

Ambiente Multimedial para la Enseñanza del Curso de Ingreso

Trabajo de Grado

Realizado por:

Claudia Marcela Cappelletti

Luisa Cecilia Luccioni

Directores:

Cristina Madoz

Rodolfo Bertone

Marzo, 1997



BIBLIOTECA
FAC. DE INFORMÁTICA
U.N.L.P.

Este informe está destinado a los alumnos ingresantes a la carrera de Informática de la Facultad de Ciencias Exactas de la Universidad Nacional de La Plata.

“Enseñar es encender una llama, no llenar un tonel.”

Platón



BIBLIOTECA
FAC. DE INFORMÁTICA
U.N.L.P.

Prólogo

El Curso de Ingreso por ser una etapa de transición entre el nivel medio y el universitario, resulta en general una tarea difícil de enfrentar por el alumno. No obstante le permitirá ingresar en el mundo universitario donde adquirirá los conocimientos y el desarrollo intelectual necesarios para enfrentar el mundo laboral con mayor solidez.

Es un hecho conocido que las Metodologías de Estudio evolucionan constantemente gracias a los avances tecnológicos producidos en los últimos años. El Curso de Ingreso de Informática debe asumir un papel protagónico en este proceso. Tal circunstancia determinó la creación de esta aplicación multimedial como herramienta de trabajo.

Este tutorial le permitirá al alumno familiarizarse con el material de estudio desde el momento de su inscripción en la facultad. Brindándole entonces la posibilidad de tomar contacto con la computadora desde el inicio de la carrera, logrando así una mayor motivación tendiente a satisfacer sus expectativas.

Es nuestro objetivo que la aplicación multimedial contribuya positivamente en el aprendizaje de los contenidos del curso y esperamos que se refleje en un mejor desempeño del alumno durante las pruebas de evaluación.

INDICE

Capítulo I - Objetivo

1

Capítulo II - Introducción y Motivación

2

Capítulo III - Multimedia en Educación

1. Introducción 5

2. Aplicaciones Multimediales 7

Capítulo IV - Análisis del Curso de Ingreso a Informática: Análisis y Expresión de Problemas

1. Introducción 8

2. Objetivos de los Capítulos

2.1. Capítulo 1 9

2.2. Capítulo 2 10

2.3. Capítulo 3 11

2.4. Capítulo 4 11

2.5. Capítulo 5 12

2.6. Capítulo 6 12

Capítulo V - Elección de la Herramienta

13



Capítulo VI - Análisis de Toolbook

1. Introducción	14
2. Posibilidades que brinda Toolbook	14
3. Lenguaje de Programación (OpenScript)	15
4. Aplicación	17
4.1. Técnicas para optimizar las aplicaciones	18
5. Tipos de Información Multimedia	18
5.1. Sonido	18
5.2. Video	19
5.3. Animación	21
6. Objetos de Toolbook	21
6.1. Propiedades de los Objetos	22
6.2. Campos	22
6.3. Campos Registro	22
6.4. Botones	23
6.5. Area especial de texto (Hotword)	23
6.6. Gráficos	23
6.7. Referencia a un segmento de fuente multimedia (Clip)	24
6.8. Escenario (Stage)	24
7. Respuesta de un objeto ante un evento (Script)	25
7.1. Manejador (Handler)	26
7.2. Búsqueda y Corrección de errores	26
7.3. Tratamiento de Mensajes	27
8. Libros y Páginas	27
8.1. Partes de una página (Background y Foreground)	28
8.2. Niveles de Trabajo	28
8.2.1. Nivel Autor	28
8.2.2. Nivel Lector	29

Capítulo VII - Implementación

1. Uso de Multimedia	30
2. Configuración Básica del Hardware Requerido	31
3. Etapas del Desarrollo de la Aplicación	32
4. Inconvenientes sorteados en la Implementación	33
5. Estructura del Libro	33
5.1. Índice General	34
5.2. Índice de Capítulo	36
5.3. Página Tipo	38
6. Convenciones Utilizadas	40

Capítulo VIII - Evaluación

41

Conclusiones

43

Apéndice

A. Manual del Usuario	44
B. Glosario	46

Bibliografía

51

Objetivo

La tecnología de multimedia es frecuentemente usada en el campo de la educación, ya que en general, un ambiente de aprendizaje multimedial provee alta flexibilidad, incremento y retención de la información, selección dinámica de los contenidos, etc.

Estas características permiten a los alumnos abandonar su rol pasivo y convertirse en los administradores de sus propios conocimientos.

La expectativa es trabajar con el recurso informático como un elemento adicional de motivación y comprensión del contexto operativo por parte del alumno. Esta modalidad de trabajo nos permite en consecuencia, acortar el intervalo de esfuerzo y tiempo existente para que el alumno pueda utilizar una computadora.

Basándonos en esto, el objetivo de fondo del trabajo, es brindar a los alumnos ingresantes a la carrera de Informática un producto interactivo para el autoaprendizaje del curso de ingreso. De esta manera podrá trabajar independientemente de los cursos tutoriales normales y llegar al mismo con una idea total de sus contenidos, que lo ayudará a comprender la clase de problemas que manejará.

Precisamente apuntando a lograr este objetivo, el producto será entregado a los alumnos en el momento de su inscripción en la carrera de Informática.



Introducción y Motivación

La iniciación de una carrera universitaria en general y en Informática en particular refleja una serie de dificultades de los alumnos que se podrían sintetizar en cinco puntos : [Ber95]

1. Falta de una adecuada orientación vocacional.

El alumno desconoce o tiene una percepción equivocada del tipo de estudios que va a realizar tanto en los fundamentos como en las aplicaciones profesionales.

2. Poco entrenamiento en pensar y expresar rigurosamente conceptos.

La escuela secundaria tiene en general un rol *informativo* donde el grado de conocimiento real alcanzado por el alumno es *voluntario*. Normalmente el alumno es poco entrenado para *interpretar, analizar y sintetizar*. La noción de *abstracción* resulta lejana a la clase de problemas planteados al alumno.

3. Dificultad de aprendizaje de los temas básicos.

Una cierta cultura general dificulta la motivación del alumno para el aprendizaje de temas básicos, que corresponden a fundamentos teóricos.

Normalmente esto requiere capacidad de abstracción y aceptación de reglas rigurosamente especificadas, lo que es contrario a las costumbres de estudio adquiridas.

4. Escasa valoración por el trabajo sistemático.

El modo de trabajo de la escuela secundaria privilegia las soluciones inmediatas (intuición, prueba y error) al desarrollo deductivo o bien inductivo a través de un proceso de elaboración de resultados parciales que conduzcan a conclusiones.

5. Gran disparidad de conocimientos y formación previa.

Esta disparidad se aprecia no sólo en lo específico de la carrera universitaria, sino en el enfoque general de la *metodología de estudio* o mejor aún de la *metodología para interpretar, analizar y resolver problemas*.

Normalmente frente a alumnos con estas dificultades, los primeros cursos de una carrera universitaria de informática intentan introducir la programación estructurada (desde un paradigma imperativo), fundamentada en la expresión y especificación lógica de algoritmos. El resultado general es una alta tasa de deserción.

Otro aspecto interesante a tener en cuenta es la *velocidad del cambio tecnológico*. El alumno al terminar su carrera (y durante su desarrollo) se encuentra con un conjunto de nuevas herramientas, que incluso pueden resultar desconocidas para sus propios docentes.

En este contexto el tema multimedial ha tenido un auge muy importante en los últimos años, y es una herramienta que facilita la motivación y la definición vocacional de los adolescentes en la etapa pre-universitaria. [Sha92]

La motivación es el modo de despertar en el alumno el interés por aprender y conocer el tema en profundidad.

La falta de motivación produce reacciones negativas del estudiante que se sintetizan en "*falta de atención*" al tema. [Ber95]

Tal vez el elemento de mayor expectativa para el alumno ingresante a Informática es la posibilidad de utilizar más o menos creativamente una computadora.

En este contexto, resulta importante incluir de algún modo herramientas informáticas que favorezcan un contacto del alumno con su realidad futura.

La idea de interacción entre un sistema que ofrece alternativas y un alumno que las recorre libremente, es especialmente atractivo en el ámbito psicopedagógico.

Se sabe que los seres humanos utilizan todos sus sentidos para recoger y procesar la información referida al mundo que los rodea. Pierden interés en aquellos objetos inmóviles, el color les transmite información para comprender que grado de importancia tienen las cosas, y el sonido les da la pauta de cómo deben sentirse ante lo que están mirando. [Sha92]

Este trabajo pretende, a través de la dinámica visual de presentar la información, captar y mantener la atención de los alumnos, como así también lograr que se involucren participando de forma activa.

No debemos olvidar que el acceso ágil a la información y al conocimiento es un punto clave para el éxito en el ámbito del aprendizaje.

Por otro lado la mente humana es asociativa por naturaleza, el proceso de pensamiento no construye ideas una a la vez sino más bien sobre fuentes diferentes al mismo tiempo y en distintos puntos en paralelo, cada idea contribuyendo con las demás. Esta estructura asociativa de la mente es muy diferente de la forma lineal en que se organizan los libros.

Multimedia es un ambiente de representación del conocimiento extremadamente flexible que es análogo en muchas formas a la asociatividad de la mente humana. Nos permite organizar, recuperar, restaurar y reestructurar la información en todos los campos.

Si analizamos la tecnología multimedial con un enfoque educativo, encontramos algunas ventajas y atractivos : [Ber95]

- **Tratamiento de información no secuencial.** Esto permite relacionar conceptos para el alumno y construir nuevos conocimientos relacionando libremente conceptos previamente almacenados o generados.
- **Adaptación a las capacidades del alumno.** Los sistemas multimediales no responden a un algoritmo del tipo "*estímulo-respuesta*", sino que permiten "*navegar*" libremente entre opciones y detenerse en cada una de ellas según los intereses y aptitudes del alumno. Además provee la posibilidad de elegir entre caminos alternativos de aprendizaje sacando provecho de las distintas características de los alumnos, como por ejemplo: conocimientos previos, capacidad, etc.
- **Posibilidad de interactuar fuertemente con los sistemas.** El alumno puede interactuar con todos sus sentidos y tener respuestas relacionadas con cada uno de ellos, ayudándose a reflexionar y obtener conclusiones de un tema (y por ende, incorporando naturalmente conocimiento).
- **Integración natural de múltiples medios.** Diversos elementos tecnológicos ya conocidos por el alumno se incorporan e integran en un sistema multimedial, favoreciendo la vinculación del alumno con el tema propuesto, en pro de un aprendizaje significativo. La utilización de los distintos medios no se debe restringir a funciones de enriquecimiento del ambiente sino también debe abarcar las funciones concernientes a la transmisión del conocimiento, es decir, hacer uso de los diferentes medios para resaltar información, logrando así una mayor percepción del alumno en aquellos conceptos que el diseñador considera más importantes.

Multimedia en Educación

1. Introducción

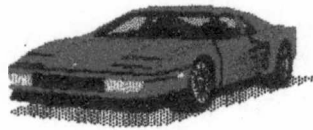
El término multimedia se refiere de una manera general a la integración de múltiples medios, *texto, gráficos, animaciones, fotografías, video, sonido y música* que se usan en aplicaciones basadas en computadora.

La posibilidad de tener a disposición distintos tipos de representación de la información, constituye una gran ventaja que ofrece multimedia en pro del aprendizaje significativo. Beneficio que consiste en poder expresar cualquier concepto en la forma más apropiada, no limitándose a una forma textual. La combinación de estos elementos mejora la comunicación, permite crear aplicaciones que no serían posibles sin sonido o movimiento y aún más, combina el impacto de la TV con el poder de la computadora personal. [TII96]

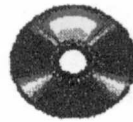
Nuestro trabajo de grado consiste en utilizar los beneficios que provee multimedia para implementar el curso de ingreso, y obtener de esta manera una aplicación en la cual el alumno pueda navegar y explorar libremente.

En este tutorial no se intenta remplazar la acción de otros medios educativos cuya calidad está bien demostrada, sino más bien, *complementarlos*. Su fin primordial, es ser utilizado como una herramienta que despierte en el alumno el suficiente interés para hacer suyo el conocimiento.

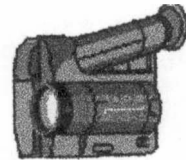
Multimedia



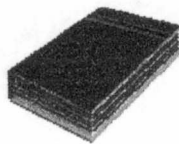
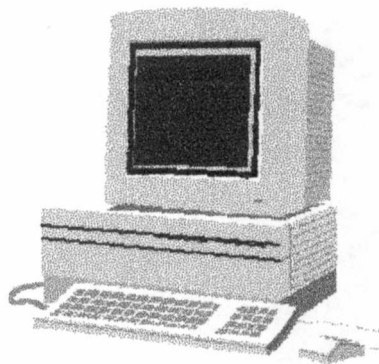
Gráficos



Sonido



Video



Texto



Animación

La base de una aplicación multimedia la componen el texto, los dibujos y gráficos importados. A esto se puede agregar sonido, video y animación.

2. Aplicaciones Multimediales

Las aplicaciones multimediales se distinguen por los siguientes factores :

- **Interactividad** : logra que el alumno deje de ser un elemento pasivo dentro del proceso de enseñanza-aprendizaje para convertirse en un elemento *activo y participe* del mismo.
- **Control del ritmo y secuencia** : permite al alumno avanzar más rápido o despacio según sus necesidades o inquietudes, explorar todas las opciones utilizando el tiempo necesario, revisar conceptos aprendidos, etc.
- **Motivación** : la riqueza del ambiente en la que el alumno se desenvuelve, sirven como una característica novedosa que mantiene la atención del alumno.

Un desarrollador multimedial es simultáneamente un *diseñador, escritor, programador, artista y director*. Sus aplicaciones son : [TooII96]

- **Eficientes**, ya que se puede reemplazar información que se lee, con información que se puede ver y escuchar, tal como un video clip.
- **Directas**, porque se brinda información utilizando el mejor medio, en el caso de una aplicación de enseñanza de un idioma, se podría utilizar la voz de un nativo para demostrar la pronunciación.
- **Personales**, pues se diseñan aplicaciones combinando de una manera creativa los elementos que la componen.
- **Interactivas**, ya que los usuarios pueden controlar el flujo de la información.

Los principales objetivos que se pretenden alcanzar cuando se está diseñando una aplicación son que ésta sea *extensible, modificable, reusable*.

Análisis del Curso de Ingreso a Informática: Análisis y Expresión de Problemas

1. Introducción

[Ber95]

En el Curso de Ingreso se introduce al alumno en un ambiente de aprendizaje diferente al modelo de la escuela secundaria. El alumno debe preguntar, decidir como avanzar en el aprendizaje y autoevaluarse permanentemente. El docente no lo conoce y posiblemente no llegue a identificarlo durante el transcurso del mismo.

Cada vez más se aprecian las diferencias de formación e información con las que el alumno llega a la etapa universitaria luego del nivel medio. A esto se agrega la gran “*dispersión*” existente en el contenido de las materias de computación del secundario. Es por ello que el Curso de Ingreso actúa como un nivelador de conocimientos.

La finalidad del Curso de Ingreso, de Análisis y Expresión de Problemas, es dar un conjunto de conocimientos básicos, utilizando un lenguaje formal que se aproxime al meta-lenguaje con el que se inicia el curso de primer año.

La dificultad de los alumnos para pensar un enunciado, dedicarle un tiempo para comprenderlo, analizar los elementos con los que cuenta para resolverlo y luego expresar racionalmente una solución constituyen las principales falencias provenientes de la formación secundaria que se intentan corregir.

Generalmente, el alumno tiende a resolver un problema enfrentando al mismo como un todo. La consecuencia de ello es que, la posterior resolución es, o bien incorrecta o bien incompleta. Introducir al alumno en técnicas que permitan pensar el problema, descubrir sus componentes principales y su interacción entre las mismas, favorecen una mejor comprensión y posterior solución del mismo.

Por lo expuesto, en este tutorial se abordan algunos conceptos útiles que sirven como punto de partida para el curso anual dedicado a Algorítmica y Programación :

- **Análisis de problemas solubles por computadora**, es decir expresables como algoritmos, enfatizando en la modelización y modularización de los mismos.
- **Expresión de los problemas en forma sintética, precisa y ordenada**, documentándolos en forma adecuada.
- **Obtención de la solución del problema**. Normalmente la solución no es única, sino que existen infinitas. La que se elija debe ser eficiente, y además debe estar escrita en forma clara y entendible.

2. Objetivos de los Capítulos

[Ing95]

2.1. Capítulo 1

- Introducir el concepto de *problema* y su representación mediante modelos para arribar a una solución.
- Identificar las cuatro etapas por las que atraviesa un problema :

- 1) **Formulación Precisa**
- 2) **Desarrollo de un modelo "matemático" que represente el problema**
- 3) **Análisis del modelo y sus variantes**
- 4) **Computación de una solución**

Rara vez se menciona la primera etapa en cursos de educación superior, siendo el desarrollo del modelo el tema principal. La educación científica usual se concentra en la tercera y cuarta etapas, las cuales constituyen lo que llamaríamos la "*solución del problema*" propiamente dicha.

Estas etapas no son solo válidas para el análisis de los problemas científicos, sino que también lo son para los problemas de tipo común y corriente.

Una de las mayores causas de dificultades es que la naturaleza del problema no se conoce con precisión. Podemos esperar esto en algunos campos "*no científicos*", tales como las relaciones personales, la delincuencia juvenil y la pobreza. Sin embargo, también ocurre con mucha frecuencia en los campos científicos o de ingeniería, supuestamente bien determinados y formalizados.

- Analizar el origen de los problemas científicos, destacando la importancia de la buena selección del modelo matemático, de la enunciación de las suposiciones o las idealizaciones hechas y de la revisión de que las soluciones sean razonables.
- Descomponer la solución de un problema en la solución de un número manejable de problemas más simples llamados *módulos*, enumerando las ventajas de la modularización.
- Definir las metodologías de resolución de problemas *Top-Down* y *Bottom-up*.
- Resaltar las ventajas de la *documentación*, la que debe ser parte integrante del proceso de resolución de un problema.

2.2. Capítulo 2

- Introducir el concepto de *lógica*, la que permite partiendo de verdades llegar a una nueva verdad. Para poder realizar esto, la lógica pretende habituarnos a una forma de pensamiento que nos conduzca a tener precisión y generalidad. Esta precisión se logra a través de símbolos cuya interpretación es rigurosamente exacta, eliminando de esta manera las ambigüedades que presenta el lenguaje común y corriente.
- Describir la existencia de dos tipos importantes de razonamientos: el *inductivo* y el *deductivo*.
A pesar que el razonamiento inductivo no siempre nos lleva a resultados exactos, es realmente un método valioso para descubrir conclusiones posibles.
- Definir el término *proposición* y su clasificación en proposiciones atómicas y moleculares. Describir los términos de enlace o conectivos lógicos empleados para obtener proposiciones moleculares a partir de la unión de varias proposiciones atómicas. Identificar las proposiciones y representarlas simbólicamente.
- Construir las *tablas de verdad* asociadas a las proposiciones para evaluar su valor de verdad. Para ello se debe determinar cuántas y cuáles son las proposiciones atómicas que componen la proposición. La cantidad de combinaciones de valores de verdad está directamente relacionado con la cantidad de proposiciones atómicas.
- Describir el armado de las tablas de verdad de: la *conjunción*, la *disyunción*, la *negación*, el *condicional* y el *bicondicional*.
- Definir formalmente cuándo una proposición es una *Tautología*, una *Contradicción* o una *Contingencia*.
- Enunciar la prioridad de los *términos de enlace* dentro de una proposición, y la utilización del paréntesis para alterar este orden natural de resolución.
- Analizar otras formas de enunciar los condicionales.
- Introducir los conceptos de *Implicación* y *Equivalencia Tautológica* y la demostración de estos conceptos a través de la construcción de las Tablas de verdad.

2.3. Capítulo 3

- Definir una parte importante de la lógica formal: *Inferencia y Deducción*.
- Introducir los conceptos de: *reglas de inferencia, premisas, conclusiones, consecuencia lógica, razonamiento, deducción formal*.
- Describir las siguientes reglas de inferencia que rigen el uso de los términos de enlace. :

- ✓ **Modus Ponendo Ponens**
- ✓ **Doble Negación**
- ✓ **Modus Tollendo Tollens**
- ✓ **Adjunción**
- ✓ **Simplificación**
- ✓ **Adición**
- ✓ **Modus Tollendo Ponens**
- ✓ **Ley del Silogismo Hipotético**
- ✓ **Ley del Silogismo Disyuntivo**
- ✓ **Leyes de De Morgan**

- Determinar si es posible arribar o no a una determinada conclusión, a partir de proposiciones dadas y las reglas de inferencia, mediante el empleo de la deducción proposicional.

2.4. Capítulo 4

- Incorporar la terminología necesaria para especificar los problemas del mundo real correctamente.
- Introducir el concepto de *algoritmo*.
- Enunciar las propiedades de un algoritmo :
 - Un algoritmo no debe ser ambiguo
 - Debe arribar a la solución del problema en un número finito de pasos
- Definir la palabra instrucción como sinónimo de paso elemental.
- Describir los elementos que componen un algoritmo :
 - Secuencia de acciones
 - Estructuras de Control : Selección, Iteración
- Aplicar el concepto de módulo para la resolución de subproblemas, comparando las técnicas Top Down y Botom Up. Definir los términos pre-condiciones y post-condiciones.

2.5. Capítulo 5

- Introducir el concepto de *lenguaje de Expresión de Problemas* enunciando las características que lo distinguen.
- Definir los términos *sintaxis* y *semántica* en un lenguaje, mencionando algunas reglas generales de sintaxis.
- Incorporar los conceptos de *Control* y *Datos*.
- Enunciar los motivos que hacen necesario la incorporación de comentarios lógicos a los algoritmos :
 - Aumentar la claridad y legibilidad del algoritmo
 - Reducir el costo de mantenimiento
- Ejemplificar el seguimiento de la *evolución de una solución*: partiendo de una pre-condición se analiza cada paso sobre los datos hasta arribar al cumplimiento del objetivo del algoritmo, reflejado en la post-condición.
- Describir la *representación de los datos*, que estará dada por las características de los objetos que nos interese conocer.
- Analizar el *esquema de un algoritmo completo*.

2.6. Capítulo 6

- Describir el método de *diseño de programación modular*, aplicarlo a un problema concreto y enumerar sus ventajas.
- Definir los términos que componen un módulo: *Proceso...FinProceso*, *Nombre*, *Datos que recibe y/o produce*, *Datos propios del Proceso*, *Instrucciones u órdenes*.
- Analizar tres problemas específicos, implementando el algoritmo asociado y describiendo en detalle la ejecución del mismo.
- Identificar las dos formas existentes de comunicación entre módulos.

Elección de la Herramienta

Para elegir la herramienta más adecuada para desarrollar la aplicación, se evaluaron varias alternativas. Una de ellas fue *Power Builder*, que se descartó por ser una herramienta (que en ese momento) no estaba disponible. Otra alternativa considerada fue *Visual Basic*, que se desestimó por ser un lenguaje básicamente orientado a aplicaciones convencionales.

Finalmente, nos decidimos por *Multimedia Toolbook* por varios motivos, entre los que podemos mencionar, el hecho de ser una herramienta orientada a objetos que brinda un ambiente óptimo y potente para el desarrollo de aplicaciones multimediales, su disponibilidad y por contar con un lenguaje de programación muy completo que permite programar objetos multimedia.

Análisis de Toolbook

1. Introducción

Toolbook es una herramienta para construir software de entrenamiento, cursos y presentaciones, aprovechando sus capacidades gráficas y de animación.

En Toolbook se llama *objeto* a los botones, campos, hotwords, gráficos, grupos, páginas, backgrounds, o libros. Cualquier objeto puede tener un script y todos los objetos tienen propiedades. Las propiedades son los atributos de un objeto. Una propiedad es un comportamiento preprogramado que puede ser modificado en formas predefinidas.

Con Toolbook se puede presentar información gráficamente, lo cual resulta más efectivo que presentar solo texto. Los *dibujos*, las *imágenes escaneadas*, la adición de *color* y aún la *animación* y el *sonido*, incrementan el impacto del texto que brinda el libro.

Toolbook es sumamente *flexible*, ya que los objetos una vez que han sido creados se pueden cambiar de tamaño así también como variar los patrones de diseño. Los botones son muy útiles para implementar la navegación a través del libro o ejecutar otras acciones.

Las opciones de linkeo de Toolbook para botones y hotword, permiten crear un script para navegación, simplemente identificando la página que Toolbook debe desplegar cuando el lector clickee el botón o la hotword.

2. Posibilidades que brinda Toolbook

[Too89]

- Uno de los mayores poderes de Toolbook es su habilidad para combinar texto y gráficos fácilmente en la misma página. Se incluye texto en los libros tipeándolos en los campos. Se pueden ubicar los campos y gráficos uno al lado del otro y aún solaparlos en la misma página.
- Producir información para compartir con otros ya que cualquier usuario Toolbook puede leer el libro que uno produce.
- Reunir información para el propio uso, por ejemplo construir un libro de direcciones, conservar los registros del personal de una empresa, o armar un catálogo de la colección de autos que uno posee, con fotos escaneadas de los mismos.

- Crear prototipos para aplicaciones que se quieran desarrollar.
- Se puede modificar un libro o agregarle nuevos contenidos después de que éste ha sido creado, con el fin de cambiar la información que posee, la forma en que está organizado o las tareas que ejecuta.
- Provee un *Help On-line* que brinda información escueta sobre comandos e instrucciones de Toolbook.
- Tiene herramientas propias para “*debugging*” o búsqueda y corrección de errores en los scripts.
- Para ayudar a los usuarios a buscar información en la aplicación rápidamente, Toolbook brinda a los diseñadores la posibilidad de incorporar la búsqueda de texto mediante palabras, frases o palabras claves.
- Permite chequear la ortografía de la aplicación realizada.
- Cuenta con un “*system book*”, que es un libro cuyo script está disponible en la jerarquía de objetos mientras aplicaciones Toolbook están corriendo.
- Permite acceder a procedimientos escritos en C, Pascal o lenguaje Assembler (DLLs).

3. Lenguaje de Programación (OpenScript)

[Ope89]

OpenScript es el lenguaje de programación de Toolbook. Ofrece equivalencias para todos los comandos del menú y las acciones en Toolbook y además incluye funciones aritméticas y estructuras de control.

Algunas de las ventajas provistas por OpenScript son :

- **Se programa menos.**

Los objetos son creados por Toolbook con un comportamiento inicial, solo se programa aquél comportamiento que se le agregue o que difiera del original. También gran parte del comportamiento de los objetos, puede ser cambiado simplemente cambiando el valor de sus propiedades.

- **Se programa en pequeñas unidades.**

Los *scripts* son por su naturaleza, módulos de programa. Un script está separado de otros scripts y su propósito es claro y consiste en definir el comportamiento de un objeto. Además un script está dividido en módulos llamados "*handlers*", y cada handler tiene un propósito muy específico. Esta modularidad, hace más fácil la programación.

- **Lo que se escribe en un script es rehusable en cualquier otro.**

Un script existente podría ser similar a un script que se necesite para un nuevo objeto. En este caso, se puede copiar el script existente al nuevo objeto y entonces modificar el script como sea necesario. Rehusar los script existentes ahorra tiempo y ayuda a evitar errores.

Un "*handler*" del script de un objeto es ejecutado cuando el objeto recibe un mensaje específico. Como se pueden enviar mensajes desde varios objetos, en gran variedad de formas se puede disparar la ejecución de un handler dentro de una aplicación. Sin embargo, no se necesita repetir el "*handler*" en varios lugares.

Como los objetos de Toolbook están ordenados en una jerarquía, esto significa que aquellos objetos ubicados más abajo en la jerarquía pueden acceder a los scripts que están en una jerarquía más alta. Ubicando instrucciones que afecten a varios objetos, en el script de un objeto que está próximo al tope de la jerarquía, se puede reducir el tamaño y número de scripts que se necesiten escribir.

- **Se pueden crear fácilmente atractivas interfases con el usuario.**

Usando OpenScript, se puede crear en forma fácil una útil y atractiva *interfase gráfica* con el usuario. Como los scripts están asociados a los objetos, es fácil incluir instrucciones en los scripts que afecten la forma en que los objetos se muestran y se comportan en la pantalla. También se pueden diseñar y cambiar los menús. Estas tareas pueden resultar tediosas en los lenguajes de programación tradicionales.

- **Es semejante al lenguaje natural.**

Los comandos OpenScript y otros términos son frecuentemente palabras en inglés, y las instrucciones en OpenScript son en general similares a las sentencias en inglés. Esto permite que los script se escriban en una forma muy natural, similar a como se podría escribir una oración en inglés.

- **Es extensible.**

Aunque el lenguaje OpenScript es poderoso, el programador puede crear y usar sus propias propiedades, funciones y mensajes, para extender la funcionalidad de OpenScript.

Además se pueden escribir procedimientos en C, Pascal o lenguaje de programación Assembler, y acceder a ellos en OpenScript como Windows Dynamic Link Libraries (DLLs).

- **Toolbook ayuda a corregir errores en los script.**

Cuando se escriben script, Toolbook controla que se sigan las reglas para escribir sentencias de manera tal que él pueda entenderlas. Si se comete un error, Toolbook da un mensaje de error y permite corregir el mismo antes de seguir avanzando. Además, para chequear estas reglas, Toolbook viene con una herramienta especial, "*the Debug window*", que provee acceso al script mientras se esta ejecutando. Esto ayuda a aislar y resolver problemas en el script.

- **No tiene tipos de datos.**

Los datos son interpretados según su ubicación, es decir de acuerdo a los comandos, funciones y operadores donde son usados. No se necesita indicar el tipo de un valor ni tampoco especificar que tipo de dato puede contener una variable en particular.

- **No tiene largos ciclos de compilación.**

La compilación se realiza cada vez que se escribe o se hace un cambio en un script. La performance se incrementa porque la ejecución del script está separada de la compilación.

4. Aplicación

Una *aplicación* es un programa de computadoras que ejecuta tareas específicas. Con Toolbook se pueden crear aplicaciones más fácilmente que con la programación convencional. Esto se debe tanto a su interfase gráfica, que hace más amigable la interacción con el usuario, como a las características de programación orientada a objetos que posee.

Una aplicación Toolbook consiste de uno o más libros que son diseñados para un propósito específico tal como entrenamiento , información gerencial o entretenimiento.

Con OpenScript, se pueden crear aplicaciones tan diversas como : [Ope89]

- Una aplicación de cálculo científico que ejecute varias funciones matemáticas y que calcule el pago de préstamos hipotecarios.

Una aplicación que enseñe los elementos de la tabla periódica, que funcione de manera que cuando el lector clickea uno de los elementos, Toolbook despliega una página con la ilustración de la estructura atómica del elemento y cierto texto que lo describe.

- Una aplicación estadística que permita crear el gráfico asociado a los datos ingresados por el lector. El gráfico puede ser de varios tipos: *barras*, *columnas*, *líneas*, *áreas* o *tortas*. En una página el lector ingresa los datos a ser usados para crear el gráfico, y clickea un botón. Cuando esto ocurre, Toolbook va a una página en blanco y traza el gráfico.

4.1. Técnicas para optimizar las aplicaciones

Una vez conocidos los fundamentos de OpenScript se pueden construir aplicaciones más efectivas aprendiendo a organizar los script. También se puede extender el poder de las aplicaciones con las siguientes técnicas :

- Definir funciones propias.
- Definir propiedades.
- Agregar menús y comandos a la barra de menú de Toolbook.
- Extender las capacidades de las aplicaciones llamando a rutinas de la Dynamic Link Libraries (DLLs).
- Intercambiando información con otras aplicaciones Windows usando Dynamic Data Exchange (DDE).

5. Tipos de información Multimedia

Para crear una aplicación, usualmente se comienza con texto, dibujos y gráficos importados. Luego se pueden adicionar tres tipos de información multimedia, *sonido*, *video* y *animación*. [TII96]

5.1. Sonido

En Toolbook, el sonido está disponible en tres formatos, archivos de audio wave, archivos MIDI (Musical Instrument Digital Interfase) y CD audio. El formato a elegir depende del tipo y la calidad de sonido que se desee ejecutar, del espacio disponible para los archivos de sonido y si se distribuirán o no CDs audio con la aplicación.

- **Wave audio**

Un archivo de audio wave (.WAV) contiene sonido grabado con un micrófono y almacenado en formato digital. Se usan archivos .WAV para incluir efectos de sonido y música de alta fidelidad.

Los archivos .WAV pueden ser de gran tamaño, dependiendo de la calidad del sonido grabado.

- **MIDI**

Los archivos MIDI se usan para música que será ejecutada en instrumentos electrónicos tales como sintetizadores. Un archivo MIDI (.MID) almacena una versión electrónica de la partitura musical, conteniendo información sobre cada nota lo que incluye *tono, duración y ruido*.

Cuando se ejecuta un archivo .MID, Toolbook envía el archivo a un sintetizador (ya sea el pequeño sintetizador de la plaqueta de sonido o un sintetizador externo), usando un software llamado secuenciador que lee la información digital y recrea la música. La calidad del sonido depende de cómo el sintetizador pueda recrear los sonidos representados por el archivo .MID.

Incluir un archivo .MID en vez de un archivo .WAV en la aplicación tiene ciertas ventajas, un archivo .MID requiere mucho menos espacio de almacenamiento, además el sonido MIDI es más fácil de manipular. Sin embargo, debido a que la información MIDI es ejecutada en un sintetizador, los archivos .MID sólo se pueden usar para almacenar música y ciertos tipos de efectos de sonido.

- **CD audio**

El CD audio es almacenado digitalmente en un CD y luego ejecutado a través de los parlantes conectados. CD audio es el formato de los comúnmente conocidos CD de música. El sonido almacenado en un CD audio es el sonido de más alta calidad en la computadora.

Cuando se ejecuta un CD audio, el drive CD Rom y no la CPU hace la mayoría del trabajo en la ejecución del sonido. Esto minimiza el impacto que el sonido tiene en otras acciones que la computadora puede ejecutar simultáneamente. Sin embargo como los drives CD Rom pueden tomarse una gran cantidad de tiempo para localizar la ubicación correcta en el disco, el CD audio resulta inapropiado para aplicaciones donde se requiere una respuesta inmediata.

5.2. Video

En Toolbook se puede usar video desde tres fuentes, *video digital* almacenado en archivos en el disco duro o en el CD, *videodisc* o *videotape*. Como con el sonido, el formato a elegir depende de la calidad del video y del nivel que se le quiera dar a la aplicación.

- **Archivo de video digital**

Los archivos de video digital contienen una *imagen de video digital* y su *sonido*. En Toolbook se pueden ejecutar los archivos de video digital en cualquier formato MCI compatible. Ejecutar los archivos de video digital no requiere hardware especial, sólo los drivers apropiados. La CPU de la computadora ejecuta el video.

Se pueden copiar los archivos de video digital fácilmente e instalarlos con la aplicación. Sin embargo, los archivos son frecuentemente muy grandes (un minuto de video puede ocupar tanto como 20 MB de espacio de disco). Para usar archivos de video grandes, la aplicación deberá distribuirse en CD Rom.

- **Videodisc**

Si la reproductora de video puede ser controlada desde la computadora, se pueden ejecutar videodisc en la aplicación. La salida de la reproductora de video puede ser visualizada en una pantalla separada o en el mismo monitor como video overlay.

La técnica "*overlaying*" consiste en agregar una plaqueta especial en la computadora, con ella se integra la señal de videodisc con información, en el monitor de la computadora.

Cuando se ejecuta video mediante esta técnica, Toolbook designa el área de la pantalla donde el video se mostrará. La señal de video es entonces interceptada por la tarjeta overlay, la cual reemplaza el área designada con datos del videodisc.

Ejecutar video desde un videodisc (en contraste a ejecutar video desde un archivo de video digital) tiene mínimo impacto con otras operaciones, porque el reproductor de videodisc y la tarjeta de overlay hacen la mayoría del trabajo. En consecuencia, los videodisc pueden contener video de mayor calidad que los archivos de video digital y con mayor resolución de color. Sin embargo, como los videodisc se deben ejecutar en un reproductor especial, ellos son más apropiados para propósitos especializados.

- **Videotape**

Al igual que en los videodisc, si la reproductora de video puede ser controlada desde la computadora, se pueden ejecutar videotapes en la aplicación. La salida de la reproductora de video puede ser visualizada en una pantalla separada o en el mismo monitor como video overlay.

Como con los videodisc, ejecutar video desde un videotape tiene poco impacto en otras operaciones, porque la unidad en sí misma hace la mayoría del trabajo. Similarmente, la calidad del video es mejor que la de los archivos de video digital. Los videotapes tienen una gran capacidad de almacenamiento (hasta ocho horas versus una o dos de los videodisc), son fáciles de grabar y pueden ser editados usando un segundo grabador de videotape.

Los videotapes tienen ciertas desventajas. Como los videodisc, requieren un hardware especial para soportar overlaying, no son aconsejables para acceso directo por el tiempo requerido para localizar algo en particular (por ejemplo rebobinar un videotape puede tomar hasta seis minutos). Los videotapes están más sujetos a daños que los videodisc o información en video digital, además la calidad de la imagen se degrada con el uso reiterado.

5.3. Animación

Como las películas y el video la animación despliega una serie de imágenes en rápida sucesión dando la ilusión de movimiento. No obstante la animación está basada en dibujos, mientras que el video y las películas son grabaciones de sucesos de la vida. Como los archivos de animación almacenan menos información, son usualmente más chicos que los archivos de video de duración equivalente.

La animación no requiere hardware especial para ser ejecutada solo la apropiada device driver.

6. Objetos de Toolbook

[Too89]

Toolbook es *orientado a objetos*. Los botones, campos y gráficos que se crean son todos objetos distintos. Cada objeto tiene una colección de características e identificadores llamados propiedades que definen la apariencia del objeto y su comportamiento. Por ejemplo, una de las propiedades de un botón es el estilo de su borde. Una vez creado el botón, es fácil cambiar la propiedad que define el borde del botón a fin de modificar su aspecto en la página.

Se puede cambiar el tamaño de un objeto o moverlo de lugar sin afectar al resto de los objetos en la página. También se pueden agrupar objetos de manera de cambiar su tamaño y moverlos como una sola entidad. Se puede crear un objeto una vez y copiarlo tantas veces como se quiera, modificando la copia si se desea.

Los objetos sobre las páginas pueden ser distribuidos, alineados o superpuestos.

Los campos se usan para escribir texto, los gráficos para ejemplificar y los botones ejecutan acciones tales como ir a otra página.

Cualquier objeto puede tener un script que define como se comporta el objeto. Un script puede hacer algo tan simple como ir a la próxima página de un libro o mover un objeto a través de una página. También puede ejecutar acciones complejas como crear o cambiar objetos o páginas, enviar instrucciones a otros objetos o ejecutar cálculos.

El tamaño, ubicación y estilo de los objetos sobre una página definen la apariencia del libro. Los scripts de los objetos definen el comportamiento del libro cuando un lector trabaje con él.

Un objeto conserva su script y propiedades cuando se lo copia o mueve, esto facilita la construcción y modificación de libros, evitando reescribir los scripts para cada uno de los elementos del libro.

Para recorrer un libro se pueden usar los comandos del menú “*Page*” de Toolbook, independientemente de los botones, hotwords y gráficos que contenga el libro.

Cada objeto puede tener dos colores, un color que es usado para los bordes o texto y un color de relleno.

6.1. Propiedades de los objetos

Una forma de cambiar el comportamiento de un objeto es cambiar el valor de sus propiedades o atributos. Las propiedades de los objetos pueden ser modificadas utilizando el menú o escribiendo sentencias particulares en los scripts.

Cada objeto tiene su propio identificador ID, el cual es el número de identificación único que Toolbook automáticamente le asigna a un objeto cuando este es creado. Este identificador no puede ser modificado. Sin embargo puede ser usado en los scripts para referirse a un objeto.

Otra de las propiedades de un objeto es su número de “*layer*”. El “*layer*” es un número posicional que Toolbook le asigna a un objeto cuando es creado, ya sea en el foreground o en el background. El “*layer*” de un objeto se puede cambiar para alterar su ubicación en la página.

Se puede pensar a los “*layer*” como una pila de transparencias donde cada una de ellas contiene un campo, botón, gráfico o grupo. Hay una pila de transparencias para el background y otra para el foreground. Estas dos pilas juntas forman la página que el lector ve.

Aunque no es obligatorio se les puede asignar un nombre a los objetos, el cual no necesita ser único. Sin embargo darle un nombre único a cada objeto nos asegura que Toolbook pueda encontrar el objeto correcto cuando nos referimos a él dentro de un script. Si un objeto no tiene nombre, se puede usar su ID para referirse al objeto en los scripts.

6.2. Campos

Un *campo* es un objeto en el foreground o background que contiene texto. El lector puede agregar o modificar texto sólo en aquellos campos no bloqueados por el autor.

6.3. Campos Registro

El *campo registro* es un tipo especial de campo del background que puede ser editado tanto por el lector como por el autor. Al estar en el background estos campos aparecen en la misma posición, con el mismo estilo y el mismo tamaño en todas las páginas que comparten ese background.

Los campos registros se pueden usar para construir libros que trabajan como bases de datos tradicionales tales como listas de teléfono y direcciones. Un campo registro es el equivalente a un campo de una base de datos, y una página es el equivalente a un registro de la base de datos. Al ingresar texto en estos campos, el texto aparece en la página donde fue tipeado, de este modo, cada nueva página puede contener texto diferente en estos campos.

6.4. Botones

Los *botones* provistos por Toolbook, son aquellos objetos familiares que tienen un área activada que el lector puede clicar para ejecutar un script. Los botones son una de las herramientas básicas para construir páginas en Toolbook. En general, los botones se utilizan para navegar a través del libro yendo de una página a otra o para iniciar una secuencia de animación.

El "*label*" de un botón es el texto que aparece dentro del botón, usualmente unas pocas palabras que describen la acción que ejecuta, además el botón tiene un nombre que es un identificador usado para referirse a él en los script. Ambos, "*label*" y nombre pueden coincidir.

6.5. Area especial de texto (Hotword)

Una *hotword* es una área especial de texto que tiene asociado un script que hace que algo suceda, como ir a otra página, desplegar texto adicional o gráficos. Un ejemplo concreto consistiría en crear una lista de tópicos en un libro y dar a cada item un script que vaya a otra página y despliegue información específica de ese tópico.

Las hotword proveen una forma dinámica de conectar información que aparece en diferentes lugares. En vez de crear botones para todos los posibles "*links*" entre el texto de una página con los de otras páginas, se puede hacer que las piezas de texto actúen como botones definiéndolas como hotwords. Clickeando la hotword se ejecuta el script asociado a la misma. En la industria del software al texto con esta habilidad se lo conoce como "*hypertexto*". Para ayudar al lector a reconocer las hotwords, el autor les da diferentes formatos, usualmente aparecen en negrita o con otro tipo de letra. Además, el puntero cambia de forma cuando se encuentra sobre la hotword.

6.6. Gráficos

Un gráfico puede ser desde una simple línea o una figura hecha con la paleta de herramientas, hasta una imagen compleja creada con un utilitario de dibujo y luego insertada en el libro.

Hay dos clases de gráficos en Toolbook :

- Objeto Dibujado, es un gráfico creado con una de las herramientas de dibujo y almacenado como un dato matemático que describe su posición y apariencia.

- Objeto Pintado, es un objeto creado en otra aplicación e insertado, que se almacena como un mapa de pixels que construyen la imagen.

No se puede cambiar el color de los objetos pintados después que fueron insertados, ni tampoco su tamaño.

Un objeto pintado usualmente usa más memoria que un objeto dibujado. Para construir los objetos dibujados, Toolbook provee tres paletas de estilo que permiten especificar la forma del polígono, el estilo de la línea y el relleno.

Los gráficos pueden tener scripts como los botones y hotwords. Clickeando el gráfico se ejecuta el script asociado al mismo.

La última versión de Toolbook [TII96] incorpora dos nuevas herramientas de gran uso para *sonido*, *video* e *imágenes*.

6.7. Referencia a un segmento de fuente multimedia (Clip)

Un *clip* es una referencia a un segmento de cualquier fuente multimedia, sonido, video, animación o imágenes fijas. Solo la referencia es almacenada en la aplicación, los datos digitales permanecen almacenados en la fuente media original. Un clip incluye los siguientes atributos, *nombre*, *punto de inicio* y *punto de fin*.

Creando clips, se simplifica el mantenimiento de la aplicación multimedia y resulta fácil referirse a los clips en los scripts. Se puede editar un clip para cambiarle el punto de inicio o de fin, sin tener que editar los scripts asociados, porque los scripts hacen referencia sólo al nombre del clip o al identificador.

6.8. Escenario (Stage)

Se puede desplegar el medio visual, animación, video digital, videodisc, videotape e imágenes fijas usando un *stage*. Para ejecutar un medio visual en un stage, se necesita crear o importar un clip de un medio visual, crear un stage para desplegar el clip y entonces escribir un script que ejecute el clip. Un stage define el área de una página en la cual el clip de un medio visual aparecerá.

Se puede agregar consistencia visual en la aplicación poniendo un stage en el background, de manera tal que los clips de media visual se posicionen en el mismo lugar en cada página que usa el background. Se puede ejecutar más de un clip en un stage, pero sólo un clip a la vez.

7. Respuesta de un objeto ante un evento (Script)

[Ope89]

Un *script* es una serie de sentencias OpenScript divididas en partes llamadas "*handlers*", que definen como un objeto responderá a eventos particulares. Cuando un evento ocurre, tal como que el lector clickee un objeto, Toolbook ejecuta el handler para ese evento que se encuentra en el script asociado al objeto.

Los script son el poder detrás de una aplicación Toolbook.

Las sentencias de un script describen acciones para que Toolbook lleve a cabo tales como desplegar más texto, comenzar una secuencia de animación o ir a otra página o libro. Un script puede desplegar otros objetos sobre la misma página, imprimir páginas o reportes, ejecutar cálculos, etc.. Los botones, campos, gráficos, hotwords y hasta las páginas y los libros, pueden tener un script.

Los scripts también se pueden construir mediante las herramientas "*scriptwriting*" que permiten crear script sin necesidad de aprender OpenScript. El "*script recorder*" de Toolbook brinda la posibilidad de almacenar casi cualquier acción que se pueda ejecutar en Toolbook. Como una macro en una hoja de cálculo, el script recorder traduce las acciones en Toolbook en sentencias OpenScript.

Las personas que han escrito macros en Lotus123, Microsoft Excel o WordPerfect, tienen un conocimiento previo que los ayudará en el entendimiento de los scripts. Escribir y almacenar scripts es similar a escribir y almacenar macros. Sin embargo, los scripts son mucho más sofisticados que las macros. Escribiendo macros se pueden mejorar aplicaciones existentes, escribiendo scripts con OpenScript se pueden crear íntegramente nuevas aplicaciones.

Algunas de las acciones que se pueden ejecutar a través de un script son :

- Calcular promedios usando datos ingresados por los lectores y ejecutar acciones basadas en los resultados de los cálculos.
- Animar gráficos.
- Mostrar y esconder objetos en respuesta a acciones ejecutadas por el lector.
- Seleccionar comandos del menú.
- Agregar y remover menús e ítems de menús.
- Hacer una pregunta al lector y ejecutar una acción basada en la respuesta.
- Desplegar y formatear texto en un campo.



- Cambiar las propiedades de un objeto.
- Recuperar información del sistema como la fecha corriente.
- Ejecutar otras instancias de Toolbook y otras aplicaciones Windows.

7.1. Manejador (Handler)

Un *handler* es un grupo de sentencias con un propósito específico. Hay tres tipos de handlers :

- 1) Messages handlers
- 2) To set handlers
- 3) To get handlers

El tipo de handler más común es el 1). Un *message handler* es un grupo de sentencias que dicen a Toolbook qué acciones ejecutar al recibir un mensaje particular.

Un mensaje es una comunicación enviada a un objeto cuando ocurre un evento. Un evento es una acción del mouse o del teclado hecha por el lector o el autor o una acción interna hecha por Toolbook.

Cuando un objeto recibe un mensaje, Toolbook busca en el script del objeto un *message handler* que corresponda a ese mensaje. Si Toolbook encuentra uno, ejecuta las sentencias que figuran en el *message handler*. De esta manera se logra que los objetos se comporten de la manera que uno quiera ante determinados eventos.

Todas las sentencias en un script, excepto los comentarios, deben estar dentro de un handler.

Las sentencias dentro de un handler deben estar en el orden en que deban ser ejecutadas. Los handler dentro de un script pueden estar en cualquier orden, porque Toolbook llama a un handler cuando éste es necesario y no basándose en su posición dentro del script.

7.2. Búsqueda y Corrección de errores

Los errores en los script ocurren inevitablemente aunque Toolbook puede chequear la sintaxis para prevenir errores. Otros tipos de errores pueden suspender la ejecución del script, proveer resultados incorrectos o causar que la ejecución sea impredecible. Con "*the Debug window*" de Toolbook, se pueden identificar y aislar varios tipos de errores de modo de poder corregirlos.

"*Debugging*" es el proceso de encontrar y corregir errores en un script. Hay varias técnicas y cada una de ellas ayuda a identificar un tipo específico de error.

7.3. Tratamiento de mensajes

Si un objeto recibe un mensaje y su script no contiene un “*message handler*” para ese mensaje, Toolbook pasa el mensaje al objeto de arriba en la jerarquía hasta encontrar el manejador o el handler correspondiente. La jerarquía de objetos determina el orden en el cual los mensajes son pasados de un objeto a otro en búsqueda del handler o del manejador correspondiente. Por ejemplo, si el script de un campo no contiene un “message handler” para el mensaje enviado a ese campo, el mensaje es enviado al grupo al cual el campo pertenece (si el campo pertenece a un grupo), luego a la página donde está el campo y así siguiendo hacia arriba en la jerarquía de los objetos. Si Toolbook no encuentra el manejador correspondiente al mensaje en ningún script de los objetos de la jerarquía, el mensaje es resuelto por el sistema, ubicado al tope de la jerarquía.

A fin de evitar repetir el mismo manejador en los scripts de varios objetos y poder usar la jerarquía de objetos como ventaja, se debe ubicar el mismo en el lugar de la jerarquía que resulte más ventajoso.

El sistema contiene manejadores propios que ejecutan acciones específicas en respuesta a mensajes predefinidos. Los mensajes predefinidos son aquellos provistos como parte del lenguaje OpenScript, tales como *buttonUp*, *enterBook*, *make* y *destroy*. Cuando el sistema recibe un mensaje predefinido, Toolbook responde con las acciones específicas para ese mensaje.

Se pueden definir nuevos mensajes y los manejadores correspondientes. Ante estos mensajes el sistema no tiene manejadores que respondan, luego si el usuario no escribe el manejador correspondiente ocurrirá un error.

8. Libros y Páginas

[Too89]

Un *libro* es una colección de *páginas* que contienen información, ejecutan una tarea o se rellenan con información.

Cuando se construye un libro, creando páginas y uniéndolas unas a otras, se diseñan las páginas de manera de lograr un libro atractivo para el lector. Las páginas de un libro son almacenadas juntas como un archivo DOS. Todas las páginas en un libro tienen el mismo tamaño. La ejecución de las acciones del libro se lleva a cabo mediante los scripts.

El lector del libro ve a la página como una unidad, pero cada página se compone de un foreground y un background.

Como otros objetos en Toolbook, un libro tiene propiedades como el número de páginas y background que contiene, el tamaño de sus páginas, el nombre que es aquél con el que se graba. Además se le puede asignar una password para limitar el acceso al libro.

8.1. Partes de una página (Background y Foreground)

Al recorrer las páginas de un libro, el lector, puede observar que varios objetos aparecen en el mismo lugar en distintas páginas, pero otros objetos aparecen solamente en una página. Esto se debe a que algunos objetos están en el *background* y otros están en el *foreground* de una página.

Un background es un diseño de página compartido por un grupo de páginas en un libro. Cada libro tiene al menos una página y un background. Los objetos del background aparecen en la misma posición, estilo y tamaño en cada página que comparte ese background. Toolbook almacena una sola copia de cada background en el libro, por este motivo los objetos del background ocupan menos espacio en disco.

Un foreground es la parte de una página que contiene los objetos únicos de esa página.

8.2. Niveles de Trabajo

En Toolbook se puede usar un libro o construirlo. Para hacer estas dos tareas fácil de llevar a cabo, Toolbook tiene dos niveles de trabajo, nivel lector y nivel autor.

8.2.1. Nivel Autor

- Provee herramientas y comandos para crear y modificar objetos, páginas y nuevos libros.
- Provee una paleta de herramientas que permite crear campos, botones y gráficos.
- Mientras se está construyendo el libro, el autor pasa frecuentemente a nivel lector a fin de testear el funcionamiento del libro.
- El autor puede agregar menús, cambiar la forma en que se comportan los comandos y hasta esconder completamente la barra de menú de Toolbook para un libro en particular. El objetivo de esto es obligar al lector a recorrer el libro usando los botones, hotwords y gráficos existentes en él.
- Se pueden crear o cambiar campos, botones y gráficos para construir un libro, definiendo cómo los objetos lucirán y se comportarán en cada página.
- En nivel autor Toolbook agrega a la barra de menú otros menús y también comandos a los menús existentes. También se cuenta con tres paletas de herramientas :

- **Tool palette**, que permite crear objetos.
- **Style palette**, que permite elegir el estilo de una línea, la forma de un polígono.
- **Color palette**, que permite asignar colores a objetos, a campos y a objetos dibujados.

8.2.2. Nivel Lector

- Provee comandos para recorrer y agregar páginas a un libro, para tipear, editar y formatear texto en campos y para imprimir.
- Un lector ve los campos, botones y gráficos como parte del diseño de la página y como items asociados con acciones en un libro. Por ejemplo, un lector clickea un botón para cambiar páginas o clickea una hotword para ver más información o clickea un gráfico para iniciar una rutina de animación.

Implementación

1. Uso de multimedia

Es necesario hacer un análisis exhaustivo de los distintos medios existentes (y a nuestro alcance) para determinar cuáles realmente se necesitan para poder expresar de la manera más adecuada el concepto que se desea enseñar. No por el hecho de tener a nuestro alcance diversos medios de comunicación (sonido, video, animaciones, gráficos, texto) debemos hacer uso de todos ellos, ya que de esta forma seguramente obtendríamos un material muy rico, novedoso pero quizás no cubra el objetivo para el cual fue desarrollado, *enseñar*.

Un sistema muy rico en medios puede dispersar la atención del alumno si los mismos fueron utilizados por el mero hecho de “*decorar*” el ambiente y no de enriquecerlo en su contenido. Por ejemplo, si bien un video es agradable y el uso del mismo llamaría la atención por lo novedoso, no aportaría información útil para una aplicación que enseñe el curso de ingreso, siendo más adecuado para otras áreas tales como : Geografía, Arte, Historia, etc..

Para nuestra aplicación decidimos hacer uso de :

- **Animaciones** : fundamentalmente para ejemplificar algún concepto, está demás aclarar que es posible mejorar y aumentar la cantidad de animaciones en el tutorial, pero debido al tiempo disponible se realizaron unas pocas a modo de ejemplo. Cabe destacar que una de sus principales utilidades es la explicación de procesos complejos, como por ejemplo programas o algoritmos de computación. En nuestro tutorial, se podría incorporar animación a los problemas referidos a los distintos recorridos que debe realizar el robot.

Al usar animaciones es necesario establecer :

- ✓ **El rol que cumplirán** (reforzar, entretener, destacar).
- ✓ **Cuándo se iniciarán** (a qué tipos de eventos estarán asociadas).
- ✓ **Su duración**.
- ✓ **Si estarán acompañadas de sonido** (en cuyo caso debe decidirse cuándo).

- **Sonido** : para ayudar a lograr ciertos objetivos es conveniente el uso del sonido. El mismo puede tener varios fines, entretener, destacar, separar o reforzar positiva o negativamente algún evento. En nuestro tutorial lo utilizamos como cortina cuando se abre y cierra una hotword, también para acompañar algunos ejemplos con el fin de llamar la atención del lector.
- **Color** : el color juega un rol importante en la interfase. La idea es lograr una armonía entre el color del fondo y los colores de los elementos de la pantalla. Por otro lado, el uso de diferentes colores o variaciones del mismo sirven para centrar la atención del usuario.
- **Texto** : que es el medio más utilizado en la enseñanza.
- **Gráficos** : existen diversos tipos de gráficos. Su uso depende del material que se intenta complementar y del dinamismo o estatismo que se le desee dar. En base a esto tenemos :
 - ✓ **Dibujos y Esquemas**, son útiles para trabajar conceptos, reforzar o dar contexto.
 - ✓ **Diagramas**, sirven para ilustrar procedimientos, relaciones entre partes o estados de un sistema.
 - ✓ **Tablas lógicas**, se usan para representar los valores de verdad de las proposiciones.

2. Configuración básica de hardware requerido

Describiremos en este punto todo lo concerniente al manejo eficiente del CD desarrollado para el aprendizaje del Curso de Ingreso.

Para la realización del libro se utilizó una PC 486 monitor color con sound blaster incorporado. La configuración del monitor es Super VGA , fuente pequeña.

Cabe aclarar que si el libro es observado con otros recursos que no sean los antes mencionados, puede variar el efecto visual esperado, en detrimento del trabajo realizado. Por ejemplo algunos campos pueden verse con un color diferente al definido, o las páginas pueden superar el tamaño de la pantalla.

Esta herramienta corre desde Windows, contando con todos los beneficios que tiene Windows incorporados.

Consideramos que para recorrer nuestra aplicación el alumno no necesita una explicación previa del tema a ver, pero sí necesita conocer los conceptos básicos de Windows. Si el alumno lee previamente el manual del usuario manejará la aplicación de una manera más eficiente.

3. Etapas del desarrollo de la aplicación

El trabajo fue realizado cumpliendo las siguientes etapas:

- Se transcribió en la herramienta Toolbook el texto completo correspondiente a la parte teórica del Curso de Ingreso, construyendo para cada capítulo un libro diferente.
- El segundo paso fue armar un modelo de página para el libro, este modelo fue mejorado con el tiempo hasta obtener el definitivo. Más adelante comentaremos más detalles acerca de cómo debemos movernos en una página o qué posibilidades tenemos dentro de ella.
- El tercer paso fue ordenar la información de cada capítulo de manera de mejorar la comprensión de los temas por parte del alumno, en otro punto aclararemos cada una de las convenciones utilizadas.
- Una vez concluidos cada uno de los capítulos, se implementaron los índices de los mismos, de manera que desde el Índice se pueda acceder a la primer página de cada uno de los temas contenidos en el capítulo. Además, una vez recorrido un tema en su totalidad, se marca el mismo con un tilde en el Índice del capítulo, de modo de reflejar cuales son los temas ya vistos.
- El siguiente paso fue la unificación de todos los capítulos del libro en un único libro, con su correspondiente script, que cubre las necesidades propias de cada capítulo.
- Finalizado el paso anterior, se implementó el Índice General. Desde el mismo se puede acceder al Índice de cada capítulo mediante el número o directamente a un tema en particular de un capítulo. También permite ir a la página que contiene la bibliografía u optar por salir del libro.
- Una vez que todas las páginas realizadas conformaron un único libro se incorporó sonido a las hotword, y a otros objetos de manera de hacer más amigable y cautivante el recorrido del libro. La grabación del sonido se realizó con editores de sonido del ambiente Windows.
- El próximo paso fue realizar un ejecutable.
- Por último grabar un CD.

4. Inconvenientes sorteados en la implementación

Al finalizar la implementación de casi todos los capítulos tuvimos acceso a la nueva versión de Toolbook, Toolbook II, que brinda mejores herramientas para desarrollar aplicaciones multimediales, en particular la incorporación de sonido.

La conversión que realizó Toolbook II de cada uno de los capítulos trajo algunos inconvenientes, entre ellos :

- Algunos campos de texto que en la versión anterior se veían sin dificultad, en Toolbook II hubo que agrandarlos debido a que quedaba texto oculto al que no se podía acceder.
- Otro de los efectos fue la aparición de campos de texto con letra distinta a la definida originariamente. También se modificó la implementación de aquellos campos “single line” a los que accede el usuario, por ser distinto el tratamiento de los mismos en Toolbook II.

5. Estructura del libro

El libro se compone de una presentación del tutorial seguida por el Índice General, el cual permite acceder a los distintos contenidos del libro que están distribuidos en capítulos. Tiene además una página con la bibliografía empleada.

Indice General

The image shows a graphical user interface for a 'General Index'. At the top, the title 'INDICE GENERAL' is displayed in a large, serif font. Below the title, there are two main columns of text. The left column contains a vertical list of numbered options: 1 **Conceptos Generales**, 2 **Conceptos de Lógica**, 3 **Razonamiento Lógico**, 4 **Terminología de Problemas**, 5 **Lenguaje de Expresión**, 6 **Programación Estructurada**, and B **Bibliografía**. The right column contains a list of sub-topics: **Problemas del Mundo Real y Modelos**, **Etapas de los Problemas**, **Descomposición del Problema**, **Documentación**, and **Práctica**. At the bottom center, there is a button labeled 'Salir'.

INDICE GENERAL		
1	Conceptos Generales	Problemas del Mundo Real y Modelos
2	Conceptos de Lógica	Etapas de los Problemas
3	Razonamiento Lógico	Descomposición del Problema
4	Terminología de Problemas	Documentación
5	Lenguaje de Expresión	Práctica
6	Programación Estructurada	
B	Bibliografía	
		Salir

Esta figura nos muestra el Indice General, en ella se enumeran cada uno de los capítulos y sus contenidos.

Las posibilidades que brinda el Índice General son :

- Presionando los *números* que se encuentran ubicados en el margen izquierdo de la página se podrá tener acceso al Índice del capítulo correspondiente al número seleccionado.
- Presionando la *B* ubicada debajo de los números se accede a la página correspondiente a la bibliografía.
- A la derecha de los números se encuentran *los títulos de cada uno de los seis capítulos* que componen el libro. Quedará subrayado aquél título que haya sido seleccionado por última vez.
- A la derecha de los títulos se pueden observar los *temas que contiene el capítulo* que está seleccionado. Presionando un tema particular se accede a la primer página correspondiente a dicho tema.
- El botón *Salir* que se encuentra en la parte inferior de la página permite salir del libro.

Indice de Capítulo

The screenshot shows a software interface for a chapter index. At the top, a dark bar contains the text "Indice del Capítulo". Below this, the interface is divided into two main sections: "Objetivo" on the left and "Indice" on the right. The "Objetivo" section features a scroll graphic with the following text: "Aplicar programación modular para la descomposición Top-Down de un problema concreto. Implementar el algoritmo asociado al problema y describir en detalle la ejecución del mismo." The "Indice" section contains a list of five items, each preceded by a checkmark: "Introducción", "Programación Modular", "Análisis de Problemas", "Conclusiones", and "Practica". At the bottom left, a dark bar displays "6 Programación". At the bottom right, there are navigation controls consisting of two rows of buttons. The first row contains buttons labeled "B", "I", a circular arrow, a left-pointing triangle, and a right-pointing triangle. The second row contains buttons labeled "S", "1", "2", "3", "4", "5", and "6".

Esta figura nos muestra el Índice de un Capítulo, en ella podemos ver el objetivo del mismo y los temas que lo componen.

En este punto debemos aclarar que una vez recorridas todas las páginas que componen un tema aparece junto a él (en el Índice del capítulo), un tilde para reflejar este hecho.

Las posibilidades que brinda el Índice del Capítulo son :

- Presionando sobre cada uno de los *temas que integran el capítulo*, se puede acceder a la primer página correspondiente a dicho tema.

En la parte inferior de la página se encuentran los siguientes botones :

- Presionando el botón indicado por la letra *B* vamos a la página correspondiente a la bibliografía.
- Presionando el botón indicado por la letra *I* vamos al Índice General.
- Presionando el *botón que está a la derecha de la I* vamos a la página donde estuvimos anteriormente, sin ser necesariamente la página previa a la actual en el ordenamiento de páginas que tiene el libro.
- Presionando la *flecha que apunta a la izquierda* vamos a la página anterior.
- Presionando la *flecha que apunta a la derecha* vamos a la página siguiente.
- Presionando el botón que contiene una *S*, salimos del libro.
- Presionando uno de los *números* que van del uno al seis vamos al Índice del capítulo del número seleccionado.

Página Tipo

capítulo

8-20

página

ok

Etapas de los Problemas

Origen de los Problemas Científicos

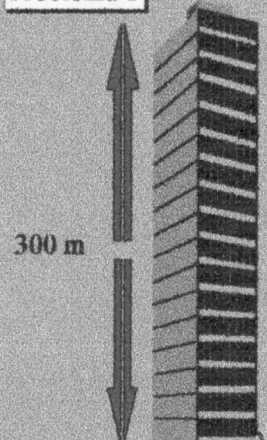
Uno de los problemas clásicos que hay que resolver en los cursos elementales de física es el de la "caída de un cuerpo"

A partir de la ley de Newton y de algunos experimentos encontramos que la fuerza de la gravedad por unidad de masa produce una aceleración de 9.8 metros por segundo cuadrado. Por lo tanto, la distancia d que recorre un cuerpo en un tiempo t con aceleración constante igual a a es : $d = a * t^2 / 2$

Consideremos el problema específico de dejar caer una piedra desde lo alto de una torre de 300 metros de altura. En este caso el tiempo de caída es de:

$t = [600/9.8]^{1/2} = 7.82$ segundos

Problema 1



300 m

1
Conceptos

Esta figura nos muestra el diseño de una página del libro.

Las posibilidades que brinda una página cualquiera del libro son :

- En el ángulo inferior izquierdo está el *número y nombre del capítulo* en el que nos encontramos.
- En el ángulo superior derecho se encuentra el *nombre del tema* que estamos recorriendo.
- Los *números* que se encuentran en la parte superior izquierda de la página nos indican el número de la página en la que estamos situados y el total de páginas que contiene dicho capítulo.
- Los botones que se encuentran en la parte inferior de la página ya fueron detallados en el Índice del capítulo. La diferencia es que en lugar de una B que permitía acceder a la bibliografía, aparece una P. Presionando la P se le indica al lector mediante un mensaje que puede elegir acceder a una página de un capítulo determinado, el lector deberá clicar el mouse sobre el OK del mensaje y escribir el número de capítulo y de página deseados en los lugares indicados. Una vez realizado esto se debe clicar sobre el OK. Esta tecla permite agilizar la navegación dentro del libro, en particular cuando se trata de ir a una página a la que no se accede directamente mediante el Índice del capítulo. Si se escribe un número de capítulo o de página incorrectos aparece un mensaje que indica el error.

6. Convenciones utilizadas

Existen varios factores que influyen en la calidad del trabajo desarrollado, como el *tipo y tamaño de la letra*, *empleo del color*, etc.. Por este motivo hicimos un análisis del uso de los mismos, que detallamos a continuación :

- Utilizamos un *color* distinto para cada capítulo, de esta forma el alumno podrá identificar un capítulo por el color de su background.
- También juega un rol importante la *tonalidad*, en este caso se empleó para identificar los estados de un botón, aquellos que han sido activados al menos una vez tienen una tonalidad menor que los que nunca fueron activados. Este efecto se mantiene mientras no se cambie de página.
- Todos los *ejemplos* tienen un fondo gris, esto marca la diferencia con la parte teórica donde se desarrolla el tema que luego se ejemplifica.
- Incorporamos algunos *gráficos*, que además de cumplir el objetivo de mejorar la estética del trabajo, facilitan al alumno la tarea de encontrar un tema de su interés, ya que se establece inconscientemente una relación entre la ubicación del gráfico y el tema pesquisado.
- Utilizamos *títulos y subtítulos* en todas las páginas, para que el alumno sepa en todo momento en que lugar del libro se encuentra.
- Las *hotword* aparecen en todos los capítulos en color azul y con un tipo de letra distinto al resto. Cuando se clickea una hotword su color cambia al color del capítulo y cuando se abandona, la misma vuelve a tomar el color azul.

Si bien todas estas consideraciones parecen superfluas, sin valor, está probado que gran parte del éxito de un material de estudio computarizado depende de la amigabilidad de la interfase hombre-máquina. Un material puede ser completo y perfecto en su contenido teórico, pero si en su interfase es pobre o no conveniente probablemente dicho material fracasará en su cometido. Es importante que el diseñador tome conciencia de esto y que utilice el tiempo necesario para analizar y establecer las consideraciones a tener en cuenta en el diseño de la interfase.

Evaluación

Una vez que el libro estaba en condiciones de ser evaluado, se seleccionaron alumnos del curso de ingreso de la carrera de Informática correspondientes al año 1997, para realizar con ellos una primer prueba.

Para efectuar la selección, se invitó a los alumnos a realizar la prueba. Con los interesados se formaron cuatro grupos, con una frecuencia semanal cada uno.

Las pruebas en máquina fueron individuales. Se dejó a los alumnos que recorrieran el libro de la forma en la que ellos desearan, sin limitar su tiempo en el recorrido.

Se observó que los grupos terminaban de recorrer el libro en dos o tres visitas a la sala de máquinas.

El aspecto más importante desde el punto de vista educativo, que consiste en verificar que el alumno es capaz de aprender mediante la aplicación, arrojó resultado positivo.

El objetivo de esta prueba fue testear el producto para verificar que esté implementado correctamente, y además evaluar el buen uso del mismo por parte de los distintos grupos participantes. Es decir, ver si el alumno podía manejarse de manera eficiente dentro del libro, utilizando todas las herramientas que el mismo provee. Si podía acceder a una página que ya había visto con anterioridad, si podía acceder al capítulo deseado, y si interactuaba en forma correcta cuando el libro le pedía que ingresara alguna información.

La prueba realizada fue beneficiosa, porque nos permitió conocer cuáles eran las dificultades de los alumnos cuando recorrían el libro, permitiéndonos mejorar el producto. Los resultados obtenidos fueron los siguientes :

- Como el libro está desarrollado en forma secuencial, los alumnos debían apretar todos los botones o hotword o demás objetos clickeables para no perder parte de la información que tal vez fuera necesaria para entender un tema posterior . Lo que notamos en los alumnos fue que salteaban algunos botones u objetos clickeables pensando que no contenían más información que la visible. Fue por ello que incorporamos el efecto que el mouse cambie a la forma de una mano cuando está posicionado sobre un objeto desplegable.

- Se observó que se perdía mucho tiempo al querer acceder a una página intermedia a la que no se tuviera acceso por el Índice general o el Índice de un capítulo. Esto motivó que agregáramos el ícono de navegación simbolizado con la letra P, que nos permite desde cualquier página acceder a cualquier otra de un capítulo determinado. Con esto ganamos en velocidad, facilitando el acceso a los temas de interés.
- Se mencionó que el elevado costo del hardware requerido para correr la aplicación, provocaba que no todos los alumnos puedan hacer uso del producto en la PC de su propiedad. Cabe aclarar sin embargo, que el producto estará disponible para su uso en la sala de PC de la Facultad.
- En los distintos grupos, los alumnos que manejaban Windows, recorrían el libro con mucha más facilidad que aquellos que nunca lo habían usado.
- Algunos alumnos, mencionaron que la presencia de un ayudante resultaría de importancia para poder evacuar las dudas de contenido que pudieran surgir. Así también, como tener la posibilidad de imprimir el libro a fin de fijar conceptos.

Conclusiones

Finalizada la prueba, el tutorial fue modificado contemplando todas las observaciones de relevancia realizadas.

A nuestro entender, el producto final logrado, cumple con las expectativas iniciales que motivaron nuestro desarrollo. Así mismo, pensamos que resulta una herramienta de gran utilidad para el alumno ingresante, por el beneficio que pueda reportarle en función del tiempo que le dedique.

Sin lugar a dudas, este tutorial podría mejorarse agregando mayor interacción con el alumno, así mismo como evaluaciones parciales del contenido.

Como objeto de futuros desarrollos, se podría mencionar el curso de ingreso correspondiente al área de Matemáticas.

Apéndice

A. Manual del Usuario

1. Requerimientos

Para poder instalar el Curso de Ingreso en una computadora, ésta deberá contar con :

- ✓ Microsoft Windows 3.1 o Windows 95.
- ✓ Una computadora con un procesador 80486 o superior.
- ✓ Un mouse
- ✓ Drive CD Rom, plaqueta de sonido.
- ✓ Al menos 8 megabytes de memoria; se recomienda 12 o más megabytes.
- ✓ 85 megabytes de disco duro disponibles.
- ✓ Monitor color Super VGA, fuente pequeña.

2. Instalación del CD

Para instalar el CD se deberá acceder al drive correspondiente y ejecutar *SETUP*, esto mismo puede ser realizado desde el Administrador de Archivos de dos maneras, una mediante la opción *Ejecutar* indicando el path correspondiente, y la otra clickeando dos veces sobre el archivo *SETUP*.

Una vez finalizada la instalación, se crea un grupo en Windows llamado Ingreso, el cual contiene un único ícono de igual nombre. Presionando dicho ícono se accede a la aplicación.

3. Navegación dentro de la aplicación

La aplicación está desarrollada en forma secuencial, es decir el alumno para conocer un tema específico no tiene una entera libertad de movimiento, sino que debe recorrer paso a paso cada página y objeto clickeable que integre el tema de estudio. De todos modos, el usuario tiene la facilidad para moverse de una página a otra no consecutiva, indicando el capítulo y número de página al que desea acceder.

En una etapa posterior, cuando el alumno realice una revisión del contenido del libro, y esté interesado sólo en repasar determinados puntos de un tema, podrá aprovechar la facilidad de navegación no secuencial.

Los objetos a ser clickeados dentro del libro, son aquellos que contienen más información que la visible, es el caso de las hotwords, los botones y algunos campos de texto. Estos objetos pueden ser identificados fácilmente, pues cuando el mouse se posiciona sobre ellos el cursor cambia a la forma de una mano.

Objetos que contienen información

Botones

Demo

Problemas del mundo real

Hotword

modelo matemático

Gráfico



Iconos de Navegación

Retrocede a la página previa



Avanza a la página siguiente



Regresa a la página que llamó a la actual (back)



Accede a una página de cualquier capítulo



Accede al Índice de un capítulo

1 2 3 4 5 6

Accede al Índice General



Accede a la Bibliografía



Salir de la Aplicación

Salir



B. Glosario

[Too89], [TII96]

Aplicación

Es un programa de computadora que ejecuta tareas específicas. Toolbook es una aplicación que corre bajo Microsoft Windows. Una aplicación Toolbook consiste de uno o más libros diseñados para un propósito particular, tal como entrenamiento, información gerencial o entretenimiento.

Background

Un background es un diseño de página compartido por un grupo de páginas en un libro. Cada libro tiene al menos un background. Los objetos del background aparecen en la misma posición, estilo y tamaño en cada página que comparte ese background.

Botón

Es un objeto creado con la paleta de herramientas. Al presionar un botón se ejecuta el script correspondiente al mismo.

Campo Registro

Es un tipo de campo especial del background en el cual un usuario puede tipear texto diferente en cada página.

Click

Es la acción de presionar y liberar el botón del mouse mientras el puntero señala el objeto que se pretende acceder.

Clip

Es una referencia a un segmento de cualquier fuente media o archivo, incluyendo archivos de sonido, CD audio, archivos de video digital, animación, videodisc o videotape. Un clip incluye un punto de inicio, un punto de fin y un nombre. Sólo la referencia es almacenada en Toolbook .

Device driver

Es un programa que permite a Windows comunicarse con una pieza específica de hardware multimedia.

Field

Es un objeto que contiene texto. Se puede leer, escribir, formatear texto y crear hotwords en un campo a nivel autor, pero el texto de campos del background y de campos bloqueados no pueden ser cambiados a nivel lector. Cada campo puede tener un script.

Foreground

Un foreground es la parte de una página que contiene los objetos únicos de esa página. Un objeto del foreground se puede copiar y pegar en otra página, pero a diferencia del background, los objetos del foreground no aparecen automáticamente en las páginas sucesivas.

Handler

Es una serie de sentencias OpenScript que describen la respuesta a un evento particular tal como el clickeo de un objeto. Un script puede contener más de un handler, cada uno respondiendo a eventos diferentes.

Hotword

Es una pieza de texto diseñada especialmente en un campo que tiene un script y un ID similar a otros objetos.

Id

Es un número de identificación único que Toolbook asigna a cada objeto cuando éste es creado.

Layer

Es el orden relativo de un objeto con respecto a todos los objetos ubicados en el mismo foreground o background. Se puede cambiar el layer de un objeto, con ello se estará cambiando la forma en que se superponen los objetos en la página.

Libro

Es una colección de páginas de Toolbook almacenadas juntas como un archivo DOS, usualmente basados en un tema o tarea.

Mensaje

- Es una comunicación que Toolbook envía a un objeto para indicar que un evento ha ocurrido.
- Un aviso o información presentada por Microsoft Windows o Toolbook.

Nivel Autor

Es el nivel de trabajo en Toolbook que provee herramientas y comandos para crear y modificar objetos, páginas, backgrounds y libros.

Nivel Lector

Es uno de los dos niveles de trabajo en Toolbook. El nivel lector provee comandos para recorrer y agregar páginas a un libro; para tipear, editar y formatear texto en campos y para imprimir.

Objeto

- En una página de Toolbook, un campo, un campo registro, un botón, un objeto dibujado, un objeto pintado o un grupo. Cualquier objeto puede tener un script, además se puede seleccionar un objeto para cambiar su tamaño, localización y estilo.
- Cualquier ítem en Toolbook que puede tener un script y otras propiedades, éstos incluyen páginas, backgrounds, libros, hotwords y demás objetos de una página.

Objetos del Background

Son los campos, botones, objetos dibujados, objetos pintados o grupos ubicados sobre el background. Dichos objetos aparecen en cada página que comparte dicho background. Un campo registro es un tipo especial de campo del background.

Objetos del Foreground

Es un campo, botón, objeto dibujado, objeto pintado o grupo que existe sobre el foreground de una página. Cada objeto del foreground tiene un único ID para la página en la cual se encuentra.

OpenScript

Es el lenguaje de programación de Toolbook.

Página

Es la unidad básica de un libro, formada por una combinación de un foreground y un background.

Paleta

Es una colección de íconos, colores o estilos que permiten elegir una herramienta, un color, el estilo de una línea o la forma de un polígono.

Paletas de colores

Son las paletas mediante las cuales el autor puede asignar colores a los objetos. Son dos : “Color Tray” y “Spectrum Palette”.

Password

Es una palabra que el autor asigna para limitar el acceso al libro. Un libro puede tener tres diferentes password, una para cambiar a nivel autor, una para salvar el libro bajo el mismo nombre, y otra para abrir el libro.

Script

Es una serie de sentencias OpenScript divididas en partes llamadas handlers que describen el comportamiento de un objeto en respuesta a un evento específico. Cuando un evento ocurre , tal como que un lector clickee un objeto, Toolbook corre el handler del script del objeto para ese evento. Cualquier objeto puede tener un script, incluyendo un libro, página, background, botón, campo, objetos dibujados, objetos pintados, grupos y hotword.

Script recorder

El “script recorder “ de Toolbook brinda la posibilidad de almacenar casi cualquier acción que se pueda ejecutar en Toolbook. El script recorder traduce las acciones en Toolbook en sentencias OpenScript.

Scroll

Sirve para mover la pantalla hacia arriba y hacia abajo según se desee, clickeando el mouse sobre las teclas de dirección.

Stage

Es un objeto en el cual se puede ejecutar animación, video digital, video desde el videodisc o videotape, o imágenes fijas.

Style Palette

Son paletas que sirven para que el autor pueda elegir la apariencia de un objeto dibujado. Son tres, la “Line palette”, la “Pattern palette”, y la “Polygon palette”.

Tool Palette

Es una colección de íconos que Toolbook despliega a nivel autor para crear y trabajar con objetos.



Bibliografía

- [Agu93] Aguilar, Luis “Fundamentos de Programación, Mc Graw Hill 1993.
- [Ber95] Bertone, De Giusti, Gorga, Madoz, “Una propuesta de curso interactivo multimedial para el ingreso a Informática en la UNLP”, II International Congress on Information Engineering Proceedings.
- [Fol90] J. Foley, A. van Dam, S. Feiner, J. Hughes, “Computer Graphics”, Addison-Wesley Publishing Comp., 1990.
- [Gar86] Garland, “Introduction to the computer systems, with application in Pascal”, Addison Wesley, 1986.
- [Gev87] Willian Gevarter, “Máquinas inteligentes”, Ediciones Diaz Santos 1987.
- [Gon92] Rafael C. Gonzáles, Richard E. Woods, “Digital Image Processing”, Addison-Wesley Publishing Comp., 1992.
- [Hil82] Suppes P., Hill S., “Introducción a la lógica matemática”, Editorial Reverté S.A., 1982.
- [IEE] Colección de “Computer Graphics and Applications”, IEEE.
- [Ing95] Lanzarini, De Giusti, Bertone, Naiouf, “Guía de Análisis y Expresión de Problemas”, Curso de Ingreso 1995-1996, Fac. de Ciencias Exactas.
- [Oub65] Oubiña, Lia, “Introducción a la teoría de Conjuntos”, Editorial Universitaria de Buenos Aires, 1965.
- [Sha92] Philip Shaddock, “Creaciones Multimedia”, Editorial Anaya, 1992.
- [Ter67] Terrell, Baker, “Exercises in Logic”, Holt, Rinehart and Winston INC. 1967.
- [Ope89] “Using Open Script”, An Introduction and Reference to the Open Script Language. Asymetrix Corporation 1989.
- [Too89] “Using Toolbook”, A Guide to Bilding and Working with Books. Asymetrix Corporation 1989.
- [TII96] “Toolbook II 5.0”, Multimedia Guide. Asymetrix Corporation 1996.

