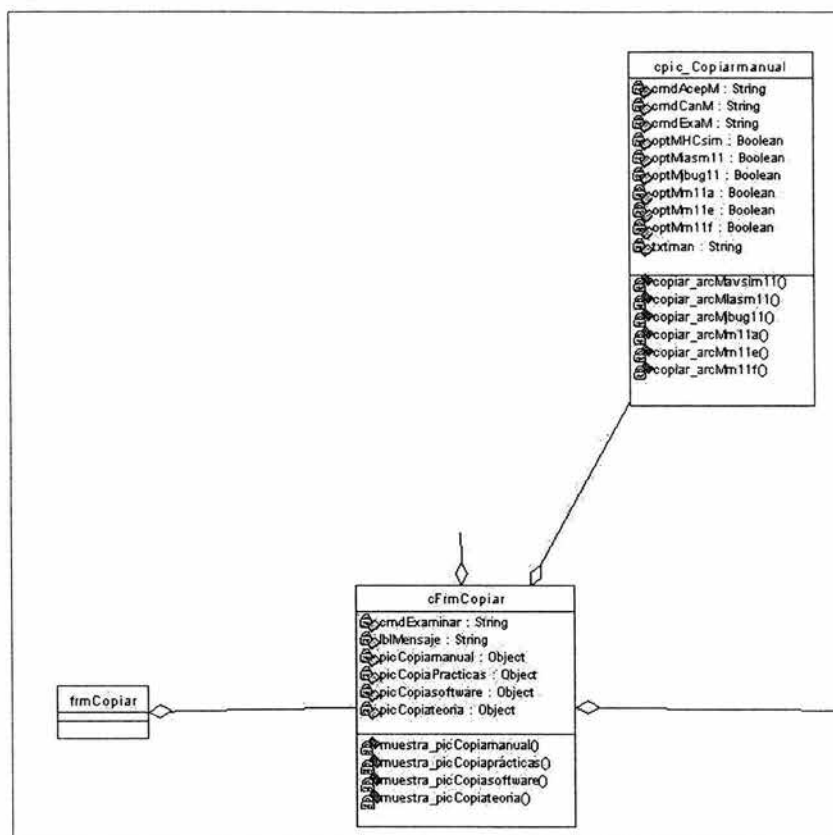


Figura 5.20 Clase `cpic_Copiarmanual` y sus relaciones con otras clases

Principalmente en ésta sección el usuario podrá tener acceso a los manuales del MC68HC11 y a los manuales de software que el tutorial posee; estos son los siguientes:

- ❖ Manual del microcontrolador MC68HC11 serie A
- ❖ Manual del microcontrolador MC68HC11 serie E
- ❖ Manual del microcontrolador MC68HC11 serie F
- ❖ Manual del IASM11
- ❖ Manual del HCSIM
- ❖ Manual del JBUG11

El usuario tendrá la opción de elegir uno de los manuales que se despliegan al hacer click en el botón MANUALES del Menú principal, por lo que se presentará una ventana como se muestra en la figura 5.21.

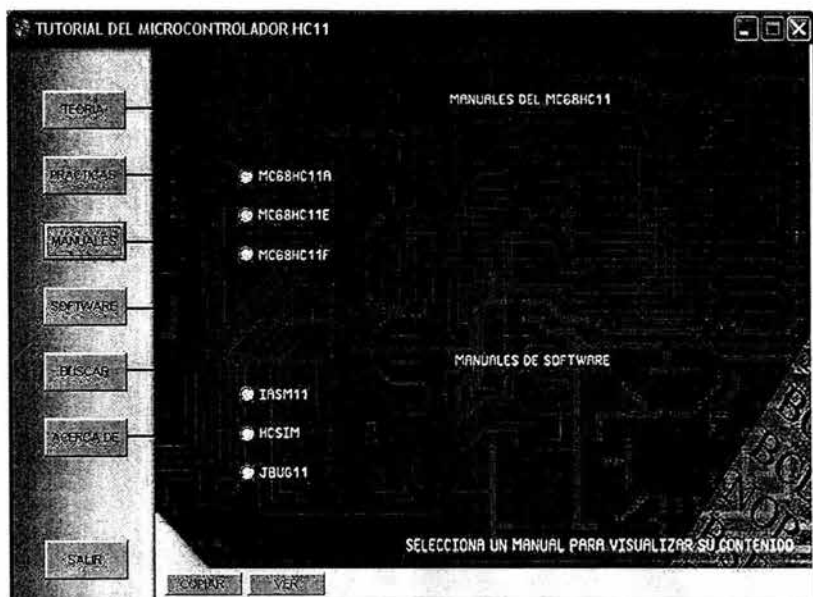


Figura 5.21 Ventana MANUALES

Ya que el usuario elige una opción de manual, encontrará en el submenú de ésta sección la opción VER, con la cual se desplegará en el área dinámica la información completa del manual y será posible consultarla.

La figura 5.22 muestra la representación de la ventana que el usuario podrá observar.

Existe otra opción dentro de la sección MANUALES; ésta opción es COPIAR, con la cual el usuario elegirá dentro de la ventana que se despliega al hacer click en dicha opción, el archivo que desea copiar, posteriormente se especifica la ruta destino del archivo y a continuación se realiza la copia, sólo con hacer click en el botón COPIAR de ésta nueva ventana. Si el usuario desea cancelar la copia del archivo, sólo deberá cerrar la ventana.

La figura 5.23 muestra la ventana COPIAR correspondiente a la sección MANUALES.

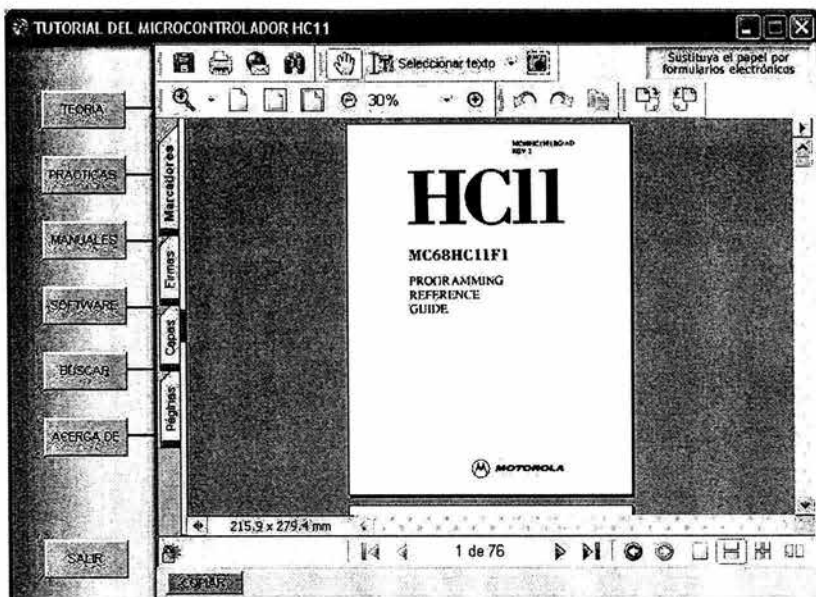


Figura 5.22 Ventana información del manual

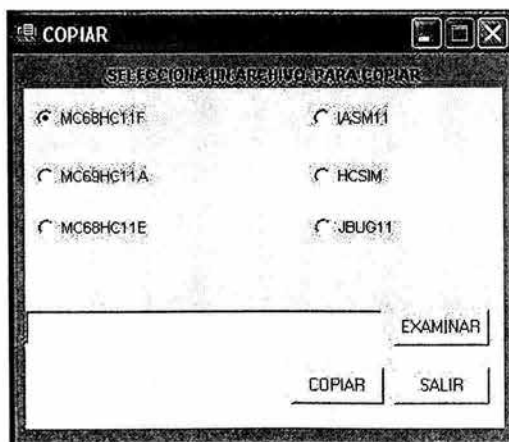


Figura 5.23 Ventana copiar manual.

5.1.4 Sección SOFTWARE

Para la implementación de ésta sección se crearon las clases cPicSoft, cpic_Copiasoftware. La primera tiene métodos que permiten desarrollar las funciones que ejecutan cada uno de los programas que el tutorial posee.

Con la creación de la segunda clase permite implementar las funciones que realizan la copia de los archivos correspondientes a los programas.

La representación de cada clase se muestra a continuación en las figuras correspondientes:

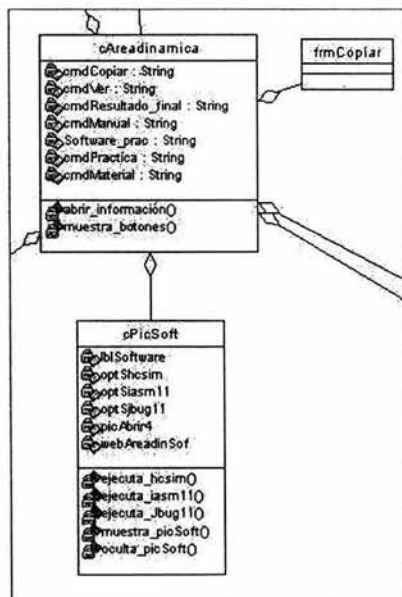


Figura 5.24 Clase cPicSoft

opción IASM11, se tiene la figura 5.28 y en la figura 5.29 se representa la ventana correspondiente al JBUG11.

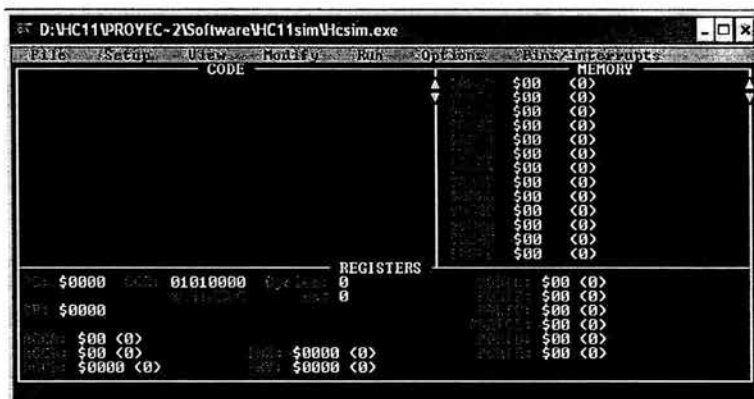


Figura 5.27 HCSIM

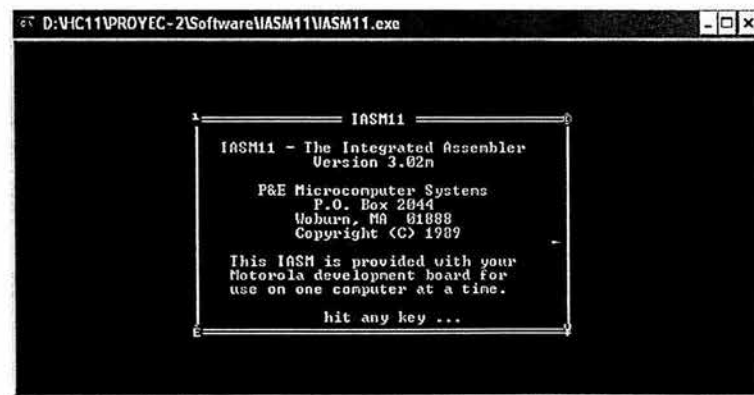


Figura 5.28 IASM11

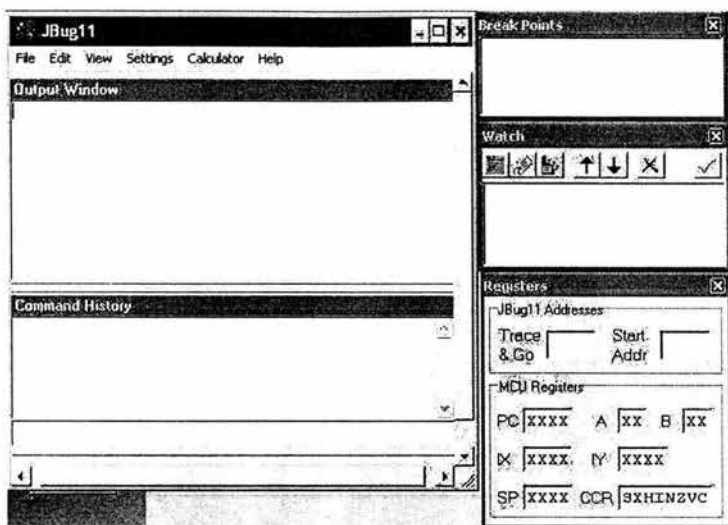


Figura 5.29 JBUG11

En cuanto a la opción COPIAR en ésta sección, la ventana es similar a otras secciones en cuanto a las acciones que se deben realizar para copiar un archivo, sólo cambiará en los archivos disponibles para ser copiados.

La ventana correspondiente a la opción COPIAR de ésta sección está representada en la figura siguiente:

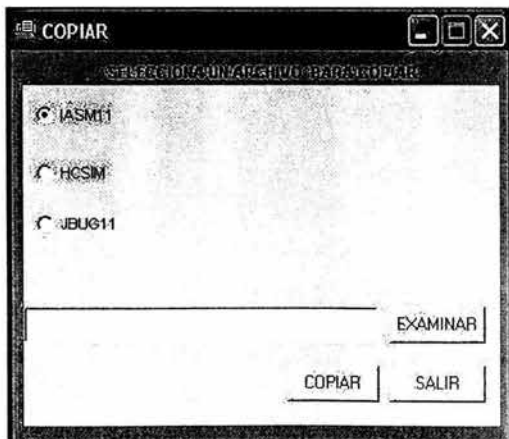


Figura 5.30 Copiar Software

5.1.5 Sección BUSCAR

Una sección de gran ayuda para el usuario es la de BUSCAR, en la que podrá consultar el documento de la teoría básica de manera más específica, es decir, el usuario podrá ingresar una palabra clave relacionada con un tema que desee consultar.

Para la implementación de ésta sección se crea la clase `Busqueda_hc11`, con la que se generarán funciones que permitan ejecutar los archivos correspondientes a la búsqueda de palabras.

Además dentro de ésta sección se encuentra el manual de usuario.

La ventana que se despliega al hacer click en el botón BUSCAR es la que se muestra en la figura 5.31

La búsqueda de palabras se presenta de dos maneras, ya sea por medio de un índice como se muestra en la figura 5.32 o por medio de otra sección que se muestra en la figura 5.33.

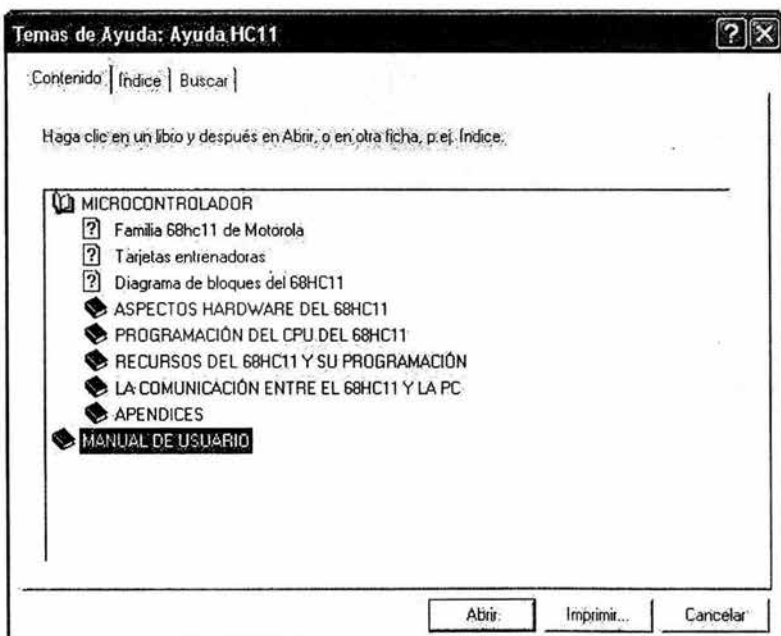


Figura 5.31 Sección BUSCAR

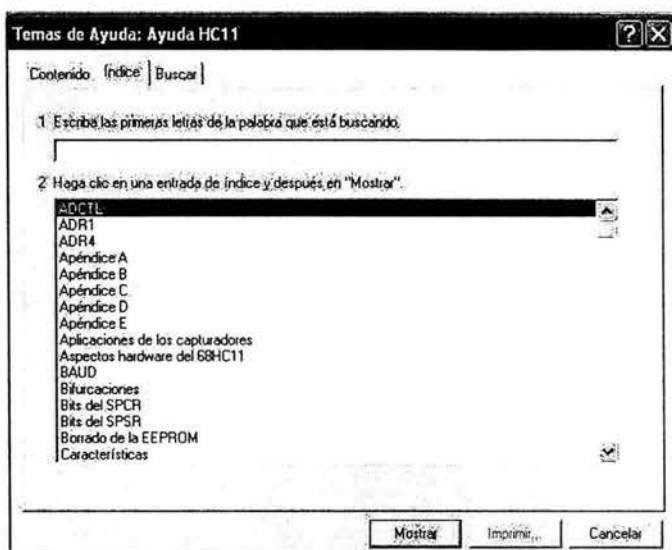


Figura 5.32 Índice

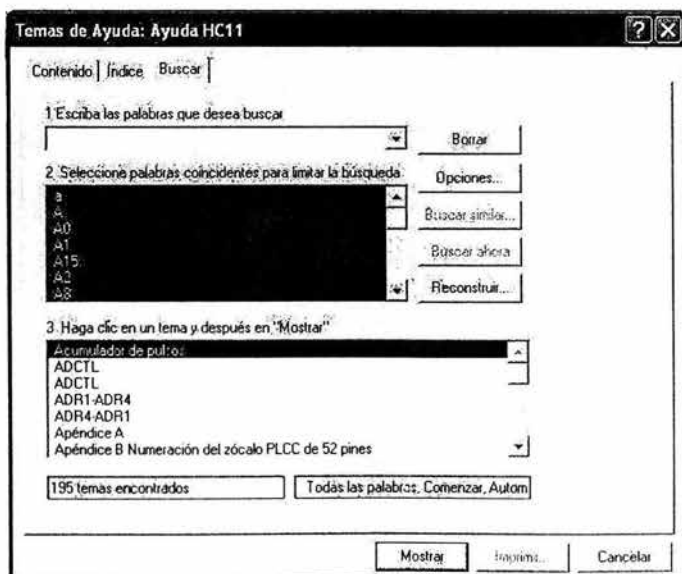


Figura 5.33 Buscar

5.1.6 Sección COPIAR

Como se ha observado en las secciones principales de éste capítulo, cada una cuenta con una opción en el submenú, es decir la opción que permite al usuario copiar archivos. Para activar dicha opción es necesario que el usuario se encuentre dentro de una sección. Al accionar dicha opción el sistema mostrará una nueva ventana (figura 5.35), la cual está compuesta por botones de opción que permitirán al usuario elegir solamente un archivo. A continuación será necesario indicar la ruta destino en la que se desea copiar dicho archivo. Esto se hará por medio del botón EXAMINAR que se encuentra dentro de la misma ventana, con éste botón se activará una nueva ventana (figura 5.36) en la que se podrá elegir la ruta, para después realizar la copia del archivo por medio del botón COPIAR. En el instante en que el sistema realiza la copia se presentará un mensaje que informa al usuario que se copio con éxito. Ver figura 5.37.

Si el usuario decide cancelar la copia del archivo seleccionado o cerrar la ventana COPIAR sólo será necesario activar el botón SALIR.

Para la realización de las funciones que permitirán en la implementación copiar archivos se crean las clases:

- ❖ cFrmCopiar, cuyos métodos permiten mostrar la ventana Copiar de acuerdo a la sección en que se encuentra el usuario.
- ❖ cpic_Copiateoria permite copiar el archivo de la sección TEORÍA.
- ❖ cpic_Copiapracticas contiene métodos para que sea posible la copia de cada uno de los archivos de las prácticas.
- ❖ cpic_Copiamanual permite establecer métodos que realizan la copia de los archivos correspondientes a los manuales.
- ❖ cpic_Copiasoftware, cuyos métodos realizan la copia de cada programa que tiene el tutorial.

La relación que existe entre éstas clases se puede observar en la figura siguiente:

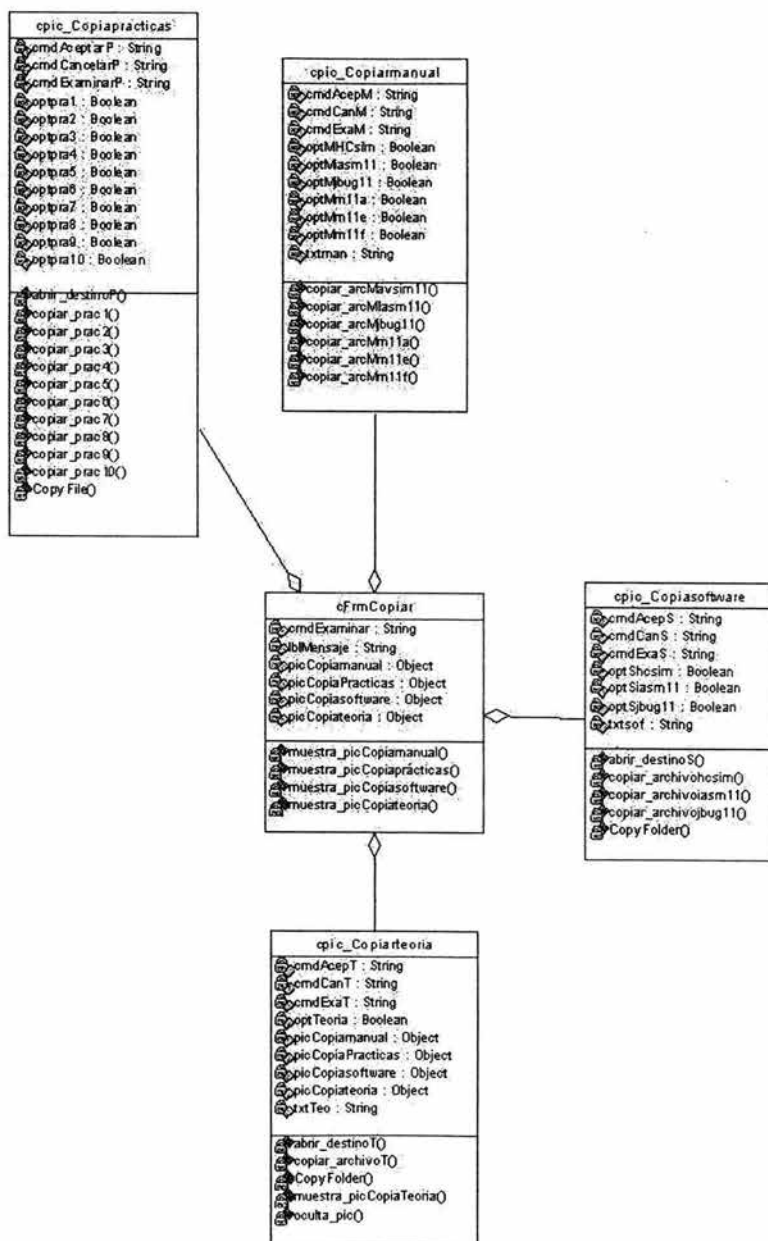


Figura 5.34 Clases Sección COPIAR

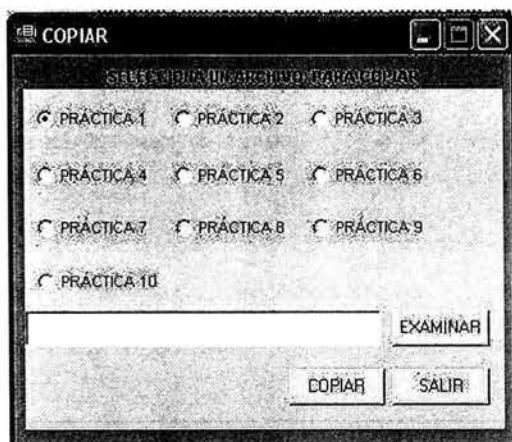


Figura 5.35 Ventana COPIAR



Figura 5.36 Ventana para elegir una ruta válida

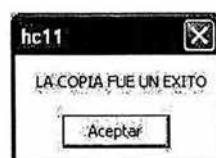


Figura 5.37 Mensaje Copia exitosa

5.1.7 Sección Acerca de

En la última sección del tutorial el usuario conocerá la información relacionada con la producción del sistema, es decir, la dependencia a la que pertenece el proyecto, la versión del sistema y las personas que colaboraron en su realización.

Y se puede visualizar ésta sección en la figura que se muestra a continuación:

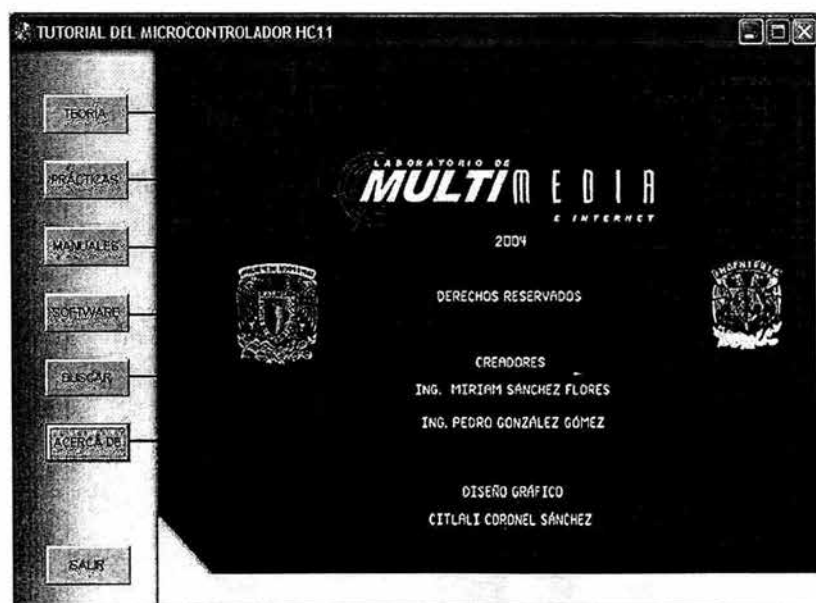


Figura 5.38 Venta ACERCA DE

5.2 Pruebas

Al finalizar la fase de implementación de cualquier proyecto de software, es necesario contar con una etapa de evaluación o pruebas para comprobar que se haya construido de acuerdo a los requerimientos establecidos.

5.2.1 Estrategia de Pruebas

Para desarrollar las pruebas se seguirán los siguientes pasos:

1. Se comprobará que el sistema se pueda instalar sin ninguna dificultad, es decir, comprobar que en las versiones de Windows XP y 2000 y en diversas máquinas con los requerimientos especificados, este se pueda instalar correctamente y en un tiempo no muy grande (máximo 10 minutos). Posteriormente el usuario podrá ejecutar el sistema desde la PC con el disco de instalación y ejecución dentro de la unidad de CD.
2. Para la parte de instalación y ejecución se va a requerir de una computadora, un disco de instalación y ejecución (que es el mismo en este caso) y un operador.
3. Los resultados que se desean obtener son primero que la instalación se ejecute correctamente y posteriormente que los archivos se carguen sin ningún problema, ya que estos son cargados a través del disco de instalación. Además que la ventana de copiar no tenga ningún problema en el momento de mandar los datos a una carpeta válida que seleccione el usuario en la computadora, que las imágenes y el sonido se puedan distinguir de una forma correcta, que se detecte la unidad en la que se encuentra en el CD del tutorial y finalmente que el botón Buscar del menú principal funcione adecuadamente.

5.2.2 Lista de pruebas

Los principales puntos que deberán cubrirse en ésta etapa son los siguientes:

1. Instalación adecuada del sistema.
2. Ejecución del sistema.
3. Cargar archivos.
4. Copiar archivos.
5. Abrir otras ventanas.
6. Ejecución del audio.
7. Cargar imágenes.
8. Cerrar ventanas.
9. Cerrar Sistema.
10. Cargar la sección Buscar.
11. Detectar la unidad en la que se encuentra el CD del tutorial.
12. Detectar rutas válidas para copiar archivos.

Ésta etapa se encuentra conformada por la etapa de pruebas de integración y pruebas del sistema. Las pruebas de integración aseguran que todas las partes se hayan incluido en el sistema y que las interfaces trabajen en conjunto. Y las

pruebas del sistema, validan las funciones y desempeño contra los requerimientos.

Además se realizaron pruebas por medio de encuestas a la comunidad de la Facultad de Ingeniería, en las que se aplicó un cuestionario que evaluó los puntos más importantes del tutorial.

Equipos con los que se contó para la realización de las pruebas y sus principales características:

EQUIPO	CARATERISTICAS
Computadora 1	Sistema Operativo: Windows 2000 Procesador: Pentium IV 1.4 GHz. Resolución de pantalla: 1024 X 720 pixeles y 800 X 600 pixeles Tarjeta de sonido integrada Tarjeta de video ATI
Computadora 2	Sistema Operativo: Windows XP Profesional Procesador: AMD Athlon(TM) XP 2100+ 1.73 GHz. Resolución de pantalla: 1024 X 768 pixeles. Tarjeta de sonido y video integrada
Computadora 3	Sistema Operativo: Windows XP Profesional Procesador: AMD Athlon(TM) XP 2100+ 1.73 GHz. Resolución de pantalla: 800 X 600 pixeles. Tarjeta de sonido y video integrada
Computadora 4	Sistema Operativo: Windows XP Profesional Procesador: AMD Athlon(TM) XP 2400+ 2.4GHz Resolución de pantalla: 1024 X 768 pixeles Tarjeta de sonido y video integrada

EQUIPO	CARATERISTICAS
Computadora 5	Sistema Operativo: Windows XP Profesional Procesador: AMD Athlon(TM) XP 2100+ 1.73 GHz. Resolución de pantalla: 1024 X 768 pixeles. Tarjeta de sonido y video integrada

El tiempo destinado a la realización de las pruebas fue de 50 minutos aproximadamente por cada equipo.

5.2.3 Forma Registro de Pruebas (REGPR)

La Forma REGPR recaba los datos que se obtuvieron en la pruebas que se realizó a cada equipo.

La cual se encuentra formada por:

1. Nombre del equipo
2. Fecha de ejecución de la prueba
3. Nombre de quien aplica la prueba
4. Nombre o número de pruebas
5. Resultados de las pruebas
6. Observaciones
7. Hora de inicio
8. Hora de terminación

A continuación se presentan las formas REGPR correspondientes a cada uno de los equipos:

Computadora 1								
Fecha de ejecución de la prueba.	12 de noviembre de 2003							
Nombre de quien aplica la prueba.	Miriam Sánchez Flores y Pedro González Gómez							
Nombre o número de pruebas.	Instalación	Ejecución	Copiar archivos	Sonido e Imágenes	Navegación	Cargar la sección Buscar	Detectar unidad	Detectar rutas válidas
Resultados de las pruebas.	Exitosa	Exitosa	Exitosa	Exitosa	Exitosa	Exitosa	Exitosa	Exitosa
Observaciones	La ejecución del software del HCSIM no fue adecuada.							
Hora de inicio	17:03 hrs.							
Hora de terminación	18:03 hrs.							

Computadora 2								
Fecha de ejecución de la prueba.	12 de noviembre de 2003							
Nombre de quien aplica la prueba.	Miriam Sánchez Flores y Pedro González Gómez							
Nombre o número de pruebas.	Instalación	Ejecución	Copiar archivos	Sonido e Imágenes	Navegación	Cargar la sección Buscar	Detectar unidad	Detectar rutas válidas
Resultados de las pruebas.	Exitosa	Exitosa	Exitosa	Exitosa	Exitosa	Exitosa	Exitosa	Exitosa
Observaciones	En el momento de la instalación se despliega un mensaje que indica que ya existe una versión de Windows Media Player y pregunta si se desea conservar la existente o reemplazarla.							
Hora de inicio	19:28 hrs.							
Hora de terminación	20:31 hrs.							

Computadora 3								
Fecha de ejecución de la prueba.	13 de noviembre de 2003							
Nombre de quien aplica la prueba.	Miriam Sánchez Flores y Pedro González Gómez							
Nombre o número de pruebas.	Instalación	Ejecución	Copiar archivos	Sonido e Imágenes	Navegación	Cargar la sección Buscar	Detectar unidad	Detectar rutas válidas
Resultados de las pruebas.	Exitosa	Exitosa	Exitosa	Exitosa	Exitosa	Exitosa	Exitosa	Exitosa
Observaciones	En el momento de la instalación se despliega un mensaje que indica que ya existe una versión de Windows Media Player y pregunta si se desea conservar la existente o reemplazarla.							
Hora de inicio	19:28 hrs.							
Hora de terminación	20:31 hrs.							

Computadora 4								
Fecha de ejecución de la prueba.	14 de noviembre de 2003							
Nombre de quien aplica la prueba.	Miriam Sánchez Flores y Pedro González Gómez							
Nombre o número de pruebas.	Instalación	Ejecución	Copiar archivos	Sonido e Imágenes	Navegación	Cargar la sección Buscar	Detectar unidad	Detectar rutas válidas
Resultados de las pruebas.	Exitosa	Exitosa	Exitosa	Exitosa	Exitosa	Exitosa	Exitosa	Exitosa
Observaciones	Este equipo de prueba no cuenta con el software Visual Basic versión 6.0 En el momento de la instalación se despliega un mensaje que indica que ya existe una versión de Windows Media Player y pregunta si se desea conservar la existente o reemplazarla.							
Hora de inicio	18:35 hrs.							
Hora de terminación	19:30 hrs.							

Computadora 5								
Fecha de ejecución de la prueba.	14 de noviembre de 2003							
Nombre de quien aplica la prueba.	Miriam Sánchez Flores y Pedro González Gómez							
Nombre o número de pruebas.	Instalación	Ejecución	Copiar archivos	Sonido e Imágenes	Navegación	Cargar la sección Buscar	Detectar unidad	Detectar rutas válidas
Resultados de las pruebas.	Exitosa	Exitosa	Exitosa	Exitosa	Exitosa	Exitosa	Exitosa	Exitosa
Observaciones	Este equipo de prueba no cuenta con el software Visual Basic versión 6.0 En el momento de la instalación se despliega un mensaje que indica que ya existe una versión de Windows Media Player y pregunta si se desea conservar la existente o reemplazarla.							
Hora de inicio	19:36 hrs.							
Hora de terminación	20:30 hrs.							

5.2.4 Resultados finales de las pruebas

1. Durante la instalación del sistema utilizando el CD no se encontró ningún problema, ésta actividad nos llevó aproximadamente 40 seg.
2. Durante la ejecución del sistema los resultados fueron satisfactorios debido a que todos los botones y ventanas realizaron la función para los que están diseñados.
3. Los archivos que componen el sistema se cargaron de forma satisfactoria a excepción de los archivos que están en formato PDF, debido a que se posee la versión 6 del Adobe Reader 6.0, por lo que se debe cargar primero el programa y mantenerlo de esa manera.
4. La acción de copiar archivos se realizó adecuadamente y cabe mencionar que los archivos que se copian, están comprimidos.
5. El siguiente punto a evaluar es que todas las ventanas del sistema se abran adecuadamente por lo que se satisfizo en el sistema, aún teniendo varias ventanas activas.

6. En cuanto a la ejecución del sonido se obtuvo lo siguiente: cuando se carga un archivo del resultado correspondiente a una práctica, éste se queda cargado al sacar el CD. Por lo que al cargar un nuevo archivo de sonido relacionado con una práctica distinta se activaba el archivo ya almacenado.
7. Al cargar las imágenes que se presentan en la sección de prácticas correspondientes al material, no se tuvo problema alguno.
8. El punto de evaluación para cerrar ventanas que se encuentran activas se cumplió satisfactoriamente.
9. Para salir del sistema se cuenta en el menú principal con un botón que realizó ésta acción satisfactoriamente.
10. La ejecución de la sección Buscar que se encuentra en el menú principal se cargó y se visualizó adecuadamente, desplegándose así una ventana adicional para la búsqueda de palabras en la teoría básica del Tutorial.
11. El sistema cumplió adecuadamente con la detección del CD del Tutorial en la unidad correspondiente a la computadora.
12. La detección de rutas válidas se realizó adecuadamente en la ventana de copiar, ya que se manda un mensaje de error al detectarse una ruta no válida en la caja de texto que es donde se especifica la ruta para copiar un archivo deseado.

5.3 Pruebas del sistema por medio de encuestas

Otro medio que se propone para la fase de pruebas es un cuestionario de 10 preguntas, el cual fue aplicado a algunos alumnos y profesores de la comunidad de la Facultad de Ingeniería. Ver sección 4.5.3.3.

Este cuestionario fue de gran utilidad para tomar en cuenta la visión que tiene la población a la cual va dirigido el Tutorial del Microcontrolador HC11 y por la cual va a ser utilizado.

Los resultados finales de la pruebas a los usuarios por medio de las encuestas se presentan a continuación:

Pregunta	Respuesta SI	Respuesta NO	No respondió
1	91.66%	8.33%	0.00%
2	66.66%	33.33%	0.00%
3	100.00%	0.00%	0.00%
4	25.00%	75.00%	0.00%
5	83.33%	16.66%	0.00%
6	100.00%	0.00%	0.00%
7	83.33%	16.66%	0.00%
8	100.00%	0.00%	0.00%
9	100.00%	0.00%	0.00%

Con base en las encuestas realizadas y en la evaluación de las mismas, las conclusiones a las que hemos llegado de nuestro sistema son las siguientes:

1. En la primera pregunta el resultado fue satisfactorio.
2. En esta pregunta a la mayoría de los encuestados les pareció que la distribución de la información es adecuada.
3. Los botones que contiene el tutorial simbolizan correctamente el contenido de cada sección del tutorial.
4. Durante la ejecución del software existieron algunos problemas :
 - ❖ En el momento de cargar los archivos correspondientes a los manuales en formato PDF el programa Acrobat Reader Versión 6.0 genera un mensaje el cual indica que solamente se podrá cargar una aplicación web a la vez.
 - ❖ Error de sólo lectura al copiar un archivo a disco, es importante mencionar que en este error solamente ocurre en un caso extremo, gracias a que fue detectado en esta fase de pruebas fue corregido.
5. Se considera que el tutorial es de gran utilidad para complementar la asignatura de Microcomputadoras.
6. El tutorial fue de gran interés para las personas encuestadas.
7. Les pareció que toda la información contenida dentro del tutorial se presenta de manera clara.
8. La sección dedicada a realizar la búsqueda tiene un funcionamiento adecuado.

9. Se propone la difusión del tutorial por medio de los profesores que imparten la asignatura de Microcomputadoras, instalándolo en el laboratorio de microcomputadoras o por medio de Internet como software libre.

10. Las principales sugerencias que aportaron las personas encuestadas fueron las siguientes:

- ❖ Manejar un diseño gráfico adecuado.
- ❖ Agregar el programa WINBUG que permite la comunicación entre el microcontrolador y la computadora.

5.4 Postmortem

Esta es la etapa final de la metodología TSPi, en la cual se hace una evaluación final del sistema y de cada uno de los integrantes de equipo.

En cuanto a la evaluación final del sistema que se desarrolló, se considera que fue un producto de gran calidad que se apegó a los requerimientos que se establecieron, con el único objetivo de crear un software didáctico que sirviera de apoyo a la comunidad estudiantil de la Facultad de Ingeniería.

Es importante también mencionar mi auto evaluación como parte de un equipo de trabajo dentro de este proyecto. El aprendizaje de una nueva metodología de desarrollo de software y herramientas necesarias para el desarrollo de éste, me han llevado a un crecimiento académico dentro de la Facultad de Ingeniería, pero sobre todo el aplicar dichos conocimientos en un elemento útil como lo es la creación de un producto de software.

CONCLUSIONES

La creación de un producto de software siempre debe basarse en un proceso para su desarrollo en el que las necesidades de los usuarios son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código probado, documentado y certificado para su uso operativo.

Todo esto se logra con el seguimiento de una metodología que en éste caso en particular es la metodología TSPI. Con el seguimiento de ésta metodología a través de sus diferentes etapas se logró trabajar con un plan establecido por medio de fechas y actividades a desarrollar. Cabe mencionar que no se establecieron roles de equipo debido a que fue un trabajo desarrollado por una sola persona, pero si se dio seguimiento a cada una de las etapas de la metodología.

Además ésta metodología se complementó con el Proceso Personal de Software, que también resultó una herramienta importante debido a que se llevaba un registro de actividades a desarrollar de manera individual. Con lo cual se observó a lo largo del desarrollo del sistema; el desempeño que se logró en cada etapa de la metodología.

Es importante mencionar como los sistemas multimedia han tenido aplicaciones en diversas áreas como son: la industria, los negocios, la administración y especialmente en la educación.

En cuanto a la educación los sistemas multimedia han tenido gran apoyo en los métodos de enseñanza de todos los niveles, ya que cumple con las funciones que un proceso pedagógico debe cumplir para tener éxito.

Los sistemas multimedia han abierto grandes posibilidades de enseñanza, ya sea desde un software disponible en un disco compacto o bien por medio de Internet. Esto ha impulsado a los estudiantes a tener una opción más de aprender, es decir, buscar una manera más autodidacta de poseer conocimientos.

Ahora la Facultad de Ingeniería cuenta con un producto de software, específicamente con un sistema multimedia. Este sistema servirá de gran apoyo didáctico para la comunidad estudiantil de la misma facultad, el cual será distribuido de manera gratuita.

En cuanto al sistema se concluye que cumplió con los objetivos que se plantearon para su creación, es decir, se cuenta con un Tutorial para el Laboratorio de Microcomputadoras disponible en CD, que contiene la información básica para complementar la asignatura, contando con secciones relacionadas con la teoría básica del microcontrolador HC11, las prácticas del laboratorio, los manuales de software y del MC68HC11, así como con el software necesario para el desarrollo de las prácticas.

CONCLUSIONES

Su implementación se basó en el diseño establecido y que a su vez tiene como base el lenguaje unificado de modelado (UML).

UML sirvió como herramienta necesaria en el desarrollo de una programación orientada a objetos como así se requirió en el sistema.

La programación orientada a objetos posee grandes beneficios como la reusabilidad del código que permite que se puedan realizar modificaciones lo que permite obtener aplicaciones modificables, fácilmente extendibles a partir de componentes reusables. Esto también se ve reflejado en la disminución del tiempo de desarrollo.

En cuanto a las pruebas que se realizaron al sistema se puede concluir que operó de manera correcta en condiciones normales y en condiciones adversas, cumpliendo así con el plan de calidad establecido.

Las pruebas finales se realizaron a un sector de la facultad y a algunos profesores relacionados con la materia para obtener una evaluación más confiable, debido a que a ésta población será dirigido y se tendrían mejores resultados, y sobre todo que el objetivo del tutorial es el de satisfacer una necesidad académica.

Las personas encuestadas durante la fase de pruebas mostraron gran aceptación hacia el sistema además que aportaron importantes sugerencias, las cuales fueron tomadas en cuenta para tener un producto final más completo.

ANEXO A

TEORÍA BÁSICA DEL TUTORIAL

1. INTRODUCCIÓN

¿Qué es un microcontrolador?
El microcontrolador 68HC11 de Motorola.
Tarjetas entrenadoras.
Diagrama de bloques del 68HC11.

2. ASPECTOS HARDWARE DEL 68HC11

El patillaje del 68HC11.
Pines de reloj.
Pines de alimentación.
Pines de reset.
Pines de transmisiones serie asíncronas.
Pines de los capturadores.
Pines de interrupción externa.
Pines de configuración del modo de arranque.
Pines de los comparadores.
Pines de transmisiones serie síncronas.
Pines de los puertos de entrada/salida.
Pines de los buses.
Pines de los convertidores analógico/digitales.

3. PROGRAMACIÓN DEL CPU DEL 68HC11

Los modos de funcionamiento del 68HC11.
Registros del CPU.
Modos de direccionamiento.
Inmediato.
Extendido.
Directo.
Indexado.
Relativo.
Inherente.
Juego de instrucciones.
Instrucciones de carga, almacenamiento y transferencia.
Instrucciones aritméticas.
Operaciones aritméticas y manipulación de bits.
Desplazamientos y rotaciones.
Bifurcaciones y saltos.
Instrucciones de modificación de los bits del registro CCR.
Otras instrucciones.

INTERRUPCIONES

Interrupción de reset.

Tipos de interrupciones.

Prioridad de las interrupciones.

Proceso de interrupción.

Vectores de interrupción.

4. RECURSOS DEL 68HC11 Y SU PROGRAMACIÓN

El mapa de memoria.

PUERTOS DE ENTRADA/SALIDA.

El puerto A.

Programas ejemplo de manejo del puerto A.

El puerto B.

Ejemplos para el puerto B.

El puerto C.

Ejemplos de utilización del puerto C.

El puerto D.

Ejemplos de utilización del puerto D.

El puerto E.

Ejemplos de utilización del puerto E.

TRANSMISIONES SERIE ASÍNCRONAS (SCI).

Introducción.

Unidad de transmisión y unidad de recepción.

Registros del SCI.

Ejemplos de programación del SCI.

Mecanismo de interrupciones del SCI.

TRANSMISIONES SERIE SINCRONAS (SPI).

Introducción.

Protocolo.

Dispositivos SPI.

Enlaces.

Registros del SPI.

Ejemplo de programación del SPI.

TEMPORIZADOR PRINCIPAL.

Introducción.

Los registros del temporizador.

Diagrama de bloques del temporizador.

Ejemplo de manejo del temporizador.

INTERRUPCIONES EN TIEMPO REAL.

Introducción.

Los registros de las interrupciones en tiempo real.

Ejemplos de manejo de las interrupciones en tiempo real.

COMPARADORES.

Introducción.

Los registros de los comparadores.

Los comparadores y el puerto A.

El comparador 1.

Ejemplos de manejo de los comparadores.

CAPTURADORES DE ENTRADA.

Introducción.

Registros de los capturadores.

Aplicaciones de los capturadores.

Ejemplos de utilización de los capturadores.

ACUMULADOR DE PULSOS.

Introducción.

Registros de control asociados al acumulador de pulsos.

Ejemplos de manejo del acumulador de pulsos.

LA INTERRUPCION EXTERNA IRQ.

Introducción.

Ejemplo de utilización de la interrupción IRQ.

CONVERTIDOR ANALOGICO/DIGITAL (A/D).

Introducción.

Registros del convertidor A/D.

Programa ejemplo de manejo del convertidor A/D.

LA MEMORIA EEPROM.

Introducción.

Registros de control de la EEPROM.

Programación de la EEPROM.

Ejemplos de programación.

El registro CONFIG.

5. LA COMUNICACION ENTRE EL 68HC11 Y LA PC

Introducción.

El programa de arranque BOOTSTRAP.

Los modos de funcionamiento del 68HC11.

El modo de funcionamiento BOOTSTRAP.

Descripción del programa de arranque BOOTSTRAP.

Listado del programa BOOTSTRAP.

Dialogando con el 68HC11.

El protocolo implementado en el BOOTSTRAP.

Estado del microcontrolador una vez ejecutado el programa BOOTSTRAP.

Velocidades de transmisión en la PC.

El registro HPRIO.

Carga de programas en la memoria externa.

El formato .S19 de Motorola.

APENDICE A: Patillaje del 68HC11.

APENDICE B: Numeración del zócalo PLCC de 52 pines.

APENDICE C: Resumen de los registros de control del 68HC11..

APENDICE D: Descripción de todos los registros de control del 68HC11.

APENDICE E: Lista de mnemónicos del 68HC11.

**ANEXO B
PRÁCTICAS DEL LABORATORIO DE MICROCOMPUTADORAS**

1. Práctica No. 1 Introducción General del Microcontrolador MC68HC11
2. Práctica No. 2 Tarjeta de Evaluación F1 y PCBUG11
3. Práctica No. 3 Puertos Paralelos del HC11
4. Práctica No. 4 Puertos Paralelos del HC11 (Control de motores de CD)
5. Práctica No. 5 Puertos Paralelos
6. Práctica No. 6 Convertidor Analógico – Digital HC11
7. Práctica No. 7 Puerto Serial SCI
8. Práctica No. 8 Timer
9. Práctica No. 9 Compilador de C
10. Práctica No. 10 Proyecto Final

ANEXO C

MANUALES TÉCNICOS MC68HC11

EL sistema presentará únicamente los manuales técnicos del MC68HC11 correspondientes a las serie A, E y F los cuales van estar contenidos en tres bloques para que el usuario dependiendo el tipo del microcontrolador los pueda consultar y cuyo contenido se menciona a continuación:

- Sección 1. Descripción General
- Sección 2. Descripción de Pines
- Sección 3. (CPU)
- Sección 4. Modos de Operación On-Chip Memory
- Sección 5. Resets e interrupciones
- Sección 6. Puerto Paralelo Input/Output (I/O)
- Sección 7. Comunicación Serial Interface (SCI)
- Sección 8. Serial Periférico (SPI).
- Sección 9. Sistema de Tiempo
- Sección 10. Convertidor (A/D)
- Sección 11. Características Eléctricas
- Sección 12. Datos de Máquina
- Sección 13. Información Ordenada

Apéndice A. Soporte

Apéndice B. Esquema EVBU

Lista de Secciones

- AN1060 — M68HC11 Bootstrap Mode
- EB184 — Enabling the Security Feature
MC68HC711E9 Devices with PCbug11
on the M68HC711E9PGMR
- EB188 — Enabling the Security Feature
on M68HC811E2 Devices with PCbug11
on the M68HC711E9PGMR
- EB296 — Programming MC68HC711E9 Devices
with PCbug11 and the M68HC11EVBU

ANEXO D ESTÁNDARES DE CODIFICACIÓN DEL TUTORIAL DEL HC11

Para los estándares de codificación nos apegaremos a la notación húngara es un sistema usado normalmente para crear los nombres de variables cuando se programa en Windows.

Consiste en prefijos en minúsculas que se añaden a los nombres de las variables, y que indican su tipo. El resto del nombre indica, lo más claramente posible, la función que realiza la variable. A continuación se muestra este tipo de codificación.

Prefijo	Significado
b	Booleano
ca	Carácter (un byte)
dw	Entero largo de 32 bits sin signo (DOBLE WORD)
f	Banderas empaquetados en un entero de 16 bits
h	Manipulador de 16 bits (HANDLE)
l	Entero largo de 32 bits
lp	Puntero a entero largo de 32 bits
lpfn	Puntero largo a una función que devuelve un entero
lpasz	Puntero largo a una cadena terminada con cero
n	Entero de 16 bits
p	Puntero a entero de 16 bits
pt	Coordenadas (x, y) empaquetadas en un entero de 32 bits
rgb	Valor de color RGB empaquetado en un entero de 32 bits
sz	Cadena terminada en cero
w	Entero corto de 16 bits sin signo (WORD)

- ❖ Todos los nombres de variables, funciones, métodos, atributos, clases, etc, serán en minúsculas sin llevar acentos y sin incluir ñ.
- ❖ Para los atributos, métodos y clases se tomará la primera letra, como prefijo, es decir para los atributos el prefijo será "a", para los métodos m y para las clases c. Este prefijo irá acompañado de un nombre distintivo. En caso de que sean compuestos se separarán por un guión bajo, por ejemplo: para un atributo pregunta que es secreta se tiene `pregunta_secreta`.
- ❖ Comentarios: se utilizarán minúsculas. Antecediendo una comilla simple (') a cada renglón, ya que este es un estándar de Visual Basic para hacer comentarios en el código de programación.
- ❖ Constantes: Las constantes siempre se escribirán con mayúsculas. Si ésta es compuesta se separará con un guión bajo.
- ❖ Para las funciones y estructuras de control: la apertura y cierre de las llaves de la función correspondiente se tendrán a 4 espacios del margen izquierdo.

ANEXO D ESTÁNDARES DE CODIFICACIÓN DEL TUTORIAL DEL HC11

- ❖ Nombres de archivos: Estarán relacionados de acuerdo a la información que contengan, y su correspondiente extensión, todo se escribirá en minúsculas. Ñ sin acentos

Para las funciones que se utilizarán en la programación se tendrá el siguiente estándar:

El prefijo que se utilizará será fun seguido de un guión bajo y posteriormente un nombre alusivo a la acción de la función.

Ejemplo: Función que realiza copiar
fun_copiar

BIBLIOGRAFÍA

Crear en multimedia

Rosello Tormo, Emilio

Universidad Politécnica de Valencia, 1996

Introduction to the Personal Software Process

Watts S. Humphrey

Adisson Wesley, 2001

Introduction to the Team Software Process

Watts S. Humphrey

Adisson Wesley, 2000

Multimedia

Bustos Martín, Ignacio de

Madrid: Anaya Multimedia, 1995

Multimedia aplicada / Centro de Computación Profesional de México

México: McGraw-Hill Interamericana, 2001

Multimedia fácil!

Haskin, David

México: Prentice Hall, 1995

Programación con Visual Basic 6.0

Charte Ojeda, Francisco

Anaya Multimedia

Utilización de UML en Ingeniería del Software con objetos y componentes

Stevens Perditá

Madrid: Addison Wesley, 2002

http://www.mundoprogramacion.com/vb_hlp.htm

<http://lcueyatl.uam.mx/~chinampa/conce.html>

<http://iteso.mx/~carlosc/pagina/documentos/multidef.html#concepto>

<http://usuarios.lycos.es/leodata/docencia/GDS/unidad%203/UNIDAD03.pdf>

<http://www.angelfire.com/scifi/jzavalar/apuntes/IngSoftware.html#IngSoft>

<http://www.monografias.com/trabajos/objetos/objetos.shtml>

<http://www.dcc.uchile.cl/~psalinas/uml/introduccion.html>