



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA DE DISEÑO GRÁFICO

**“DESARROLLO DE UN CURSO VIRTUAL DE ACTION SCRIPT 3 COMO
APOYO DEL PROCESO ENSEÑANZA APRENDIZAJE PARA LOS
ESTUDIANTES DE DISEÑO GRÁFICO EDG ESPOCH”**

TESIS DE GRADO

Previa la obtención del título de:
INGENIERA EN DISEÑO GRÁFICO

Presentado por:

Paulina Cumandá Naranjo Armijo.

RIOBAMBA – ECUADOR

-2013-

Agradezco a Dios por cada día darme la oportunidad de ver un nuevo día y permitirme disfrutarlo, a mi familia por esa fuerza y apoyo constante, mi madre por estar a mi lado en cada momento de lucha, mi padre por enseñarme que cada día es un comienzo. A mis maestros y tutores por su invaluable aporte durante estos años. A mis amigos y compañeros por su valiosa presencia en cada instante de mi vida.

Y de manera general a todas las personas las cuales sin su enriquecedor apoyo el presente trabajo no hubiese sido posible.

*Dedico la presente tesis a mi madre
por ser ella quien con su sabiduría y
paciencia ha estado a mí lado en
todos los momentos difíciles,
apoyándome con sus sabios
consejos, y junto a ella a la memoria
de mi abuelita Teresa.*

FIRMAS DE RESPONSABLES Y NOTA

NOMBRE

FIRMA

FECHA

Ing. Iván Menes C.

**DECÁNO DE LA FACULTAD
DE INFORMÁTICA Y ELECTRÓNICA**

Arq. Ximena Idrobo

**DIRECTORA DE LA ESCUELA
DE DISEÑO GRÁFICO**

Lcdo. Luis Viñan

DIRECTOR DE TESIS

Ing. Miguel Duque

MIEMBRO DEL TRIBUNAL

Lcdo. Carlos Rodríguez

**DIRECTOR DEL CENTRO
DE DOCUMENTACIÓN**

NOTA

DERECHOS DE AUTORÍA

“Yo, Paulina Naranjo Armijo, soy responsable de las ideas, doctrinas y resultados expuestos en la presente Tesis, y el patrimonio intelectual de la misma pertenece a la Escuela Superior Politécnica de Chimborazo”.

Paulina Cumandá Naranjo Armijo

INDICE DE ABREVIATURAS

AS3.0: ActionScript 3.0

AVI: (Audio Video Interleave)

CEV: Centro de Educación Virtual

CIV: Centro de Interacción Virtual

CMYK: Cyan, Magenta, Yellow, y black

DEL: Departamento de Educación en Línea

EVA: Entornos Virtuales de Aprendizaje

EXE: Executable

GIF: Graphics Interchange Format

IE: Institución Educativa

JPG: Joint Photographic Experts Group

POO: Programación Orientada a Objetos

RGB: Red, Green, Blue (Rojo, verde, azul)

SWF: Small Web Format (archivos flash)

SENESCYT: Secretaría nacional de educación superior, ciencia, tecnología e Innovación

TIC: Tecnologías de la información y la comunicación.

TIFF: Tagget Image File Format

WAV: Formato de audio digital (Waveform Audio Format)

ÍNDICE GENERAL

PORTADA

AGRADECIMIENTO

DEDICATORIA

FIRMAS Y CALIFICACIÓN

FIRMA DE RESPONSABILIDAD

INDICE DE ABREVIATURAS

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN.

CAPÍTULO I	
1. MARCO REFERENCIAL.....	19
1.1. ANTECEDENTES.....	19
1.2. JUSTIFICACIÓN DEL PROYECTO DE TESIS.....	22
1.3. OBJETIVOS.....	26
OBJETIVOS GENERALES:	26
OBJETIVOS ESPECÍFICOS:.....	26
1.4. HIPÓTESIS.....	26

CAPÍTULO II	
2. METODOLOGÍA.....	27
2.1. INVESTIGACIÓN.....	27
2.2. OBJETIVO DE LA INVESTIGACIÓN.....	28
2.3. POBLACIÓN DE ESTUDIO	28
2.4. SEGMENTACIÓN.....	29
2.5. OBTENCIÓN DE LA INFORMACIÓN	29
2.6. ANÁLISIS E INTERPRETACIÓN DE LA INFORMACIÓN	30
CAPÍTULO III	
3. MARCO TEÓRICO.....	36
3.1. DEFINICIÓN MULTIMEDIA	36
3.1.1. INTERACTIVIDAD.....	37
3.1.2. TIPOS DE ARCHIVOS MULTIMEDIA	37
3.1.3. CLASIFICACIÓN DE LOS MULTIMEDIA	39
3.1.4. CARACTERÍSTICAS APLICACIONES MULTIMEDIA.....	40
3.1.5. SOFTWARE DE CREACIÓN MULTIMEDIA.....	41
3.1.6. HERRAMIENTAS DE DESARROLLO MULTIMEDIA.....	42
3.1.7. PRODUCCIÓN PARA MULTIMEDIA.	43
3.2. DISEÑO GRÁFICO PARA MULTIMEDIA	44
3.2.1. IDENTIDAD PARA MULTIMEDIA.....	45
3.2.2. TIPOGRAFÍA.....	45
3.2.3. EL COLOR.....	47
3.2.4. DIAGRAMACIÓN Y RETÍCULA.....	51
3.2.5. ERGONOMÍA EN MULTIMEDIA.	52
3.2.6. USABILIDAD EN MULTIMEDIA.	54
3.3. EDUCACIÓN MULTIMEDIA	56
3.3.1. EDUCACIÓN VIRTUAL	56
3.3.2. MULTIMEDIA EN LA EDUCACIÓN VIRTUAL.....	59
3.3.3. RECURSOS MULTIMEDIA PARA EL APRENDIZAJE.....	60
3.3.4. BENEFICIOS DE LA EDUCACIÓN VIRTUAL	62
3.3.5. PROBLEMAS DE LA EDUCACIÓN VIRTUAL.....	63

3.3.6.	LA EDUCACIÓN VIRTUAL EN EL MUNDO	64
3.3.7.	LA EDUCACIÓN VIRTUAL EN ECUADOR	66
3.3.8.	LA EDUCACIÓN VIRTUAL EN LA ESPOCH	68
3.4.	METODOS DE ENSEÑANZA APRENDIZAJE VIRTUALES.....	71
3.4.1.	E-LEARNING Y B-LEARNING.....	71
3.4.2.	TEORÍAS DE APRENDIZAJE EN LOS ENTORNOS VIRTUALES .	72
3.5.	PROGRAMACION ORIENTADA A OBJETOS Y ACTION SCRIPT 3.0	77
3.5.1.	PROGRAMACIÓN ORIENTADA A OBJETOS.....	77
3.5.2.	FUNDAMENTOS DE LA POO.....	78
3.5.3.	CONCEPTOS DE LA POO.....	82
3.5.4.	EL LENGUAJE DE PROGRAMACIÓN ACTION SCRIPT 3.0	86
3.5.5.	CREAR Y USAR VARIABLES	95
3.5.6.	USAR CONDICIONALES	99
3.5.7.	USAR LOOPS (CICLOS).....	102
3.5.8.	CREAR Y USAR FUNCIONES.....	105
3.5.9.	GESTIÓN DE EVENTOS.....	110
3.5.10.	CLASES Y OBJETOS	115
3.5.11.	OBJETOS DE VISUALIZACIÓN Y CLIPS DE PELÍCULA.....	119
3.5.12.	CARGAR ELEMENTOS EXTERNOS	129
CAPÍTULO IV.....		
4.	PROPUESTA	130
4.1.	RECOPIACIÓN DE LA INFORMACIÓN	130
4.2.	ANÁLISIS Y ORGANIZACIÓN DE LA INFORMACIÓN.	137
4.3.	DISEÑO DEL CURSO VIRTUAL MULTIMEDIA.	139
4.3.1.	REQUERIMIENTOS BÁSICOS	139
4.3.2.	CONTENIDOS MULTIMEDIA.....	140
4.3.3.	CONTENIDOS HIPERMEDIA INFORMATIVO PDF.....	141
4.3.4.	MAPA DE NAVEGACIÓN.....	143
4.3.5.	DISEÑO DEL ICONO O LOGO	144
4.3.6.	DISEÑO DEL INTERFAZ.....	148

4.3.7. DISEÑO DEL PDF INFORMATIVO	152
4.3.8. PRUEBA DE USABILIDAD	154
CAPÍTULO V	
5. VALIDACIÓN.....	157
5.1. ANÁLISIS DE ENCUESTAS	157
CONCLUSIONES	
RECOMENDACIONES	
RESUMEN	
ABSTRACT	
GLOSARIO	
ANEXOS	
BIBLIOGRAFÍA	

ÍNDICE DE FIGURAS

FIGURA II. 1: Genero del segmento	30
FIGURA II. 2: Autoimagen como estudiante	31
FIGURA II. 3 Autoimagen como desarrolladores de aplicaciones.....	31
FIGURA II. 4 Campos de interés.....	32
FIGURA II. 5: Beneficios que espera	33
FIGURA II. 6: Nivel de utilidad del as3.0.....	33
FIGURA II. 7: Estudiantes que han usado adobe flash.....	34
FIGURA III. 8: Navegación lineal	39
FIGURA III. 9: Navegación reticular.....	39
FIGURA III. 10: Navegación jerarquizada.....	40
FIGURA III. 11: Herramientas adobe para desarrollo multimedia	42
FIGURA III. 12: Modelos de color CMYK y RGB.....	47
FIGURA III. 13: Ejemplo saturación	48
FIGURA III. 14: Ejemplo brillo	48
FIGURA III. 15: Ejemplo contraste	49
FIGURA III. 16: Tipos de contrastes	49
FIGURA III. 17: Modelo de aprendizaje clásico centrado en el profesor.....	56
FIGURA III. 18: Modelo de aprendizaje virtual centrado en el estudiante.....	57

FIGURA III. 19: Uso del internet en el mundo	54
FIGURA III. 20: Uso del internet en Ecuador por sexo.....	67
FIGURA III. 21: Uso de internet en el país	67
FIGURA III. 22: Metodología y esquema de desarrollo.....	65
FIGURA III. 23: Usabilidad de la plataforma virtual ESPOCH.....	70
FIGURA III. 24: Teoría Gestalt	73
FIGURA III. 25: Ejemplo de jerarquía.....	82
FIGURA III. 26: Ejemplo de clase	85
FIGURA III. 27: Interfaz flash CS6	88
FIGURA III. 28: Paneles de flash cs6.....	85
FIGURA III. 29: Nuevo Archivo de Tipo Actionscript.....	86
FIGURA III. 30: Documento de tipo ActionScript.....	90
FIGURA III. 31: manejo de código ActionScript3.0 en el panel de acciones ...	91
FIGURA III. 32: Ejemplo estructura if	100
FIGURA III. 33: Ejemplo estructura If...else,.....	101
FIGURA III. 34: Ejemplo estructura switch	102
FIGURA III. 35: impesión de 5 números	102
FIGURA III. 36: Ejemplo estructurado while.....	104
FIGURA III. 37: Ejemplo estructura do-while	104
FIGURA III. 38: Ejemplo estructura for.....	101
FIGURA III. 39: Ejemplo funciones personalizadas.....	102
FIGURA III. 40: Ejemplo funciones entrega de resultados.....	105
FIGURA III. 41: Tipos de Eventos.....	110
FIGURA III. 42: Vinculación de clases a la película	117

FIGURA III. 43: Convertir en símbolo.....	118
FIGURA III. 44: Jerarquía de clases de visualización	120
FIGURA III. 45: Ventana de propiedades de una clase	128
FIGURA IV. 46: Conocimientos previos - uso as3.0	131
FIGURA IV. 47: Conocimientos previos - fundamentos de la POO.....	128
FIGURA IV. 48: Conocimientos previos - funciones.....	128
FIGURA IV. 49: Conocimientos previos - variables.....	129
FIGURA IV. 50: Conocimientos previos – eventos.....	133
FIGURA IV. 51: Conocimientos Previos – Ciclos y Loops	134
FIGURA IV. 52: Conocimientos previos – errores y aciertos por tarea, ejercicio práctico en el uso del software adobe flash	135
FIGURA IV. 53: Conocimientos previos – errores y aciertos totales, ejercicio práctico en el uso del software adobe flash	135
FIGURA IV. 54: Conocimientos previos – errores y aciertos por parámetro, ejercicio práctico programación en AS3.0.....	136
FIGURA IV. 55: Conocimientos previos – errores y aciertos totales, ejercicio práctico programación en as3.....	137
FIGURA IV. 56: Mapa de navegación del curso virtual de AS3.0	143
FIGURA IV. 57: Posible icono del multimedia opción1.....	141
FIGURA IV. 58: Posible icono del multimedia opción2.....	142
FIGURA IV. 59: Helvetica 55 Roman.....	143
FIGURA IV. 60: Helvetica27-CondensedUltraLight.....	147
FIGURA IV. 61: Retícula del logo.....	148
FIGURA IV. 62: retícula Pantalla de bienvenida	149

FIGURA IV. 63: Pantalla de bienvenida curso virtual.....	149
FIGURA IV. 64: Distribución de las areas del multimedia	150
FIGURA IV. 65: retícula pantalla principal.....	150
FIGURA IV. 66: Pantalla principal curso virtual.....	151
FIGURA IV. 67: distribución del menú principal	151
FIGURA IV. 68: organización del documento pdf.....	153
FIGURA V. 69: Resultados de la aplicación- uso as3.0	158
FIGURA V. 70: Resultados de la aplicación - fundamentos de la POO.....	155
FIGURA V. 71: Resultados de la aplicación - FUNciones.....	156
FIGURA V. 72: Resultados de la aplicación - Variables.....	157
FIGURA V. 73: Resultados de la aplicación – Eventos.....	158
FIGURA V. 74: Resultados de la aplicación – ciclos y loops.....	158
FIGURA V. 75: Resultados de la aplicación – errores y aciertos por tarea...	159
FIGURA V. 76: Resultados de la aplicación – Errores y aciertos totales, ejercicio práctico programación en as3.....	163
FIGURA V. 77: Resultados de la aplicación – errores y aciertos por parámetro.....	159
FIGURA V. 18: Resultados de la aplicación – errores y aciertos totales.....	160
FIGURA V. 79: Resultados previo la aplicación del curso virtual.....	160
FIGURA V.80: Resultados generales posteriores la aplicación del curso virtual.....	161

ÍNDICE DE TABLAS

TABLA II.I Segmentación.....	35
TABLA III.II: Medidas de retícula según el monitor	52
TABLA III.III: Aspectos para el diseño de material multimedia respecto a la usabilidad.....	55
TABLA III.IV: Usuarios de internet en el mundo.....	65
TABLA III.V: Comparativa entre el modelo presencial y el modelo virtual-presencial de aprendizaje	72
TABLA III.VI: Ejemplo Diferentes modelos de abstracción	79
TABLA III.VII: Ejemplo de encapsulación.....	80
TABLA III.IIX: Ejemplo de modularidad.....	81
TABLA III.IX: Ejemplo de objeto.....	83
TABLA III. X: UI Components (componentes de interfáz)	92
TABLA III. XI: Tipos de datos sencillos o fundamentales	93
TABLA III. XII: Tipos de datos complejos.....	94
TABLA III. XIII: Operadores	95
TABLA III. XIV: variables.....	95

TABLA III. XV: Tipos de datos de valores simples con sus valores predeterminados	97
TABLA III. XVI: Tipos de datos de valor complejo.....	97
TABLA III. XVII: Descripción de URL absoluta y relativa	129
TABLA III. XVIII: Conocimientos previos de los estudiantes	138
TABLA IV. IXX: comparación de iconografías identificativas	145
TABLA IV. XX: código cromático.....	146
TABLA IV. XXI: Resultado de usabilidad	154
TABLA IV. XXII: Resultado de contenidos	155
TABLA IV. XXIII: Resultado de usabilidad	156

INTRODUCCIÓN.

La incorporación de tecnologías al ámbito educativo con el uso combinado de métodos pedagógicos y materiales de auto-aprendizaje, posibilita procesos educativos y comunicacionales que implican el acercamiento entre los agentes involucrados en la enseñanza-aprendizaje.

El desarrollo del producto multimedia propuesto pretende apoyar el proceso de enseñanza - aprendizaje de la programación orientada a objetos precisamente del lenguaje Action Script 3.0 para los estudiantes de la Escuela de Diseño Gráfico de la Facultad de Informática y Electrónica perteneciente a la Escuela Superior Politécnica de Chimborazo, mediante la integración de videos y guías informativas, con el propósito de mejorar sus conocimientos y capacidades para el desarrollo de aplicaciones multimedia en el software Adobe Flash con el uso del ActionScript 3.0 para esta investigación se ha utilizado la metodología inductiva-deductiva, con el fin de indagar en las características y conocimientos previos de los estudiantes de la escuela de diseño gráfico.

Para este proyecto se han estudiado de las características demográficas y psicográficas del grupo de estudio, las definiciones sobre multimedia y diseño

gráfico y el análisis de la educación virtual, beneficios y problemas que tienen tanto estudiantes como profesores al utilizar tecnologías multimedia como complemento de sus actividades académicas. Siendo el índice de resultado favorable debido a que el grupo de estudio pudo ampliar sus conocimientos y capacidades considerablemente para el desarrollo de aplicaciones multimedia en flash con el uso de AS3.0.

CAPÍTULO I

1. MARCO REFERENCIAL

1.1. ANTECEDENTES

La Escuela Superior Politécnica de Chimborazo es una institución ecuatoriana con sede en la ciudad de Riobamba, provincia de Chimborazo, creada el 18 de Abril de 1969. En el mes de Septiembre de 1998 mediante aprobación de Concejo Directivo se crea un programa temporal de Licenciatura en Diseño Gráfico, programa que entra en funcionamiento en Octubre de 1998. A partir de Julio del 2006 se oferta la carrera de Ingeniería en Diseño Gráfico aprobada mediante Resolución No. 239, Actualmente el SENECYT clasifica a la carrera

dentro del área de conocimiento de Humanidades Y Artes, Acogiéndose a La Clasificación Internacional Normalizada de la Educación, para que desde Marzo del 2013 se oferte la carrera “Diseño Gráfico” más no como una ingeniería, para poder acogerse al proceso de acreditación, incluyendo varios cambios en cuanto a su malla curricular, obteniendo en sexto semestre la materia de Programación II, misma que comprende contenidos sobre la Programación Orientada a Objetos, siendo esta la base para la comprensión de cualquier lenguaje de programación que se fundamente en sus paradigmas, como el lenguaje Action Script3.0.

Adobe Creative Suite es la paquetería más importante y con más herramientas en el área de la creatividad y el diseño, sus bondades son múltiples y cada una de ellas puede beneficiar y darle mayor creatividad, diseño, innovación y calidad a cada uno de los trabajos que un diseñador realice.

Flash es el software de creación más avanzado para crear animación interactiva para el Web o también logotipos animados, controles de navegación de sitios Web, animaciones de gran formato o sitios Web completos de Flash, la capacidad y flexibilidad de Flash es el medio ideal para desarrollar la creatividad, que mediante el manejo de código del lenguaje ActionScript 3.0 amplía de manera significativa las posibilidades del diseñador.

El lenguaje de programación Action Script es utilizado para crear scripts en Flash, gracias a la programación, el computador puede entender, probar y realizar las acciones u órdenes solicitadas por el usuario, al igual que con los humanos, la comunicación con el computador se realiza mediante un

vocabulario y una gramática ya establecidos, el lenguaje action script nos permite proporcionar mayor dinamismo a las animaciones creadas en Flash, algunas opciones son: Controlar la línea de tiempo, Controlar las propiedades de los objetos, Generación de contenido programado, Comunicación con el servidor.

Los archivos de Flash aparecen muy a menudo como animaciones en sitios web multimedia, en aplicaciones de Internet, anuncios en la Web, y actualmente debido en gran medida al aumento del ancho de banda, y la mayor potencia de las máquinas su uso es masivo siendo muchos sitios desarrollados en su totalidad con esta tecnología, entre ellos su uso en el desarrollo de multimedia educativos, la utilización de la tecnología multimedia en la educación contribuye a elevar la calidad del proceso de enseñanza-aprendizaje al posibilitar que el estudiante interactúe con un programa multimedia para complementar y reforzar su aprendizaje,

Los sistemas multimedia, cuya ventaja principal es la interactividad, se adaptan muy bien a este nuevo enfoque, ya que favorece el uso de la información en un contexto apropiado, de forma personalizada y la creación de un entorno virtual en el que el alumnado puede valorar instantáneamente el impacto de sus acciones, esta tecnología puede ser fácilmente alojada en el internet permitiendo el acceso fácil, y económico.

Desde la década de los 90, Internet se convirtió en una herramienta fundamental de información, comunicación e integración permitiendo al usuario economizar en tiempo y dinero, esta es una poderosa herramienta para ayudar a la difusión del conocimiento y educación.

1.2. JUSTIFICACIÓN DEL PROYECTO DE TESIS

El SENECYT clasifica a nuestra carrera dentro del área de conocimiento de Humanidades Y Artes, acogiéndose a la Clasificación Internacional Normalizada De La Educación, por este motivo la carrera debiera de sufrir varios cambios en su malla curricular, misma que implica que ya no sea la carrera de “Ingeniería en Diseño Gráfico” la que se oferte a los estudiantes como se habría establecido desde Julio del 2006 hasta Febrero del 2013, y a cambio sea la carrera de “Diseño gráfico” la que se oferte otorgando el título de “Diseñadores Gráficos”, entre estos cambios se reforma la malla curricular de la carrera con la implementación de varias materias.

En el plan analítico de sexto semestre de la asignatura de programación II se incluyen contenidos iniciales de programación de alto nivel, específicamente, programación orientada a objetos con el que posteriormente el estudiante tiene la posibilidad de comprender el lenguajes como el Action Script 3.0; parte esencial del adobe flash, necesario para los diseñadores por su utilidad en la animación y aplicaciones multimedia; con el diseño y aplicación de este curso virtual de AS3.0 además de ser un nexo complementario entre el diseño audiovisual y la programación orientada a objetos, pretende reducir considerablemente las posibles dificultades o dudas que los estudiantes de la escuela de Diseño Gráfico de la ESPOCH puedan presentar respecto a este lenguaje de programación orientada a objetos, complementando los

conocimientos adquiridos en clase, para lograr una comprensión y manejo adecuado sobre la programación Action Script 3.0.

Tomando en cuenta también que actualmente son pocas las asignaturas en las que se interactúa de forma virtual con el estudiante, existiendo una evidente falta de apoyo didáctico virtual extra para la enseñanza-aprendizaje, en observación a este inconveniente se propone implementar este tipo de cursos complementarios a los contenidos en clase mismos que serían desarrollados por el docente, entregando al estudiante material adicional a la vez que permite la posibilidad de ser proveídos mediante el internet, ya que este tiene varios potenciales educativos puesto que facilita el aprendizaje colaborativo mejora las vías de comunicación y satisface las necesidades de aprendizaje de los estudiantes.

La ausencia de una experiencia previa en el manejo de un lenguaje de alto nivel para muchos de los estudiantes de diseño gráfico puede presentar varias limitaciones en el desarrollo de multimedia basada en el entorno Adobe Flash CS6, creando como consecuencia aplicaciones básicas y sin interacciones, interacciones muy útiles hoy en día en el desarrollo multimedia como catálogos flash, juegos, publicidades, animaciones y otras publicaciones. Actualmente estas aplicaciones flash para ser adecuadamente desarrolladas necesitarían del apoyo de un programador, esto implicaría un costo de inversión adicional, además de otras dificultades que un diseñador con estos conocimientos y competencias puede evitar.

Para el desarrollo de estas competencias en los estudiantes de diseño gráfico de Sexto semestre de la ESPOCH se propone lograr una autoeducación

sustentable en el tema, misma que puede ser complementaria a los contenidos dictados en clase, mediante un curso virtual, que consiga que el estudiante desarrolle varios ejemplos de programación orientada a objetos dentro del lenguaje Action Script 3 con el asesoramiento del docente.

ActionScript 3.0 es un lenguaje orientado a objetos que permite ampliar las funcionalidades que Flash ofrece en sus paneles de diseño y además permitir la creación de películas o animaciones con altísimo contenido interactivo. Provee a Flash de un lenguaje que permite al diseñador añadir nuevos efectos o incluso construir la interfaz de usuario de una aplicación compleja, La versión 3.0 de ActionScript ha marcado un cambio significativo en este lenguaje, puesto que en esta versión prácticamente se ha decidido prescindir de los prototipos y se lo ha encaminado a ser un lenguaje orientado a objetos solamente a través de clases. El lenguaje ActionScript 3.0 puede simplificar el proceso de programación con un modelo más prolijo y ordenado gracias a la encapsulación junto la magnífica ventaja de la reutilización de código, entre las mayores ventajas que la programación orientada a objetos puede brindar están el control de la línea de tiempo o fotogramas, las propiedades de los objetos, la generación de contenido programado, además de una comunicación con el servidor relacionando el sistema con el mundo real; otra de las ventajas significativa de la programación orientada a objetos es la reutilización y extensión del código, permitiendo en muchos de los casos la creación de sistemas más complejos como programas visuales.

Se pretende que este curso pueda ser útil tanto fuera como dentro del aula, además de ser un apoyo para las aplicaciones independientes que los

estudiantes deseen desarrollar; consiguiendo así que este tipo de programación no represente más una limitante para el desarrollo óptimo de las aplicaciones flash multimedia. Ya que también que está en las manos del diseñador gráfico el desarrollar contenidos atractivos para que lleguen al target de forma efectiva y eficiente, teniendo el papel de maquetar los contenidos de los recursos virtuales multimedia de forma que su acceso sea ergonómico y sugestivo para el usuario, resolviendo los posibles problemas que muchas de las veces se presentan cuando la comunicación no es completa, a esto me refiero que no solo basta que existan los contenidos textuales sino también los contenidos gráficos y visuales, mismos que deben tener elocuencia con el tema para que el usuario se sienta en total interés del producto multimedia a él entregado.

1.3. OBJETIVOS

OBJETIVOS GENERALES:

Desarrollar un curso virtual de ActionScript 3.0 como apoyo del proceso enseñanza aprendizaje para los estudiantes de sexto semestre de la escuela de Diseño Gráfico EDG de la ESPOCH del periodo académico Marzo-Julio 2013

OBJETIVOS ESPECÍFICOS:

- Investigar las limitaciones de los estudiantes de la EDG, para el desarrollo de aplicaciones flash multimedia con lenguaje AS3.0
- Analizar los métodos de enseñanza-aprendizaje para multimedia y educación virtual.
- Estudiar los conceptos y paradigmas de la Programación Orientada a Objetos y el lenguaje de Programación AS3.0
- Determinar el sistema de navegación del curso virtual.
- Crear la interfaz gráfica del curso virtual.
- Validar el curso virtual desarrollado a través de su aplicación a los estudiantes de la EDG.

1.4. HIPÓTESIS

La aplicación del curso virtual de ActionScript 3.0 a los estudiantes de sexto semestre de la escuela de diseño gráfico, permitirá mejorar sus capacidades para el diseño de aplicaciones flash multimedia.

CAPÍTULO II

2. METODOLOGÍA

2.1. INVESTIGACIÓN

Para lograr obtener los mejores resultados en nuestra investigación nos hemos basado en la investigación inductiva-deductiva ya que indagamos en las características de nuestro objeto de investigación, es decir los estudiantes de sexto semestre de la escuela de diseño gráfico, proporcionándonos los datos para poder establecer los requerimientos del curso virtual, contenidos, esquema y diseño.

Para esto se utilizaron métodos cualitativos y cuantitativos como parte de la estrategia de recolección de información en una investigación de campo, dado en la relación con los estudiantes para observar el grado de aceptación del curso virtual en el aula de clases.

Como herramientas de investigación se definieron la encuesta y la observación, para lograr obtener los resultados tanto cuantitativos y cualitativos respectivamente.

2.2. OBJETIVO DE LA INVESTIGACIÓN

Determinar las variables demográficas, psicográficas y conductuales de los estudiantes de sexto semestre de la escuela de diseño gráfico de la ESPOCH.

2.3. POBLACIÓN DE ESTUDIO

Nuestra población de estudio se limita a los estudiantes de sexto semestre de la escuela de diseño gráfico de la ESPOCH, por lo que nuestra investigación se enfoca a una población finita, ya que está delimitada y conocemos el número de elementos que la integran.

Fuente de información: secundaria: secretaria de la escuela de diseño gráfico.

Periodo académico: Febrero-Julio 2013

Semestre: Sexto

Número de estudiantes matriculados: 42

2.4. SEGMENTACIÓN

Segmentación demográfica. En esta segmentación, el mercado está dividido en diferentes grupos en base a variables como edad, sexo, tamaño de la familia, estado civil, ocupación, raza, escolaridad, Ciudad.

Segmentación Psicográfica. Es una segmentación basada en los factores de interés para una persona, sus opiniones y actividades que conforman su estilo de vida.

Segmentación del comportamiento o Conductual. Consiste en dividir al segmento demográfico-psicográfico en grupos de personas con características conductuales similares.

Después de aplicar el método de la encuesta (ver anexo 1) para comprobar nuestro segmento han resultado los siguientes datos:

2.5. OBTENCIÓN DE LA INFORMACIÓN

Para obtener la información de nuestra investigación se decidió aplicar el método de la encuesta (ver anexo 1) para comprobar nuestro segmento.

2.6. ANÁLISIS E INTERPRETACIÓN DE LA INFORMACIÓN

Se aplicó el cuestionario a los 42 estudiantes de sexto semestre matriculados en la materia de programación II de donde resultaron los siguientes datos:

DEMOGRÁFICOS:

Como variable demográfica establecemos que la edad de los estudiantes de sexto semestre de la escuela de diseño gráfico de la ESPOCH fluctúa entre los 20 a 27 años, donde el 52% son hombres y el 48% mujeres

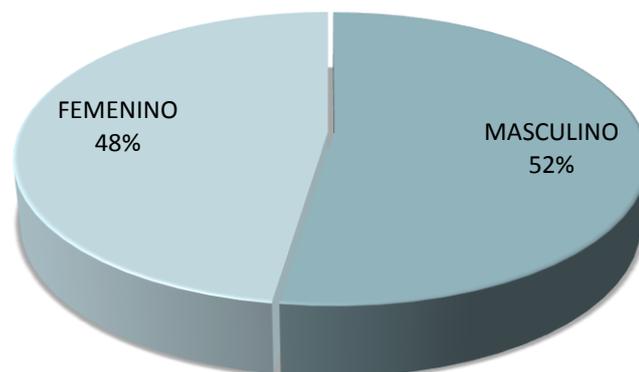


FIGURA II. 2: GENERO DEL SEGMENTO
Fuente: Elaboración Propia

PSICOGRÁFICOS:

Dentro de los parámetros de segmentación psicográfica encontramos que nuestro segmento tiene intereses sobre el estudio y el trabajo, donde el 55% tiene una autoimagen Buena como estudiantes, el 31% Muy buena, un 7% Excelente y también un 7% Regular.

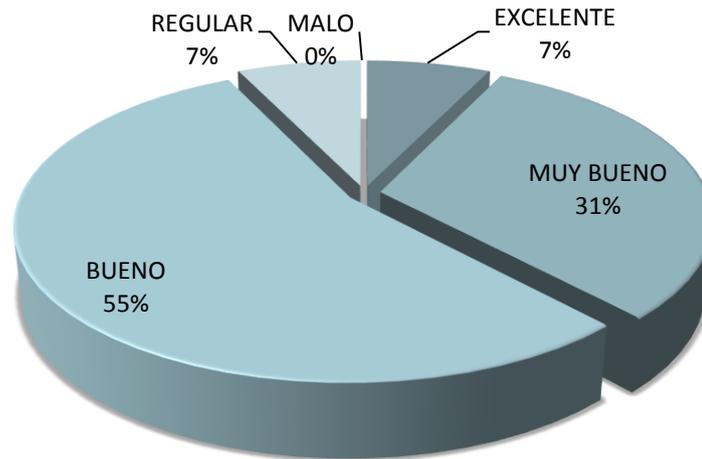


FIGURA II. 3: AUTOIMAGEN COMO ESTUDIANTE

Fuente: Elaboración Propia

En cuanto a su autoimagen como desarrolladores de aplicaciones varía ya que el 33% se considera Bueno, sin embargo el 67% no se siente conforme con sus capacidades para desarrollar aplicaciones multimedia.

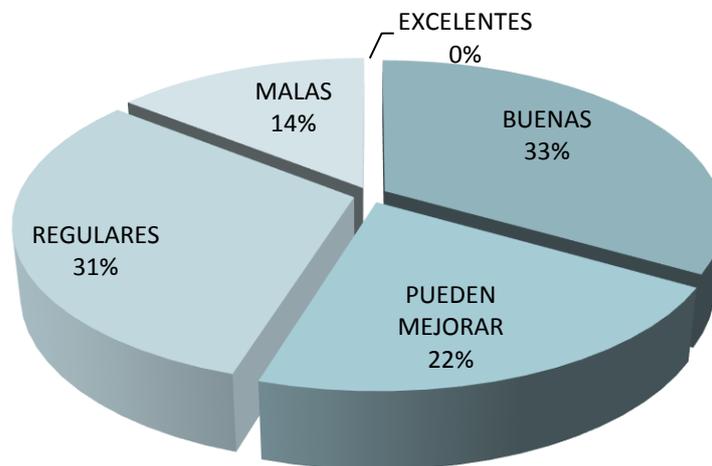


FIGURA II. 4 AUTOIMAGEN COMO DESARROLLADORES DE APLICACIONES

Fuente: Elaboración propia

CONDUCTUALES

En cuanto a intereses, nos hemos enfocado hacia el campo ocupacional al que desearían pertenecer, para esto se hizo que el estudiante de una valoración de 0 a 3 siendo 0 nada interesante y 3 el que mayor grado de interés representa, dando como resultado que el campo con mayor grado de interés es el Diseño de identidad corporativa, seguido por las Aplicaciones Multimedia y Presentaciones interactivas, dejando Medios impresos, Juegos flash y diseño web como las opciones con menor grado de interés.

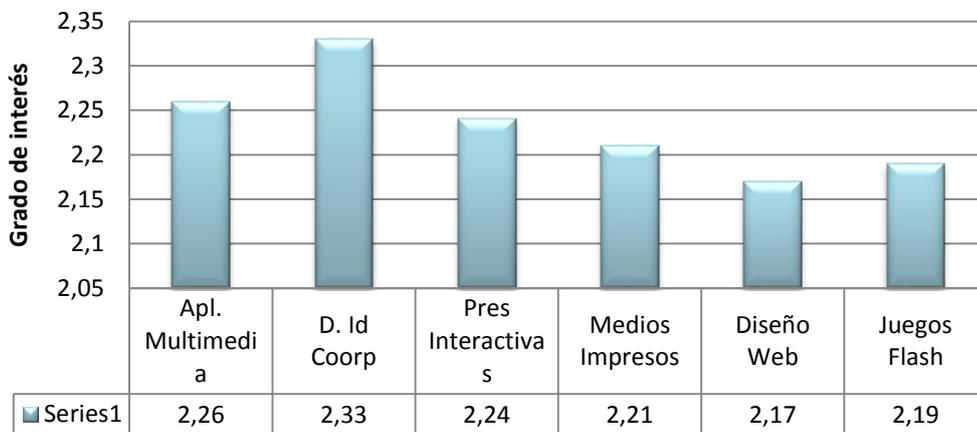


FIGURA II. 5 CAMPOS DE INTERÉS
Fuente: Elaboración propia

Dentro también de la segmentación conductual se ha encontrado los beneficios esperados, entre estos en primer lugar ejercicios prácticos con el 25%, Presentación atractiva con el 21%, con el 19% Facilidad de ingreso, 15% fuentes de información, autoevaluación con el 13% y en último lugar espacios de discusión con el 7%.

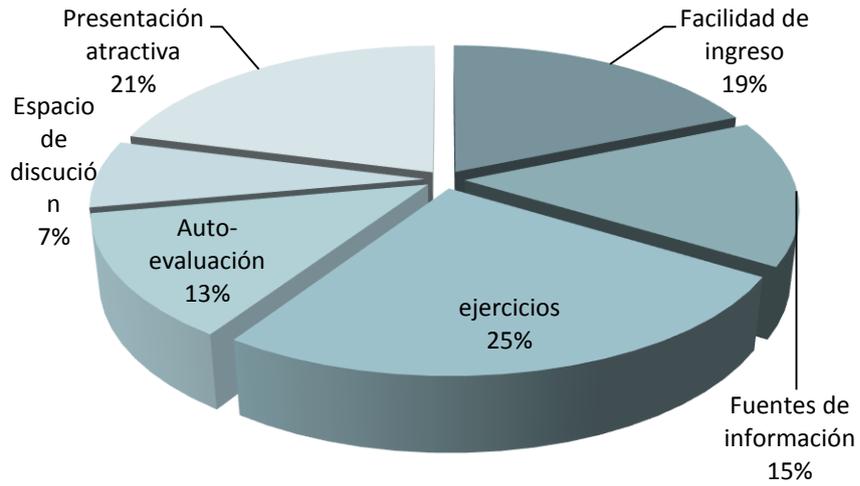


FIGURA II. 6: BENEFICIOS QUE ESPERA

Fuente: Elaboración propia

La opinión del segmento sobre el nivel de utilidad del lenguaje AS3.0 también fue de importancia relevante, surgiendo como resultado que el 50% desconoce de la existencia del lenguaje de POO, el 33% lo considera útil en algunas ocasiones y el 17% siempre.

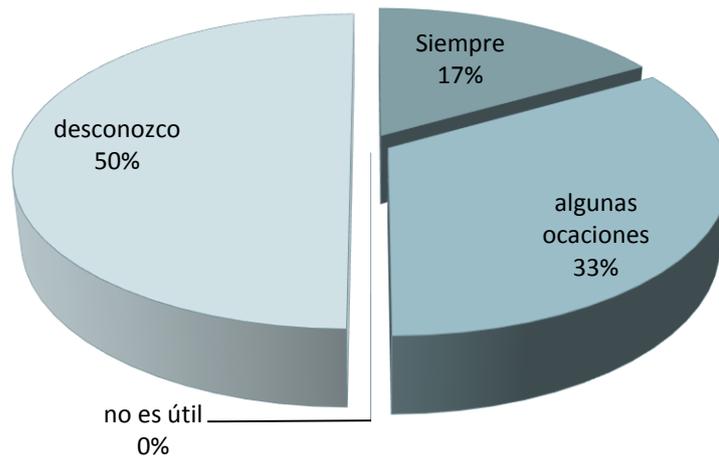


FIGURA II. 7: NIVEL DE UTILIDAD DEL AS3.0

Fuente: Elaboración propia

Para complementar el estudio de nuestro segmento dentro del campo conductual, preguntamos cuantos de nuestros estudiantes han usado el software Adobe Flash, donde resultó que el 36% si lo han usado y el 64% sin embargo no lo han usado nunca.

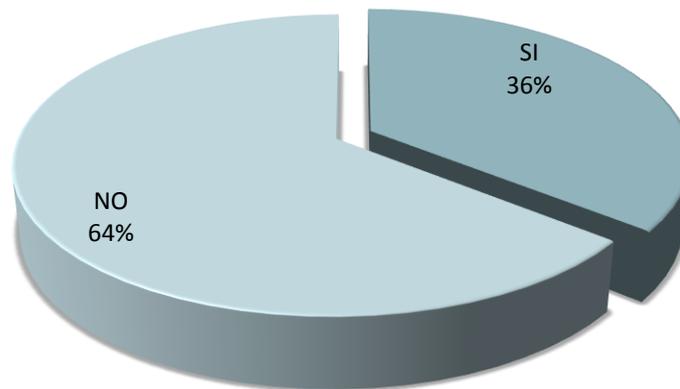


FIGURA II. 8: ESTUDIANTES QUE HAN USADO ADOBE FLASH

Fuente: Elaboración propia

CONCLUSIÓN DE LA INVESTIGACIÓN DEL SEGMENTO:

Con esto concluimos que nuestro segmento, está conformado por hombres y mujeres, estudiantes de entre 20 y 27 años, con una autoimagen buena, cuyas actividades son el estudio y el trabajo, y sus intereses se centran mayoritariamente en el diseño de identidad corporativa, aplicaciones multimedia, y presentaciones interactivas, y los beneficios esperados en el curso virtual, son los ejercicios prácticos, presentación clara y atractiva de los contenidos, facilidad de ingreso a la información y fuentes de información extra, siendo estudiantes que en su mayoría desconocen del lenguaje de programación AS3.0 y no tienen experiencia previa en el uso del software Adobe Flash.

TABLA II.I SEGMENTACIÓN

Segmentación	
Demográfica	Edad: Estudiantes entre los 20 a 27 años
	Género: 52% hombres y 48 % mujeres
Psicográficos	Autoimagen: 7% Excelente 31% Muy bueno 55% Buena 7% Regular 0% Mala
	Autoimagen como desarrolladores de aplicaciones: 0% Excelente 33% Buena 22% Pueden mejorar 31% Regulares 14% Malas
	Actividades: Estudio y trabajo.
Conductuales	Campos de interés laboral Diseño de identidad corporativa Aplicaciones Multimedia Presentaciones interactivas
	Beneficios esperados: Ejercicios prácticos Presentación atractiva y clara de los contenidos. Facilidad de ingreso a la información
	Consideración de utilidad : 33% En algunas ocasiones
	17% Siempre
	0% No es útil
	50% Desconozco del lenguaje
	Experiencia previa: 64 % NO 36% SI

Fuente: Elaboración propia

CAPÍTULO III

3. MARCO TEÓRICO

3.1. DEFINICIÓN MULTIMEDIA

“Multimedia es, en esencia, una tendencia de mezclar diferentes tecnologías de difusión de información, impactando varios sentidos a la vez para lograr un efecto mayor en la comprensión del mensaje”. (POLEE, 1999)

Actualmente, el término multimedia hace referencia al uso combinado de diferentes medios de comunicación: texto, imagen, sonido, animación y video. Los programas informáticos que utilizan de forma combinada y coherente con sus objetivos diferentes medios, y permiten la interacción con el usuario son aplicaciones multimedia interactivas. Para que una aplicación sea considerada multimedia deberá integrar por lo menos tres de estos cinco tipos de datos: Texto, gráficos, imagen fija, imagen en movimiento (vídeo – animaciones) y audio (música, voz, sonidos, etc.), que puede difundirse por computadora u otros medios electrónicos.

“Multimedia estimula los ojos, oídos, yemas de los dedos y, lo más importante, la cabeza.” (VAUGHAN, 2010)

“Los sistemas Multimedia, en el sentido que hoy se da al término, son básicamente sistemas interactivos con múltiples códigos”. (BARTOLOMÉ, 1995)

3.1.1. INTERACTIVIDAD

Para que un programa interactivo sea de calidad y pueda ser utilizado fácilmente por el usuario, es necesario que:

- Los códigos y símbolos utilizados por el programa, en este caso los códigos presentados en la pantalla del ordenador, sean comprensibles para el usuario.
- El programa responda con rapidez a las acciones del usuario.
- El sistema utilizado para mostrar las diferentes opciones que puede seleccionar el usuario (sistema de navegación) sea sencillo y comprensible para él.

3.1.2. TIPOS DE ARCHIVOS MULTIMEDIA

TEXTO.

Para Daniel Insa y Rosario Morata (1998: 5) "El texto refuerza el contenido de la información y se usa básicamente para afianzar la recepción del mensaje icónico, para asegurar una mejor comprensión aportando más datos y para inducir a la reflexión" este puede ser estructurado o no estructurado.

IMÁGENES.

Son elementos gráficos codificados según algún formato estándar (mapa de bits, jpeg, png, tiff, etc). Estos pueden ser:

- Iconográficos: permiten representación de conceptos mediante imágenes, Como señala (MARTÍNEZ, 1997) "Siempre un lenguaje icónico tiende a la abstracción por ser un modo de expresión que busca la realidad en los códigos universales".
- Imágenes estáticas: Podemos distinguir diferentes tipos de imágenes: fotografías, representaciones gráficas, fotogramas, ilustraciones, etc.
- Imágenes dinámicas: Transmiten de forma visual sentencias de contenido, dando un sentido propio, la animación permite mayor control sobre ciertas situaciones y esquemas que el video no puede posibilitar.

VIDEO.

Es la secuencia temporal de imágenes a una determinada velocidad (fps), es un recurso que en muchos de los casos se lo utiliza para dar una descripción explicativa de un problema, como por ejemplo los tutoriales.

AUDIO.

Sirven para facilitar la comprensión de la información clarificándola. puede ser :

- Locuciones: Se utilizan para completar el significado de las imágenes.
- Música o efectos sonoros: Dan un efecto motivador captando la atención del usuario.

3.1.3. CLASIFICACIÓN DE LOS MULTIMEDIA

La estructura seguida en una aplicación multimedia es de gran relevancia pues determina el grado y modo de interactividad de la aplicación.

Los sistemas de navegación más usuales en relación a la estructura de las aplicaciones son:

LINEAL.

El usuario sigue un sistema de navegación lineal o secuencial para acceder a los diferentes módulos de la aplicación, solo existe un camino, ejemplo los libros multimedia.



FIGURA III. 9: NAVEGACIÓN LINEAL
Fuente: BELLOCH, C. (2006). "Aplicaciones Multimedia"

RETICULAR.

Se utiliza el hipertexto para permitir que el usuario tenga total libertad para seguir diferentes caminos, ejemplo una enciclopedia electrónica.



FIGURA III. 10: NAVEGACIÓN RETICULAR
Fuente: BELLOCH, C. (2006). "Aplicaciones Multimedia"

JERARQUIZADO.

Combina las dos modalidades anteriores, brinda libertad de selección por parte del usuario y organización de la información atendiendo a su contenido, dificultad, etc.



FIGURA III. 11: NAVEGACIÓN JERARQUIZADA
Fuente: BELLOCH, C. (2006). "Aplicaciones Multimedia"

3.1.4. CARACTERÍSTICAS APLICACIONES MULTIMEDIA

En el contexto de las tecnologías de la información, las aplicaciones multimedia deben cumplir las siguientes características:

- **Controlados por ordenador:** debe ser controlada por un ordenador, aunque el ordenador también participa en distintos grados en la producción de medios, almacenamiento, edición, transmisión.
- **Integrados:** deben minimizar la cantidad de dispositivos necesarios para su funcionamiento;
- **Almacenamiento digital de la información:** los estímulos que percibimos son magnitudes físicas que varían en función del tiempo y/o del espacio. Para almacenar esa información en un ordenador hay que digitalizarla.
- **Interactividad:** aunque es posible la presentación de información multimedia a un observador pasivo, consideraremos que una aplicación multimedia permite al usuario un cierto grado de interacción.

3.1.5. SOFTWARE DE CREACIÓN MULTIMEDIA

Actualmente, existen paquetes de software pensados especialmente para que el diseñador multimedia desarrolle un producto de la forma más rápida y efectiva posible. Estos sistemas se adhieren a diversos paradigmas de desarrollo, estos paradigmas o metodologías son los siguientes:

- **Por guion** (“Scripting Language”): Se trata de entornos de programación en un lenguaje simplificado y especializado. Ofrece la máxima flexibilidad, pero el tiempo de aprendizaje y de desarrollo es normalmente el más largo.
- **Por iconos y flujo de control** (“Iconic/Flow Control”): Es el modelo en general más ágil, Una línea de flujo, que suele dibujarse como si de un programa de diseño gráfico se tratara, une los iconos entre sí describiendo los enlaces y la secuencia de funcionamiento. Sus representantes más acreditados son Authorware, de Macromedia e IconAuthor de Asymetrix.
- **Por tramas** (“Frame”): En este caso también existe una paleta de iconos y se dibujan éstos junto con líneas que los enlazan, pero estos enlaces no definen una línea de flujo de control sino relaciones conceptuales entre ellos. El desarrollo es rápido, aunque los resultados son difíciles de depurar.
- **Orientado a objetos, jerárquico** (“Hierarchical Object”): Utiliza el modelo orientado a objetos con una representación visual basada en iconos y las propiedades de éstos. Su dominio entraña una cierta dificultad, pero permite construir aplicaciones muy complejas.

- **Por enlaces hipermedia** (“Hypermedia Linkage”): Es del estilo del paradigma por tramas pero sin un entorno capaz de representar visualmente los enlaces entre elementos. Es poco abundante.
- **Por códigos de marcado** (“Tagging”). Es el más sencillo y se basa en usar un formato de texto enriquecido con etiquetas, como el SGML o HTML, para construir el documento multimedia.

3.1.6. HERRAMIENTAS DE DESARROLLO MULTIMEDIA

Existen diversas herramientas para diseñar, entre ellas encontramos la suite de Adobe, donde Adobe Flash CS6 es el software encargado para la creación de presentaciones multimedia en combinación de las demás herramientas para el diseño de los elementos que compondrían este tipo de presentaciones.



FIGURA III. 12: HERRAMIENTAS ADOBE PARA DESARROLLO MULTIMEDIA

Fuente: Adobe creative suite

Encajando Adobe Flash entre la calificación de un tipo de software por íconos y flujo de control, por el estilo mismo del manejo del software, y también de orientación a objetos, por tratarse de ActionScript 3.0, lenguaje que como se menciona anteriormente es de POO.

3.1.7. PRODUCCIÓN PARA MULTIMEDIA.

A continuación se muestran algunas de las pautas para la producción de proyectos multimedia:

Contenidos acabados: Los contenidos como texto e ilustraciones deben estar listos ya que fijarán el contenido.

Videos renderizados: formato de archivo de exportación de video no debe generar posibles inconvenientes en el momento de incrustarlo al proyecto.

Temas: deben estar expuestos de forma visible y sencilla, pudiéndose aumentar con links para ampliaciones del tema.

Material gráfico: El material gráfico debe ser propio o disponer de los permisos correspondientes.

Interface: ha de establecerse el formato, cromática, diseño, enlaces, etc. Que deberá contener para soportar a todos los materiales didácticos.

Plataforma: debe ser adaptable para futuros cambios.

Usabilidad: Debe ser ergonómicamente distribuidos los contenidos de manera que el acceso a sus contenidos se sencilla, sin complicaciones para el usuario, independientemente si el usuario haya tenido o no experiencia previa con elementos multimedia.

Navegación: debe ser sencilla de manera que se pueda avanzar y retroceder sin problemas.

Visualización: los videos o imágenes deben visualizarse sin salir del interfaz.

Consultas: desde cualquier página el usuario debe tener la posibilidad de consultar el glosario y bibliografía.

3.2. DISEÑO GRÁFICO PARA MULTIMEDIA

En la actualidad, la imagen audiovisual ha sobrepasado las limitaciones del lenguaje verbal, convirtiéndose en la forma específica de comunicar.

El diseño gráfico como actividad comunicacional, anclada y relacionada con una cierta cultura en un momento dado, es de importancia fundamental en el diseño de interfaces y en el arte de hacerlas más usables.

Los condicionamientos o convenciones culturales y la apreciación estética, junto con los factores humanos y la ergonomía, pueden potenciar o desalentar el uso y la venta de un sistema o herramienta. Por esto, cuanto más elaborado está el mensaje que presenta la “interfaz” y cuantos más estímulos adecuados contenga, más probabilidades existen para que se entienda lo que se trata de comunicar; mejor será el resultado y el mensaje más notorio, grabándose fácilmente en la mente del usuario. Esto hace que perdure más tiempo y sea utilizado con un mínimo esfuerzo.

Cuando diseñamos un material en formato electrónico, como es el caso del material multimedia, es fundamental que tengamos en cuenta cómo debe ser el espacio y la estructura de éste para garantizar una correcta interacción entre el material y el usuario. A este espacio común se le denomina interfaz.

El diseño gráfico tiene la función de solucionar problemas de comunicación, y el diseño del interfaz es de hecho un problema de comunicación que requiere del diseño gráfico para llegar de forma efectiva al usuario.

3.2.1. IDENTIDAD PARA MULTIMEDIA

Muchas veces la identidad corporativa incluye una identidad multimedia, previo al diseño de esta identidad se debe considerar varios aspectos como la proyección cromática y su diferencia en los sistemas de color ya sean CMYK, RGB o Lab de igual forma muchas de las veces una tipografía puede ser legible en un medio impreso mientras que en un medio digital puede presentar problemas de legibilidad o viceversa

3.2.2. TIPOGRAFÍA

En un entorno multimedia, la tipografía es la materia que intermedia entre el receptor y la información. La mala selección de esas formas puede interferir negativamente en la comunicación. Junto con el color, la tipografía puede alterar por completo el significado que asociamos a un diseño.

Para lograr una composición tipográfica adecuada, es necesaria una correcta selección de las familias tipográficas a utilizar, teniendo en cuenta su legibilidad, sus proporciones, el contraste entre los trazos gruesos y delgados, la existencia del remate o su falta, su inclinación y su forma.

LEGIBILIDAD.

En la legibilidad intervienen fundamentalmente: el tipo, el color y el tamaño de los caracteres y números. Para comprobar que los valores y características del texto que aparece en pantalla son correctos se deben cumplir una serie de requisitos básicos:

Tamaño de letra: La letra de 12 puntos es el valor más pequeño en que resultan legibles cómodamente muchas fuentes. Es aconsejable un tipo de letra sencillo en

el que de una relación óptima entre el grosor de los trazos y la altura de los mismos.

Kerning (Interletrado): Se refiere al espacio entre caracteres, un interletrado proporcionado al tamaño de la letra es lo más adecuado, podemos usar el que la fuente nos da por defecto.

Tracking (Interlineado): Es el espacio entre líneas, debería ser dos o tres puntos más altos que el tamaño de la fuente, para que sea cómodo de leer en pantalla, si es muy ajustado, puede provocar que se corte la parte inferior de las letras.

Color y contraste: Color y contraste son dos factores muy importantes. La gama de color escogida para la letra y el fondo van a determinar un factor de integración entre ambos. Un contraste acentuado puede constituir una fuerte llamada de atención, pero si es demasiado agresivo, puede disminuir la legibilidad, será preferible un contraste moderado, en que se logre destacar a la letra del fondo.

Cantidad de texto: Es importante controlar la cantidad de texto que aparece en la pantalla, ya que ésta tiene una resolución mucho más pobre que una página impresa; la pantalla llena de texto es mucho más difícil de leer, el aspecto que no podemos ignorar a la hora de componer con texto, y que supone un error estético, es que a los usuarios no les gusta hacer scroll, o sea, buscar la información que se encuentra debajo del tope de la página.

3.2.3. EL COLOR

El color, además de su función estética y de carácter comunicacional, ofrece al usuario señales de navegación, organización del contenido, jerarquización, áreas activas, etc.

En los sistemas multimedia es imprescindible mantener la atención y concentración del usuario para lo cual presenta un inconveniente crítico encontrar un esquema cromático adecuado para potencializar el contenido evitando distracciones o confusión a los usuarios, cuando se añade color a un diseño, casi siempre éste se convierte en el principal punto de atención, el que más afecta inmediatamente a nuestra memoria.

TEORÍA DEL COLOR.

La paleta de colores RGB consta, básicamente, de tres colores primarios aditivos: Rojo-Verde-Azul.

Además de estos tres colores primarios aditivos RGB, existen tres colores primarios sustractivos o CMY. Estos colores surgen de la siguiente combinación de los primarios aditivos:

Cian = Verde + Azul

Magenta = Rojo + Azul

Amarillo = Rojo + Verde

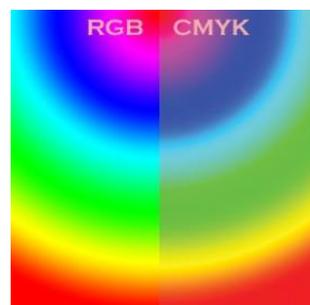


FIGURA III. 13: MODELOS DE COLOR CMYK Y RGB
Fuente: www.proyectacolor.cl

SATURACIÓN, BRILLO Y CONTRASTE

Todo color posee una serie de propiedades que le hacen variar de aspecto y que definen su apariencia final. Entre estas propiedades cabe distinguir:

Saturación: Este concepto representa la pureza o intensidad de un color particular, la viveza o palidez del mismo, y puede relacionarse con el ancho de banda de la luz que estamos visualizando.

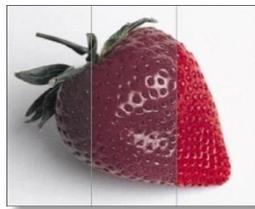


FIGURA III. 14: EJEMPLO SATURACIÓN

Fuente: www.proyectacolor.cl

Brillo: Se usa para describir que tan claro u oscuro parece un color, y se refiere a la cantidad de luz percibida.



FIGURA III. 15: EJEMPLO BRILLO

Fuente: www.proyectacolor.cl

Contraste: El contraste es la diferencia de luminosidad entre las zonas claras y oscuras de una imagen, si la diferencia es grande el contraste es mayor, si la escena tiene los tonos o brillos muy igualados el contraste es pequeño.

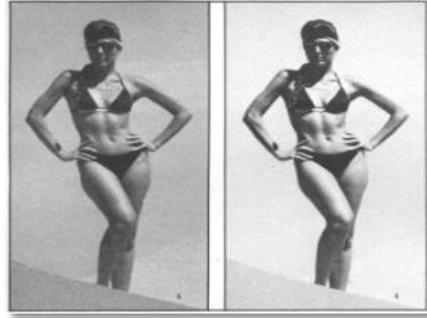


FIGURA III. 16: EJEMPLO CONTRASTE
Fuente: www.proyectacolor.cl

Se pueden diferenciar colores atendiendo a la luminosidad, al color de fondo sobre el que se proyectan.

Cuando dos colores diferentes entran en contraste directo, el contraste intensifica las diferencias entre ambos. El contraste aumenta cuanto mayor sea el grado de diferencia y mayor sea el grado de contacto, llegando a su máximo contraste cuando un color está rodeado por otro.



FIGURA III. 17: TIPOS DE CONTRASTES
Fuente: www.proyectacolor.cl

PSICOLOGÍA DEL COLOR:

Además de muchas otras funciones que cada sitio pueda tener, ante todo intenta comunicar y lo hace con palabras, con imágenes y, en primera instancia, con colores, el color nos produce diferentes sensaciones, pueden alterar nuestras emociones y sentimientos.

Rojo: Asociado a la sangre y el fuego, puede también representar amor, pasión, erotismo, comida, impulsos, acción, aventura, así como también puede tener aspectos negativos como, peligro, agresividad, violencia, e ira. Es un color vigoroso, impulsivo y simpático.

Naranja: Es incandescente, ardiente y brillante. Es sociable y amable, como reconfortante y estimulante. Representa calidez de hogar, confort, creatividad, celebraciones, diversión, es un color en esencia jovial.

Amarillo: Es el color del sol, optimista y moderno también representa la curiosidad, alegría y diversión, el amarillo puede llegar a tener una energía abrumadora por lo que no se aconseja su uso para dominar periodos de tiempo largos.

Verde: Es calmante y tranquilizador, se asocia con lo comprensivo, tolerante, condescendiente y sensitivo, calma la excitación, sugiere vida, estabilidad, ecología, naturaleza, salud, armonía, dinero, induce a tener paciencia.

Azul: Se asocia con el cielo y el agua. Es el color de la modernidad, espiritualidad, pasividad, confiabilidad, éxito, seriedad, calma, poder, profesionalismo. En el plano emotivo, el azul inspira la paz y la introspección, En la práctica está comprobado que el azul es el color más calmante y universalmente preferido.

Violeta: Representa la fantasía, impulsividad, realeza, justicia, ambigüedad, lujo, sueños, demencia. Es un color calmante y algo melancólico. Su calidad es delicada, fresca y aérea.

Rosa: Suavidad, dulzura, inocencia, juventud

Marrón: Tierra, naturaleza, simplicidad, primitivo. Produce una depresión cuando se emplea solo; se debe asociar con el amarillo o el anaranjado.

Gris: Neutralidad, indiferencia, reserva.

Negro: Seriedad, oscuridad, misterio, elegancia, secretos. Asociado a ideas de la muerte y creencias o magias del lado oscuro. Empleado solo, es deprimente, pero es útil para provocar contrastes.

Blanco: Más usado como descanso visual actúa como un maravilloso mediador entre los colores dando equilibrio, cuantos más espacios de descanso existan facilitamos más su visualización, especialmente cuando su contenido es denso (mucho texto) además el blanco psicológicamente representa la inocencia, pureza, limpieza, y simplicidad.

Es aconsejable tener un cierto límite, una paleta de colores que conste de 2 a 3 colores y sus variaciones de luminosidad.

3.2.4. DIAGRAMACIÓN Y RETÍCULA

“En la retícula se debe especificar los elementos que conforman la página, el lugar donde van colocados y las dimensiones que puedan tener, se puede utilizar hojas de papel o algún programa de diseño para dibujar el esquema del sitio solo utilizando texto y líneas simples” (SERRANO, s.f).

Es importante la colocación de elementos de navegación como botones para volver al inicio de la página, ir atrás o adelante y enlaces de ubicación y volver a la página inicial.

El uso de la misma retícula para todas las páginas que conforman el multimedia, aporta una fuerte identidad gráfica, que genera en el usuario la sensación de estar situado y recordar claramente.

MEDIDAS DE LA RETÍCULA.

TABLA III.II: MEDIDAS DE RETÍCULA SEGÚN EL MONITOR

Medidas de la retícula según el monitor	
Monitor	Dimensión de la retícula
Para monitores de 800 x 600 pixeles	783 pixeles de ancho en resoluciones, página sin margen. 769 pixeles de ancho con margen en la página.
Para monitores de 1024 x 768 pixeles	1003 pixeles de ancho en resoluciones sin margen en la página. 983 pixeles de ancho con margen en la página.
Para que todo el contenido esté dentro de la ventana del navegador la medida debe ser de 400 pixeles de altura.	

Fuente: SERRANO José M. "Guía basada en conceptos de usabilidad" (2013).

3.2.5. ERGONOMÍA EN MULTIMEDIA.

“La ergonomía es el criterio que determina la usabilidad de una aplicación dada por la armonía de los elementos del diseño gráfico, su funcionalidad y accesibilidad en la interacción de los usuarios para los que fue concebida” (MOREJÓN, 2009).

Para lograr ergonomía en el hipertexto es necesario el estudio de la interacción de las personas con los objetos con que entran en contacto, particularmente los textos.

Las características que definen la ergonomía en un hipertexto son:

- Comodidad y facilidad de utilización
- Buena relación figura-fondo
- Tipografía adecuada
- Márgenes a ambos lados del texto y entre párrafos
- Imágenes que completan la información textual o la entorpecen
- Visualización agradable
- Luminosidad, Visibilidad

Estos elementos básicos se combinan unos con otros en un producto multimedia, y de esta combinación surge un resultado final en el que tienen importancia una serie de conceptos propios del diseño gráfico, entre los que se destacan:

- **Las agrupaciones:** conjuntos de elementos relacionados mediante proximidad, semejanza, continuidad o simetrías.
- **La forma:** forma de cada elemento gráfico aislado y de las agrupaciones de elementos.
- **Los contornos:** partes límites de los elementos, que permiten distinguirlos de los demás y del fondo, pudiendo estar definidos mediante borde, cambios de color o cambios de saturación.
- **La ubicación:** lugar que ocupa cada elemento gráfico o agrupación de ellos en el espacio.

- **El tamaño:** tamaño relativo de cada elemento gráfico respecto a los que le rodean. Escalas.
- **El color:** color de cada elemento individual, colores de cada agrupación de elementos, conjunto total de colores usados en un producto multimedia, disposición relativa de los elementos con color y armonía entre colores.
- **El contraste:** intensidad de visualización de cada elemento con relación a los que le rodean y al producto multimedia completo.
- **El equilibrio:** cada producto multimedia conlleva un sistema de referencia espacial que consigue un nivel de equilibrio mayor o menor.
- **La simetría:** disposición espacial regular y equilibrada de los elementos que forman la composición gráfica.

Se integran todos los elementos del diseño gráfico abordados con anterioridad para el logro de una correcta comunicación visual con el usuario, a partir de la selección de cada uno de ellos en función garantizar la calidad de producto informático.

3.2.6. USABILIDAD EN MULTIMEDIA.

Los siguientes elementos tienen mucha importancia para determinar la usabilidad de un multimedia:

- La utilidad.
- La facilidad de uso.
- La curva de aprendizaje (el tiempo que hemos de invertir para garantizarnos un dominio aceptable de los elementos que componen la interfaz).

Aspectos generales que hay que tener en cuenta para un diseño de material multimedia que brinde usabilidad:

TABLA IIII.III: ASPECTOS PARA EL DISEÑO DE MATERIAL MULTIMEDIA RESPECTO A LA USABILIDAD

Aspectos para el diseño de material multimedia respecto a la usabilidad	
Ámbitos	Aspectos que debemos considerar
General	<p>Simplicidad: Sencillo y Directo</p> <p>Coherencia:</p> <ul style="list-style-type: none"> • Uso del color • Uso de la letra • Estilo de los gráficos • Espacio en blanco
Componentes de una pantalla	<ul style="list-style-type: none"> • Información (de orientación y de navegación) • Direccionamiento-Instrucciones • Interactividad
Pantallas	<ul style="list-style-type: none"> • Centrar la atención • Despertar y mantener el interés • Implicar al usuario • Facilitar la navegación a través del contenido
Lenguaje	<ul style="list-style-type: none"> • En función de los usuarios finales • Ideas claras • Frases cortas • Informal • Ejemplos familiares • Integrador (no sexista/no racista)
Estética	<ul style="list-style-type: none"> • Equilibrio (gráficos y texto) • Coherencia de estilos y repetición • Unidad (de estilos y de distribución) • Espacio en blanco • Mínima memorización
Color	<ul style="list-style-type: none"> • Realistas • No saturación
Tipos de letra	<p>Recomendadas</p> <ul style="list-style-type: none"> • Arial, Courier, Helvética, Times, etc <p>No recomendadas, entre otras</p> <ul style="list-style-type: none"> • London, Zapf Chancery
Audio	<ul style="list-style-type: none"> • Agradable, divertido, no monótono, cuando sea necesario • Como refuerzo (positivo/negativo) • Que el usuario pueda controlarlo • Volumen, pausa y repetición pueden ayudar a captar y mantener la atención <p>Formatos de audio más comunes:</p> <ul style="list-style-type: none"> • WAV, AU, MIDI (para sonido más profesional)
Vídeo	<ul style="list-style-type: none"> • En función de las necesidades • Cuando se necesiten procesos dinámicos • Cuando se necesitemos fotogramas realistas • El usuario tiene que poder controlarlo <p>Principales formatos</p> <ul style="list-style-type: none"> • Quicktime, AVI, MPEG
Derechos de autor	<p>A la hora de diseñar material multimedia con finalidades divulgativas, hemos de tener en cuenta la utilización de informaciones originales (texto, audio y vídeo, imágenes, etc.) para evitar problemas legales.</p>

Fuente: (GRUPO DE INVESTIGACIÓN EN INTERACCIÓN Y ELEARNING, 2010)

3.3. EDUCACIÓN MULTIMEDIA

3.3.1. EDUCACIÓN VIRTUAL

“La enseñanza virtual tiene asociado un modelo educativo propio, con varios cambios destacados con respecto al modelo clásico de aprendizaje presencial” (CORDÓN, 2005). La educación virtual tiene como eje evaluador principal al estudiante ya que es el principal beneficiario del éxito o fracaso de los conocimientos transmitidos, siendo el docente un eje de aportación de los contenidos. Los aspectos del proceso educativo que se han visto más directamente afectados por la utilización de los medios digitales son:

- La presentación de la información del profesor a los estudiantes.
- El material de estudio y consulta
- La interacción profesor-estudiante
- El propio proceso de aprendizaje y autoevaluación del estudiante.

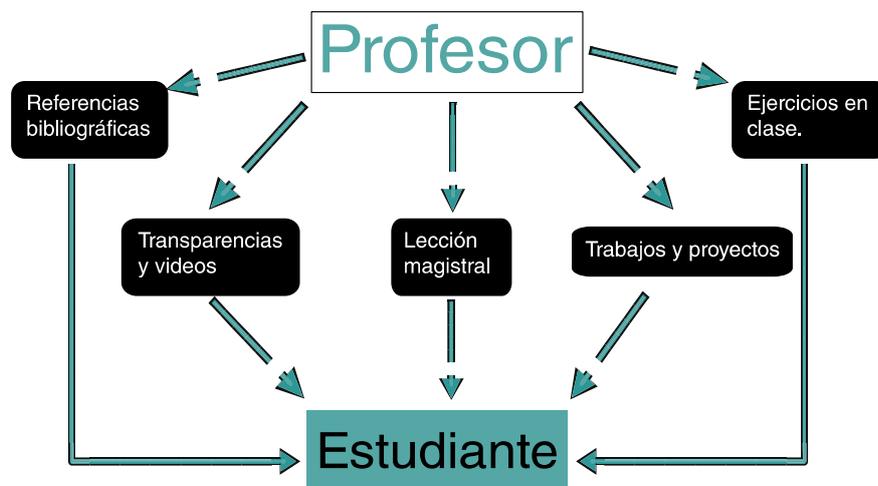


FIGURA III. 187: MODELO DE APRENDIZAJE CLÁSICO CENTRADO EN EL PROFESOR

Fuente: *ENSEÑANZA VIRTUAL: FUNDAMENTOS, PERSPECTIVAS ACTUALES Y VISIÓN*, Oscar CORDÓN (2005)

La habitual enseñanza presencial, centrada en muchos casos en la lección magistral en la que el profesor tiene el papel principal, se transforma en un modelo de enseñanza virtual no presencial, en el cual es el estudiante el elemento más activo en el proceso de (auto) aprendizaje.

Todos los elementos que forman parte del modelo pedagógico virtual se ponen a disposición del estudiante para que pueda gestionar su propio proceso formativo.

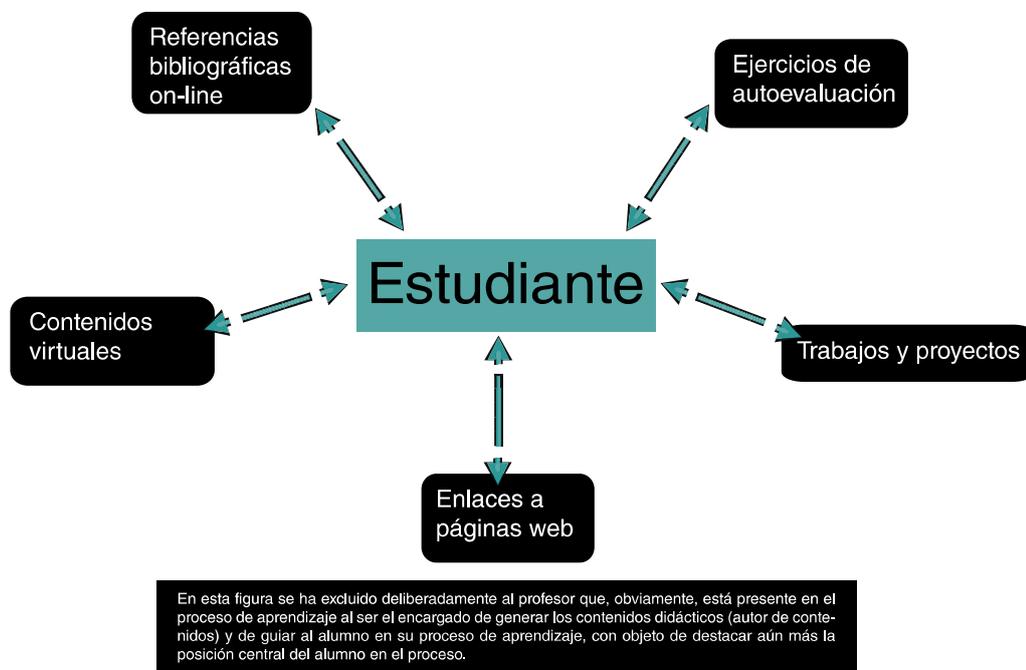


FIGURA III. 198: MODELO DE APRENDIZAJE VIRTUAL CENTRADO EN EL ESTUDIANTE

Fuente: *ENSEÑANZA VIRTUAL: FUNDAMENTOS, PERSPECTIVAS ACTUALES Y VISIÓN*, Oscar CORDÓN (2005)

Para optimizar este modelo, hay que tener presentes dos aspectos cruciales: la calidad pedagógica (de los materiales virtuales diseñados y de la metodología

docente empleada) y el énfasis en el apoyo personalizado. En consecuencia, dicho modelo se sustenta en dos pilares básicos:

Una vertiente didáctica, que permita al docente estructurar adecuadamente estos materiales mediante las metodologías del diseño instructivo y guiar de un modo adecuado el proceso de auto-aprendizaje realizado por el estudiante, haciendo énfasis en el apoyo personalizado.

Una vertiente tecnológica, basada en el uso de las redes telemáticas y las TIC en la creación de contenidos digitales multimedia que sean atractivos y fácilmente accesibles desde navegadores web.

ESTUDIANTE.

Debido a que la educación virtual es diferente a la presencial y por lo tanto se requiere de disposición y apertura para actuar en consecuencia. Los roles como docente y estudiante cambian, de la misma forma que los procesos de enseñanza y aprendizaje.

DOCENTE.

El profesor en calidad de tutor tiene que cumplir los siguientes roles y funciones:

Roles

- Consultor de información.
- Colaborador de grupo.
- Trabajador solitario.
- Facilitador de aprendizaje.
- Desarrollador de cursos y materiales.
- Supervisor académico.

3.3.2. MULTIMEDIA EN LA EDUCACIÓN VIRTUAL

La educación es seguramente el espacio en el que el uso del modelo multimedia puede aportar una mayor innovación y beneficio. La difusión de este modelo puede suponer una reforma radical del proceso educativo a todos los niveles, desde los primarios a los superiores, en este caso, las presentaciones interactivas tienen mayor protagonismo, ya que permiten al alumno manejar elementos y escenarios interesantes.

El potencial pedagógico de los multimedia interactivos ha sido confirmado por numerosos estudios llevados a cabo en Europa y en Estados Unidos. Por ejemplo en el Reino Unido, el National Council for Education Technology ha realizado un inventario de todos los efectos positivos que proporciona el uso de las producciones multimedia en educación, destacando: la motivación de estudiantes que fracasan con métodos tradicionales, la reducción de tasa de fracaso escolar, el estímulo de la cognición, el gusto por la lectura y la escritura y la adaptación a las capacidades individuales, entre otros.

Los programas multimedia educativos se encuentran en el núcleo de un debate sobre el cambio de los sistemas de educación y formación. Los métodos de enseñanza, que se basaban en el siglo pasado en una formación de masas, han evolucionado hasta satisfacer las necesidades individuales de formación. Los sistemas multimedia, cuya ventaja principal es la interactividad, se adaptan muy bien a este nuevo enfoque, ya que favorece el uso de la información en un contexto apropiado, de forma personalizada y la creación de un entorno virtual en el que el alumnado puede valorar instantáneamente el impacto de sus acciones.

3.3.3. RECURSOS MULTIMEDIA PARA EL APRENDIZAJE.

Estas tecnologías están centradas específicamente en el aprendizaje activo por parte del alumno, mediante la interacción del mismo con los objetos de aprendizaje. Con estas tecnologías, principalmente conductistas, se pueden abordar objetivos formativos relacionados con el entrenamiento para ciertas acciones, la simulación de procesos o la adquisición de habilidades mediante la interacción con la propia herramienta.

HIPERMEDIAS INFORMATIVOS.

Son documentos que nos aportan información y, al igual que las enciclopedias y diccionarios en papel, son recursos de consulta de información tales como:

- **Libros multimedia.** Se parecen a los libros convencionales en formato papel en cuanto a que mantienen una estructura lineal para el acceso a la información.
- **Enciclopedias multimedia.** Al igual que las enciclopedias y diccionarios en papel son recursos de consulta de información, por lo que su estructura es principalmente reticular para favorecer el rápido acceso a la información.
- **Hipermedias.** Son documentos hipertextuales, esto es con información relacionada a través de enlaces, que presentan información multimedia. Su estructura es en mayor o menor grado jerarquizada, utilizando diferentes niveles de información.

HIPERMEDIAS PARA EL APRENDIZAJE Y LA INTERVENCIÓN.

- **Tutoriales.** Son semejantes a los programas de ejercitación pero presentan información que debe conocerse o asimilarse previamente a la realización de los ejercicios.
- **Resolución de problemas.** Estas aplicaciones multimedia tienen por objeto desarrollar habilidades y destrezas de nivel superior, basándose en la teoría constructivista. Para ello, se plantean problemas contextualizados en situaciones reales, que requieren el desarrollo de destrezas tales como comprensión, análisis, síntesis, etc. Para ello se proporcionan materiales y recursos para su solución, junto a materiales adicionales para profundizar en el tema planteado.
- **Caza del tesoro.** Una caza del tesoro popularizada en estados unidos más no como debería en los países de habla hispana es un documento hipermedia (página web) en la que se presentan una serie de preguntas sobre un determinado tema, junto a una lista de direcciones web en las que se pueden buscar las respuestas. Como punto final se incluye una pregunta "la gran pregunta", que los alumnos deben responder a partir de la comprensión e integración de lo aprendido durante la búsqueda y resolución de las preguntas.
"Las cazas del tesoro son estrategias útiles para adquirir información sobre un tema determinado y practicar habilidades y procedimientos relacionados con las tecnologías de la información y la comunicación en general y con el acceso a la información a través de la Internet en particular" (ADELLE, 2003).

3.3.4. BENEFICIOS DE LA EDUCACIÓN VIRTUAL

Luego de revisar varias ponencias, escritos y estudios de educación virtual se ha determinado que los principales beneficios de la educación virtual frente a otros tipos de educaciones son:

- Los estudiantes pueden adaptar el estudio a sus horarios personales.
- Los estudiantes podrán seguir fácilmente el ritmo de trabajo marcado por el profesor y sus compañeros del curso.
- Existe mejora de la calidad de aprendizaje.
- Ahorro de tiempo y dinero. El estudiante no tiene que centrarse al centro de estudio.
- Promueve la interacción del compañerismo.
- El estudiante recibe una instrucción más personalizada.
- Mejora el desempeño del docente, por cuanto parte del tiempo que antes se dedicaba a la clase, se invertirá en un mejor diseño curricular e investigación.
- Ampliación de cobertura, la cual mejora el acceso a la educación, eliminando las barreras de lugar y tiempo, características de la educación tradicional.
- Desarrolla la creatividad del estudiante, motiva a este tiene que buscar la información por sí mismo.

3.3.5. PROBLEMAS DE LA EDUCACIÓN VIRTUAL

La educación virtual ofrece una gran cantidad de ventajas pero como cualquier método o práctica de trabajo tiene ciertos problemas o déficit los cuales están sintetizados en las siguientes premisas:

- Acceso desigual en la población debido a limitaciones económicas.
- Limitaciones técnicas, entre ellas desconexiones e imprecisiones.
- La comunicación de red y la vía excedente de los alumnos puede desviar la atención de los alumnos.
- Alto costo del material de los equipos y de la producción del material.
- Los materiales pueden no estar bien diseñados y confeccionados, por lo que necesita de un grado de experiencia para diseñar el entorno de trabajo.
- Los estudiantes podrían aislarse y no planificar correctamente sus actividades y horarios.
- No se ofrece el mismo contacto persona a persona así como las clases presenciales.
- Se requiere un esfuerzo de mayor responsabilidad y disciplina por parte del estudiante.

3.3.6. LA EDUCACIÓN VIRTUAL EN EL MUNDO

Para una cuarta parte de la población mundial la vida cotidiana ya es muy diferente de lo que fue hace doce años. El comercio electrónico, el gobierno digital y la educación virtual en todos sus niveles hoy son una realidad en los países digitalizados

PENETRACIÓN DEL INTERNET EN EL MUNDO.

Un 30% de la población de la Tierra tiene acceso a la Red global, según informe del portal Pingdom en el 2012.

Si hablamos de los internautas, unos 920 millones, vive en Asia y casi 480 millones, en Europa. En tercer lugar está América del Norte, luego vienen América Latina y África, y concluye la lista la región de Australia y Oceanía. Casi la mitad de todos los que usan la Red son jóvenes de menos de 25 años.

Existen unos 3.100 millones de direcciones de correo electrónico, más de 800 millones de perfiles de Facebook y alrededor de 225 millones de cuentas en Twitter.

La siguiente tabla muestra la penetración del internet en las diferentes regiones mundiales:

TABLA IVI.IV: USUARIOS DE INTERNET EN EL MUNDO

ESTADÍSTICAS MUNDIALES DEL INTERNET Y DE LA POBLACION					
Regiones	Poblacion (2012 Est.)	Usuarios Junio 2012	% 30, Población (Penetración)	Usuarios % Mundial	Facebook Sept 30, 2012
África	1,073,380,925	167,335,676	15.6 %	7.0 %	48,262,820
Asia	3,922,066,987	1,076,681,059	27.5 %	44.8 %	235,989,160
Europa	820,918,446	518,512,109	63.2 %	21.5 %	243,230,440
Oriente Medio	223,608,203	90,000,455	40.2 %	3.7 %	22,793,140
Norte América	348,280,154	273,785,413	78.6 %	11.4 %	184,177,220
Latinoamérica / Caribe	593,688,638	254,915,745	42.9 %	10.6 %	188,339,620
Oceanía / Australia	35,903,569	24,287,919	67.6 %	1.0 %	14,614,780
TOTAL MUNDIAL	7,017,846,922	2,405,518,376	34.3 %	100.0 %	937,407,180

NOTAS: Las Estadísticas de Usuarios Mundiales del Internet fueron actualizadas a Junio 30, 2012. Los datos de población se basan en cifras para 2012 del US Census Bureau, en su mayoría. Los datos de usuarios provienen de información publicada por Nielsen Online , por ITU , por Internet World Stats, Los suscriptores de Facebook se basan en datos de dicha organización

Fuente: <http://www.internetworldstats.com/stats.htm>
 Basado en: 2.405.518.376 usuarios de internet en Junio 30 del 2012

Como se puede ver en la siguiente gráfica, del porcentaje total de usuarios de Internet actualmente Asia impone el mayor porcentaje de usuarios de este servicio, lo que llevaría a pensar que Asia posiblemente sea el candidato óptimo para el uso de educación virtual, pero esta suposición no es real pues al analizar la tabla anterior vemos que el porcentaje de penetración de internet en Asia es muy bajo en comparación al porcentaje de penetración que tienen tanto América del Norte como Oceanía.

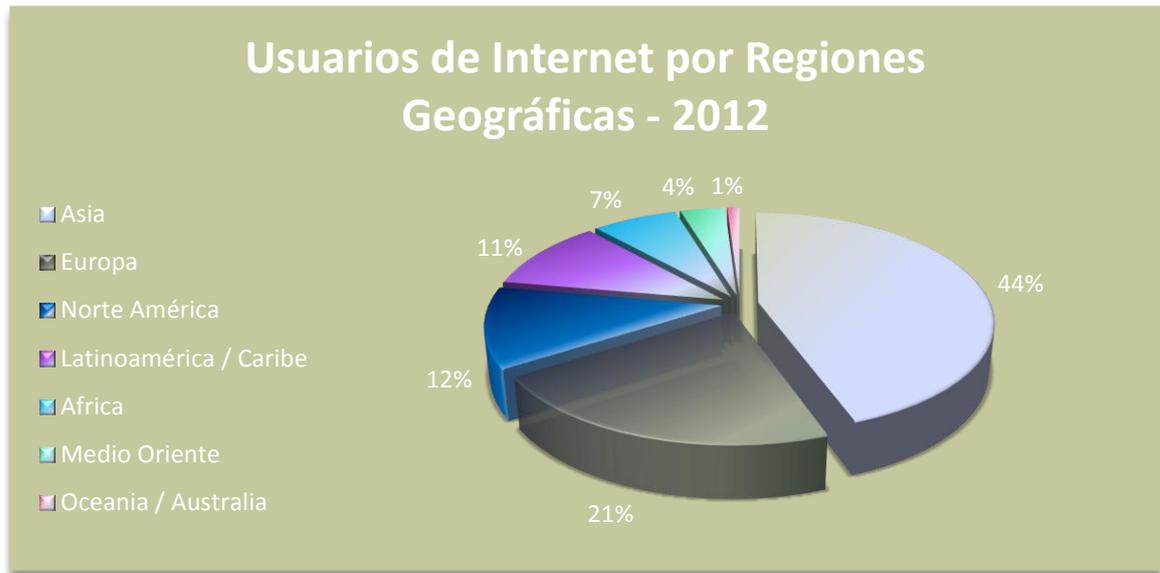


FIGURA III: 19 USO DEL INTERNET EN EL MUNDO
Fuente: <http://www.internetworldstats.com/stats.htm>

3.3.7. LA EDUCACIÓN VIRTUAL EN ECUADOR

Una vez que hemos analizado cuan avanzada está la educación virtual a través del mundo y sabemos cuáles han sido los motivos para que en ciertos sectores no se haya desarrollado al ritmo que se esperaba, vamos a investigar el estado de la educación virtual en el Ecuador y luego analizar los motivos por los que se ha llegado a este punto de desarrollo.

Según encuesta del INEC (Instituto Nacional de Estadísticas y Censos) El 31,4% de la población utiliza Internet en el Último año. Los pobres aumentaron su acceso. La mayor parte del uso es por razones de comunicación.

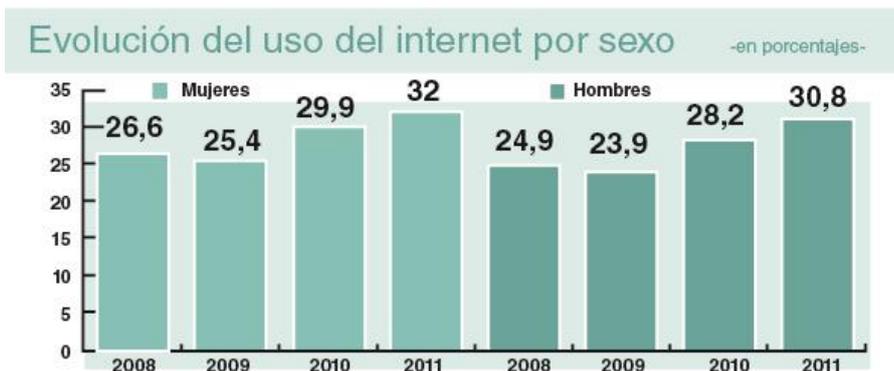


FIGURA III. 20: USO DEL INTERNET EN ECUADOR POR SEXO

Fuente: *Explored.com/ datos INEC*

En donde usan internet



Para qué usaron internet

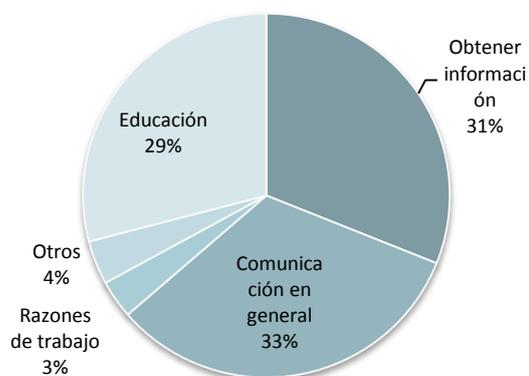


FIGURA III. 211: USO DE INTERNET EN EL PAÍS

Fuente: *Explored.com/ datos INEC*

La última encuesta del Instituto Nacional de Estadística y Censos (INEC) sobre el acceso de los ecuatorianos a las Tecnologías de la Información y Comunicación (TIC), realizada en diciembre de 2011, revela que el 24,7% de los hogares tiene una computadora de escritorio y el 9,8% tiene una portátil. Esto es, el 35% de los 3 815 000 hogares que existen en el país. El sondeo realizado en 579 centros poblados abarca una muestra de 21 768 viviendas.

El porcentaje de personas que usa Internet en los Últimos 12 meses corresponde a 31,4%; hace cuatro años, esa estadística era del 25,7%, devalando un alza de 5,7 puntos porcentuales. Quienes mayor uso le dieron en 2011 a Internet en el

Ecuador fueron los hombres con el 32%, frente al 30,8% para las mujeres. En ambos casos, la cifra subió seis puntos en cuatro años (ver gráfico).

En cuanto a las edades, se determina que seis de cada 10 jóvenes de entre los 16 y 24 años tuvo un uso activo del Internet (59,4%), seguidos por quienes tienen de 25 a 34 años (39,6%). Quienes menos usaron el Internet fluctúan entre los 65 y 74 años (3,3%).

El 57,3% de los usuarios ingresa por lo menos una vez al día; mientras el 36,9% lo hizo en promedio una vez por semana. El principal uso del Internet se enfoca en las comunicaciones con la familia y amigos, y los ingresos a la web se realizaron en su mayoría desde el hogar del usuario.

3.3.8. LA EDUCACIÓN VIRTUAL EN LA ESPOCH

En el año 2006 gracias a la iniciativa del Ing. Byron Vaca ex Director de DESITEL, Ing. Alberto Arellano docente FIE, y Dr. Carlos Buenaño ex Director de DESITEL, la plataforma virtual de la ESPOCH fue creada y alojada e instalada, luego de ello el DESITEL (Departamento de Sistemas y Telemática) se dispone a construir un módulo de migración de datos que permita migrar las mallas académicas, cargas horarias, horas clases, estudiantes y docentes de la ESPOCH, este procedimiento debía ser desde el Sistema Académico Institucional (OASIS) al Sistema de Manejo de Contenidos MOODLE.

La producción la nueva plataforma virtual de la ESPOCH (EVIRTUAL-ESPOCH) da inicio el jueves 17 de enero del 2008. Desde la fecha la plataforma ha sufrido varias actualizaciones llegando a tener una plataforma MOODLE estable.

Para con ella atender a los usuarios docentes y estudiantes en lo que respecta a las herramientas SMC de la ESPOCH. La plataforma en estos 6 años ha evolucionado y su usabilidad ha crecido un **36,92%** y se aspira llegar hasta finales del 2013 a un **45%** de los cursos virtuales migrados desde el sistema académico. Además con una media de transacciones diarias de 6657, 31961 en total usuarios entre activos y pasivos; 1625 (1025 pasivos y 600 activos) cursos en el periodo Septiembre 2012- Febrero 2013. 12472 cursos virtuales entre activos y pasivos. 28 módulos de actividades instaladas.

Metodología y esquema de desarrollo.

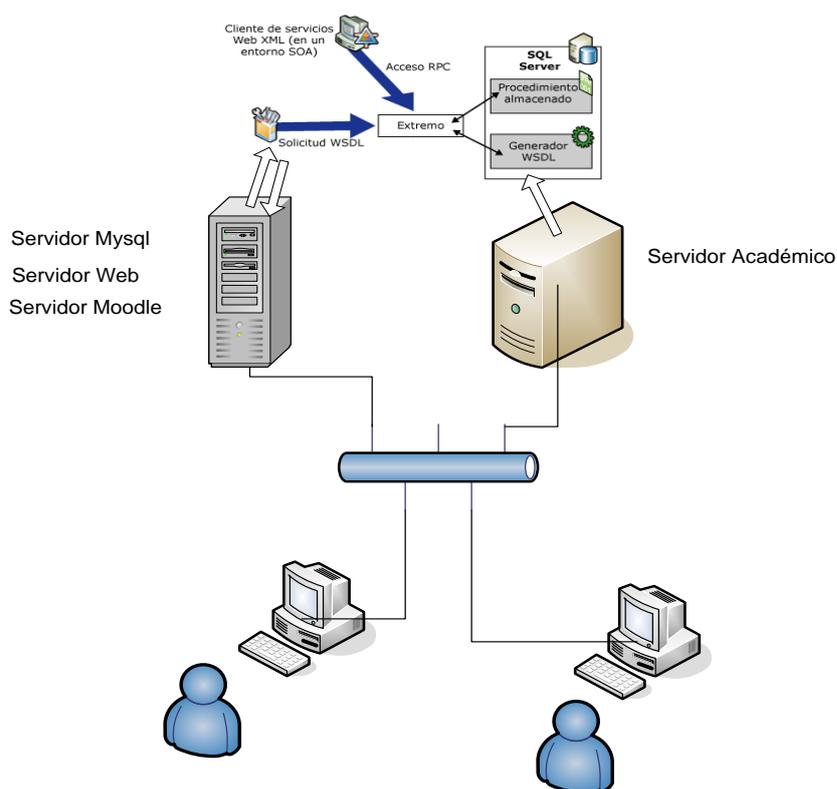


FIGURA III. 22: METODOLOGÍA Y ESQUEMA DE DESARROLLO
Fuente: Departamento de Sistemas y Telemática ESPOCH

USO DE LA PLATAFORMA VIRTUAL.

A continuación detallaremos las estadísticas de la usabilidad de la plataforma virtual de la ESPOCH en el periodo académico Septiembre 2012-Febrero 2013.

1024 cursos pasivos: Es decir cursos que existen dentro de la plataforma virtual pero que no son utilizados por los docentes

599 cursos activos: Refiriéndose a los cursos que son utilizados por los docentes dentro de la plataforma virtual.

Teniendo un total de **1623** cursos existentes en el periodo académico Septiembre 2012-Febrero 2013.

Usabilidad de la plataforma virtual ESPOCH

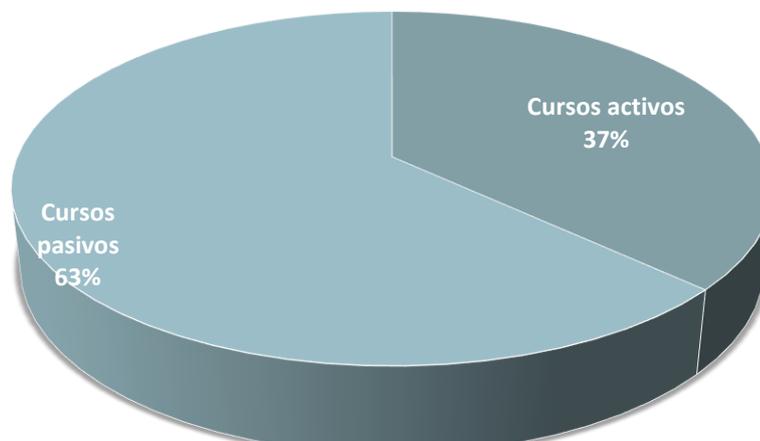


FIGURA III. 22: USABILIDAD DE LA PLATAFORMA VIRTUAL ESPOCH

Fuente: Elaboración propia basado en la información proporcionada por el "Departamento de Sistemas y Telemática ESPOCH"

3.4. METODOS DE ENSEÑANZA APRENDIZAJE VIRTUALES

3.4.1. E-LEARNING Y B-LEARNING

EL E-LEARNING. Tuvo un gran auge en la década de los noventa provocando una oferta a veces indiscriminada de cursos y postgrados en internet que llevó en ocasiones a la brecha o separación con respecto a las clases magistrales y la presentación tradicional unilineal del conocimiento como dos fórmulas opuestas de docencia.

Algunos de los puntos en los que el e-learning tuvo un fracaso fueron, que la mayoría de estudiantes carecían de iniciativa personal y auto-organización para cumplir con las exigencias de este modelo formativo, careciendo en su diseño de motivación, además de escasa formación de sus docentes.

EL B-LEARNING. Es el diseño docente en el que tecnologías de uso presencial (físico) y no presencial (virtual) se combinan con objeto de optimizar el proceso de aprendizaje, nos plantea un uso eficaz de las TICs (Tecnologías de la Información y Comunicación), se plantea como una solución al problema del rechazo o dificultad de uso de las nuevas tecnologías, y constituye un modelo de aprendizaje en el que conseguimos una serie de combinaciones fundamentales:

**TABLA VI.V: COMPARATIVA ENTRE EL MODELO PRESENCIAL Y EL MODELO VIRTUAL-
PRESENCIAL DE APRENDIZAJE**

Comparativa entre el modelo presencial y el modelo virtual-presencial de aprendizaje	
MODELO PRESENCIAL	MODELO VIRTUAL-PRESENCIAL
Presencialidad	+ virtualidad
Relación profesor-alumnos	+ relación alumnos-propio aprendizaje
Transmisión de conocimientos	+ desarrollo de capacidades
Cultura escrita-oral	+ cultura audiovisual
Uso tradicional tecnologías (pizarra, libro, etc.)	+ nuevas tecnologías (campus virtual)

Fuente: "BLENDED LEARNING: MODELO VIRTUAL-PRESENCIAL DE APRENDIZAJE Y SU APLICACIÓN EN ENTORNOS EDUCATIVOS" ALEMANY Dolores (2012)

VENTAJAS DEL B-LEARNING

- Desarrollo de habilidades y disciplina para aprender a aprender.
- El b-learning es esencial para profesionales en constante actividad.
- Promueves la autogestión fomentando el desarrollo de comunicación y manejo de información por computadora.
- Interacción con compañeros y maestros conociendo nuevas maneras de comunicación. (foros, chats, portafolio).
- Aprender a ritmo propio, al igual que en el e-learning.
- Optimizar tiempos, se flexibilizan los horarios de clases presenciales.
- Tienes la experiencia de manejar plataformas virtuales.
- El proceso de enseñanza-aprendizaje se vuelve más dinámico y continuo

3.4.2. TEORÍAS DE APRENDIZAJE EN LOS ENTORNOS VIRTUALES

Para crear un entorno virtual para el aprendizaje se debe inspirar en las mejores teorías educativas y pedagógicas, para esto Leflore (2000) indica en propuesta el uso y análisis de tres teorías de aprendizaje en entornos virtuales:

TEORÍA GESTALT.

Señala que la presentación de lo visual interfiere en el aprendizaje, es decir debe basarse en las leyes de la percepción como el contraste figura-fondo, la simplicidad, proximidad, simetría, similaridad, y cierre.

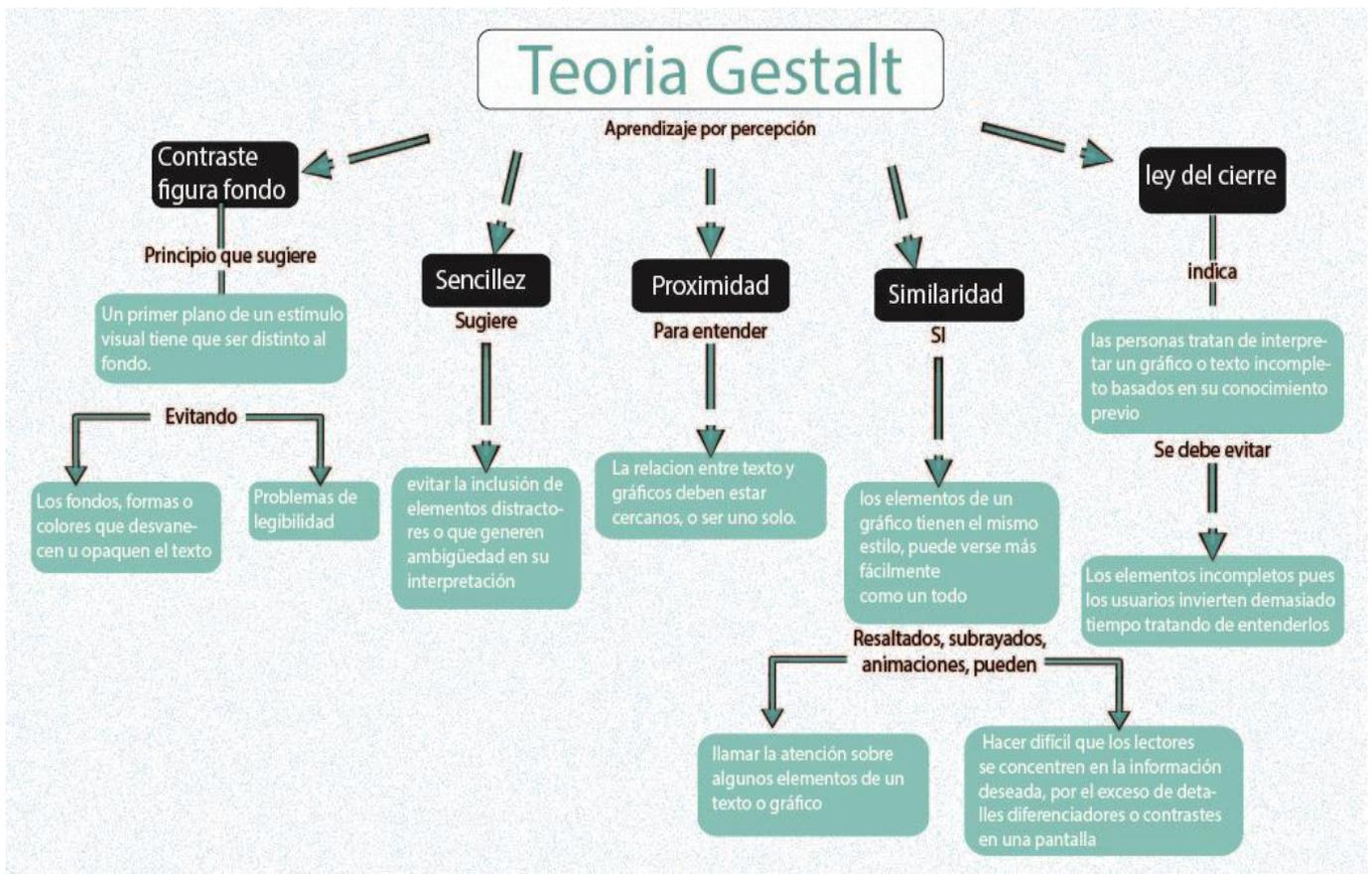


FIGURA III. 234: TEORÍA GESTALT

Fuente: Elaboración propia, basada en: "LA ENSEÑANZA VIRTUAL EN LA EDUCACIÓN SUPERIOR" ÁLVAREZ Octavio (2002)

De acuerdo con (LEFLORE, 2000), estas serían algunas pautas esenciales para el diseño de instrucción en la Red basadas en la teoría Gestalt:

- Asegurar que el fondo no interfiera con la nitidez de la información presentada en el primer plano.
- Utilizar gráficos sencillos para presentar información.

- Agrupar la información que tenga relación entre sí, de tal manera que el usuario pueda captar fácilmente su unidad o conexión.
- Utilizar discretamente el color, la animación, los destellos intermitentes, u otros efectos para llamar la atención hacia ciertas frases del texto o áreas gráficas.
- No utilizar información textual o gráfica incompleta.

Al introducir un tema nuevo emplear vocabulario sencillo.

TEORÍA COGNITIVA.

(ÁLVAREZ, 2002) Sugiere que las actividades de desarrollo conceptual, el uso de medios para la motivación, y la activación de esquemas previos, pueden orientar y apoyar de manera significativa el diseño de materiales de instrucción siendo estos medios que representan la actividad cognitiva.

Estos son generalmente formas geométricas con texto incluido y conectadas por medio de líneas. La presentación inicial de un concepto en la red puede apoyarse en estos recursos gráficos, estos pueden ser esquemas de organización de contenidos que proporcionan una síntesis con el tema y conexión con temas de conocimiento previo siendo posible manejarlos como texto en pantalla o archivos adjuntos. La activación de esquemas es un método aplicable en la enseñanza virtual. La información nueva debe relacionarse con las estructuras cognitivas que posee el alumno. Una forma de activar esquemas es elaborar una serie de preguntas y un programa que evalúe la variedad de respuestas que puedan dar los estudiantes.

(LEFLORE, 2000) Sintetiza algunas pautas para el diseño de instrucción en la Red derivadas de las teorías cognitivas:

- Ayudar a los alumnos a estructurar y organizar la información que deben estudiar mediante listas de objetivos, mapas conceptuales, esquemas, u otros organizadores gráficos.
- Utilizar actividades para el desarrollo conceptual conformando pequeños grupos de alumnos y proporcionándoles listas de ejemplos y contraejemplos de conceptos. Las discusiones pueden darse en forma sincrónica o asincrónica.
- Activar el conocimiento previo mediante listas de preguntas que respondan los estudiantes, colocando una pregunta en una lista de correo, o mediante organizadores previos.
- Utilizar en forma discreta recursos motivacionales como gráficos, sonidos, o animaciones. No es necesario usarlos en cada página, unidad o lección

CONSTRUCTIVISMO.

(ÁLVAREZ, 2002) “Cada individuo posee una estructura mental única a partir de la cual construye significados interactuando con la realidad. Una clase virtual puede incluir actividades que exijan a los alumnos crear sus propios esquemas, mapas, redes u otros organizadores gráficos”.

Gran parte de lo que aprendemos sobre el mundo depende de la comunicación con otras personas. En la educación virtual la interacción social puede lograrse a través de chats, correo electrónico, foros de discusión, videoconferencias, etc. Los alumnos contestan preguntas, resuelven problemas, y realizan actividades en forma grupal.

Es así que se definen algunas pautas derivadas del constructivismo para la enseñanza virtual.

- Organizar actividades que exijan al alumno construir significados a partir de la información que recibe. Se le pide que construya organizadores gráficos, mapas, o esquemas.
- Proponer actividades o ejercicios que permitan a los alumnos comunicarse con otros. Orientar y controlar las discusiones e interacciones para que tengan un nivel apropiado.
- Cuando sea conveniente permitir que los estudiantes se involucren en la solución de problemas a través de simulaciones o situaciones reales.

MÉTODO DE CASOS

“El método de casos difiere de los sistemas de enseñanza tradicionales porque exige que el alumno tome parte activa en el análisis de los problemas y en la toma de decisiones para la solución a situaciones reales muy específicas” (OÑATE, 2009).

Un aspecto muy importante del método de casos, es el aprender de la experiencia de otros casos similares para la toma de decisiones en el momento de solucionar un problema. La solución de un mayor número de casos incrementará la habilidad para identificar problemas y formular soluciones a problemas reales.

En el método de casos, el proceso que se siga para tomar decisiones y las decisiones mismas, sustentadas en un análisis adecuado, son la clave.

3.5. PROGRAMACION ORIENTADA A OBJETOS Y ACTION

SCRIPT 3.0

3.5.1. PROGRAMACIÓN ORIENTADA A OBJETOS

Los lenguajes de programación tradicionales no orientados a objetos, como C, Pascal, BASIC, o Modula-2, basan su funcionamiento en el concepto de procedimiento o función. Una función es simplemente un conjunto de instrucciones que operan sobre unos argumentos y producen un resultado. De este modo, un programa no es más que una sucesión de llamadas a funciones, ya sean éstas del sistema operativo, proporcionadas por el propio lenguaje, o desarrolladas por el mismo usuario.

Un objeto es la representación en un programa de un concepto, y contiene toda la información necesaria para abstraerlo: datos que describen sus atributos y operaciones que pueden realizarse sobre los mismos.

“La programación orientada a objetos es una nueva forma de pensar, una manera distinta de enfocar los problemas” (IZQUIERDO, 2007).

Ahí radica la dificultad de aprender un lenguaje totalmente orientado a objetos, sin conocer previamente los pilares de la programación orientada a objetos.

Está basado en varias técnicas, incluyendo herencia, modularidad, polimorfismo y encapsulamiento.

VENTAJAS:

- Reutilización de código
- Desarrollo de interfaces gráficas en forma rápida
- Bases de datos más eficientes.

3.5.2. FUNDAMENTOS DE LA POO

Este paradigma de programación orientada a objetos se fundamenta en las siguientes propiedades.

ABSTRACCIÓN:

La abstracción es la propiedad de los objetos que consiste en tener en cuenta sólo los aspectos más importantes desde un punto de vista determinado y no tener en cuenta los restantes aspectos. Es la abstracción la que permite ignorar los detalles internos de un dispositivo complejo tal como una computadora, un automóvil, una lavadora o un horno de microondas, etc., y usarlo como una única unidad comprensible.

“Un medio de reducir la complejidad es la abstracción. De este modo, las características complejas se hacen más manejables”. (RÍOS, 2008)

Básicamente es la capacidad de separar los elementos (al menos mentalmente) para poder verlos de forma singular.

Características de la abstracción.

- Para un mismo problema pueden existir diferentes abstracciones, dependiendo del propósito.
- Todas las abstracciones son incompletas e inexactas.
- La abstracción permite limitar el universo de manera que podamos construir modelos.

- No existen abstracciones correctas, sólo existen abstracciones adecuadas o inadecuadas.

TABLA VII.VI: EJEMPLO DIFERENTES MODELOS DE ABSTRACCIÓN

Diferentes modelos de abstracción del término coche (carro).	
CARACTERÍSTICAS:	
Partes	<ul style="list-style-type: none">• motor, carrocería, cuatro ruedas, cinco puertas, etc.
Fabricante	<ul style="list-style-type: none">• Audi, BMW, SEAT, Toyota, Nissan
categoría	<ul style="list-style-type: none">• turismo, deportivo, todoterreno
La abstracción coche se utilizará siempre que la marca, la categoría o el carburante no sean significativos. Así, un carro (coche) se utilizará para transportar personas o ir de paseo a la playa.	

Fuente: Elaboración propia

ENCAPSULAMIENTO:

La encapsulación se encarga de mantener ocultos los procesos internos que necesita para hacer lo que sea que haga, dándole al programador acceso sólo a lo que necesita.

Esto da dos ventajas iniciales: Lo que hace el usuario puede ser controlado internamente (incluso sus errores), evitando que todo colapse por una intervención indeseada. La segunda ventaja es que, al hacer que la mayor parte del código esté oculto, puedes hacer cambios y/o mejoras sin que eso afecte el modo como los usuarios van a utilizar tu código.

El diseño de un programa orientado a objetos contiene, al menos, los siguientes pasos:

1. Identificar los objetos del sistema.
2. Agrupar en clases a todos objetos que tengan características y comportamiento comunes.

3. Identificar los datos y operaciones de cada una de las clases.
4. Identificar las relaciones que pueden existir entre las clases.

En AS3 como en la mayor parte de los lenguajes orientados a objeto el encapsulado se logra definiendo los atributos propios de la clase como privados y el acceso a estos se hace a través de los métodos de acceso set y get.

TABLA VIII.VII: EJEMPLO DE ENCAPSULACIÓN

Ejemplo de encapsulación	
Celular	<ul style="list-style-type: none">• Formada internamente por circuito, placas, teclado, cámara, etc.  <p>La encapsulación de un celular permite que el usuario pueda utilizar el teléfono celular sin necesidad de manipular sus elementos internos, esto se representaría como la interfaz del usuario.</p>

Fuente: Elaboración propia

MODULARIDAD:

Consiste en organizar el programa en módulos que contienen objetos pertenecientes a determinadas clases, cuya interacción entre ellos desarrollan una tarea específica, lógica y completa. La suma de estos módulos y su correspondiente tarea que realizan, son la resolución del problema.

Permite subdividir una aplicación en partes más pequeñas, siendo cada una independiente.

TABLA VIII.IIX: EJEMPLO DE MODULARIDAD

Ejemplo de modularidad	
Juego multimedia de cocina	<ul style="list-style-type: none">• Chef• Cocina• Clientes• Mesas, etc.
Es subdividir una aplicación en partes más pequeñas siendo cada una independiente	

Fuente: Elaboración propia

JERARQUIA Y HERENCIA:

Se refieren a compartir atributos y métodos entre clases, que se relacionan de manera jerárquica.

“La herencia es la capacidad que tiene una clase de derivar las propiedades y métodos de otra” (JOYANES, y otros, 2006).

Tratemos de explicarlo con un ejemplo:

Decimos que una gallina es un ave; esto quiere decir que las gallinas tienen características comunes con otras aves (pico, plumas, etc.), es decir que la gallina hereda las características comunes de todas las aves. Pero además, resulta que un ave es un animal, lo que significa que también comparte características comunes al caballo, el perro, el hombre y cualquier otra cosa que pueda ser clasificada como animal.

La herencia nos permite, entre otras cosas, evitar tener que escribir el mismo código una y otra vez, puesto que al definir que una categoría o clase pertenece a otra, automáticamente estamos atribuyéndoles las características generales de la primera, sin tener que definir las de nuevo

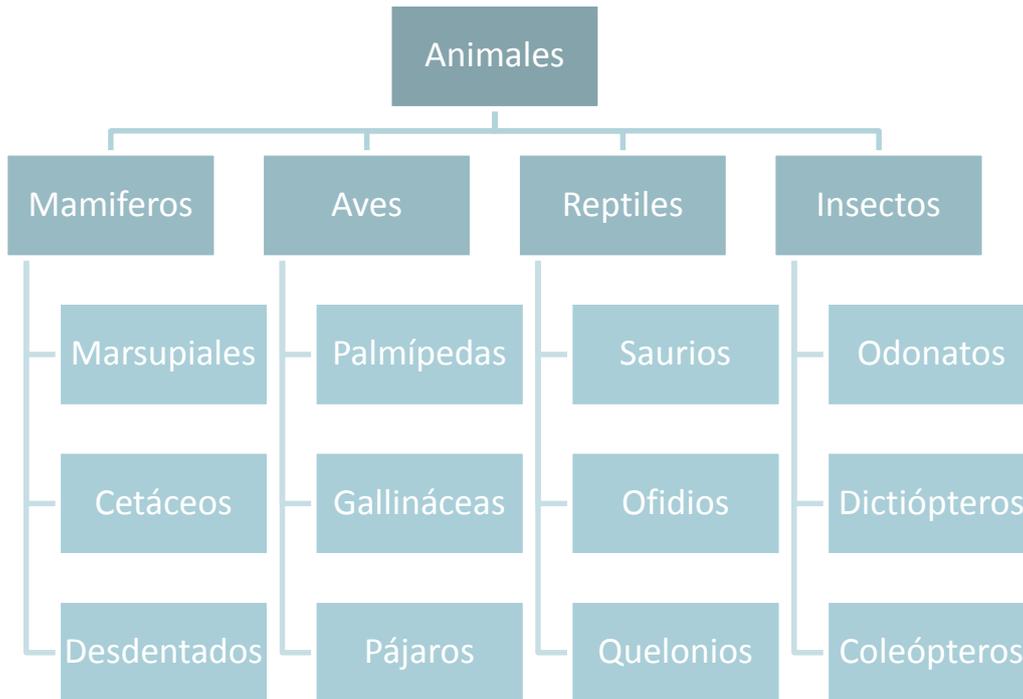


FIGURA III. 245: EJEMPLO DE JERARQUÍA

Fuente: Elaboración propia

3.5.3. CONCEPTOS DE LA POO

OBJETOS

“Un objeto es algo que se visualiza, se utiliza y juega un rol o papel. La estructura interna y el comportamiento de un objetivo, en una primera fase, no tiene prioridad. Es importante que un objeto tal como un carro o una casa jueguen un rol” (KASHIWAMOTO, y otros, 2004) .

Un carro puede ser ensamblado de partes tales como un motor, una carrocería, unas puertas o puede ser descrito utilizando propiedades tales como su velocidad, su kilometraje o su fabricante, estos atributos indican el objeto.

Un objeto no necesariamente ha de realizar algo concreto o tangible. Puede ser totalmente abstracto y también puede describir un proceso. Por ejemplo, un partido de baloncesto o de rugby puede ser descrito como un objeto. Los atributos de este

objeto pueden ser los jugadores, el entrenador, la puntuación y el tiempo transcurrido de partido.

¿Qué tipos de cosas son objetos en los programas orientados a objetos? La respuesta está limitada por su imaginación aunque se pueden agrupar en categorías típicas que facilitarán su búsqueda en la definición del problema de un modo más rápido y sencillo.

MÉTODOS

Es el procedimiento o función que se invoca para actuar sobre un objeto

ATRIBUTOS

Son variables que almacenan datos relacionados al objeto.

TABLA IXI.IX: EJEMPLO DE OBJETO

Ejemplo de objeto		
OBJETO	ATRIBUTOS	MÉTODOS
Persona	<ul style="list-style-type: none">Nombre, Dirección, cabello, altura, peso, etc.	<ul style="list-style-type: none">Respirar, alimentarse, hablar, caminar, etc.
Pelota	<ul style="list-style-type: none">Tamaño, tipo, color, textura, etc.	<ul style="list-style-type: none">Rodar, rebotar, inflarse, etc.

Dentro de un objeto residen los datos (atributos) y funciones o subrutinas (métodos)

Fuente: Elaboración propia

CLASES

“En POO los objetos son miembros de clases, en esencia, una clase es un tipo de datos al igual que cualquier otro tipo de dato definido en un lenguaje de programación. La diferencia reside en que la clase es un tipo de dato que contiene datos y funciones. Una clase contiene muchos objetos y es preciso definirla,

aunque su definición no implica creación de objetos” (KASHIWAMOTO, y otros, 2004).

Una clase es, por consiguiente, una descripción de un número de objetos similares.

Ozzy Osborne, Freddie Mercury, Paul McCartney, John Lennon son miembros u objetos de la clase “músicos de rock”. Un objeto concreto, Ozzy Osborne o Freddie Mercury, son instancias de la clase "músicos de rock".

Una clase es una descripción general de un conjunto de objetos similares. Por definición todos los objetos de una clase comparten los mismos atributos (datos) y las mismas operaciones (métodos). Una clase encapsula las abstracciones de datos y operaciones necesarias para describir una entidad u objeto del mundo real.

Así, podríamos escribir la clase ave colocando en ella todas las características que tienen las aves (pico, color, alto, ancho, patas,...) esas **características** las llamaremos en lenguaje de programadores, **propiedades**.

Pero la cosa no termina allí, resulta que las aves tienen también ciertos mecanismos específicos, como comer, dormir, reproducirse, etc. Estos **mecanismos** los llamamos **métodos**.

Por último, también sabemos que las aves reaccionan ante ciertos sucesos, como peligro, atracción, defensa... A esto lo llamaremos **eventos**.

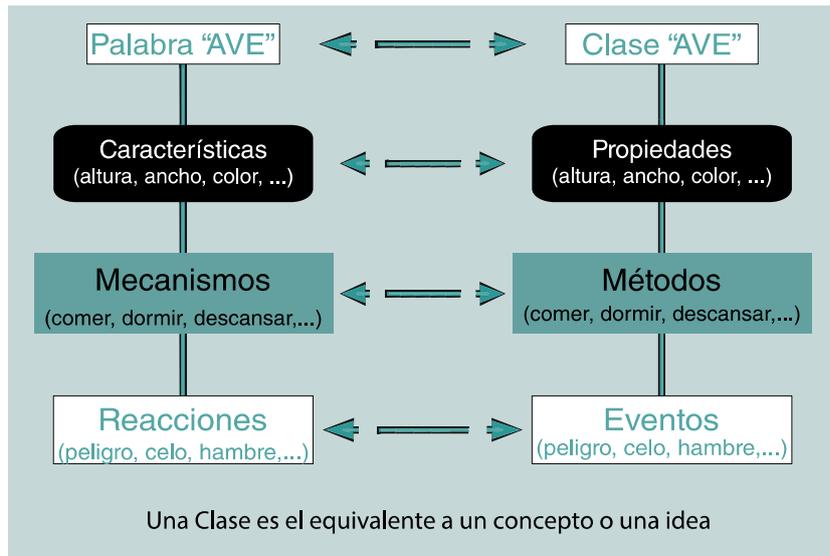


FIGURA III. 256: EJEMPLO DE CLASE

Fuente: thefricky.wordpress.com

INSTANCIA

Bien, decíamos que una clase es como la definición de un objeto, pero no es el objeto en sí, Así que para sentarnos necesitaremos convertir esa idea en algo, en un objeto real; a ese objeto lo llamamos instancia.

En un mismo proyecto puedo tener una o más instancias de una misma clase sin problemas.

Cada vez que creamos una nueva instancia, ésta adquiere las propiedades, métodos y eventos de la clase a la que pertenece (es lo que permite la relación es un), sin embargo, cada instancia es independiente de las otras; esto nos da dos ventajas:

1. Si hago algún cambio en la clase, todas las instancias de esta clase se actualizarán automáticamente; esto nos permite hacer cambios sin tener que ir a cada una de las instancias (se aplica el mismo principio de herencia, aunque a un nivel diferente).

2. Al ser independientes de las otras instancias, puedo darles valores diferentes sin que afecten a las demás (como tener una silla negra, una roja, una más alta, etc.). Aunque comparten la misma estructura, pueden programarse individualmente, dando versatilidad y flexibilidad al código.

POLIMORFISMO

La propiedad de polimorfismo es aquella en que una operación tiene el mismo nombre en diferentes clases, pero se ejecuta de diferentes formas en cada clase. Así, por ejemplo, la operación de abrir se puede dar en diferentes clases: abrir una puerta, abrir una ventana, abrir un periódico, abrir un archivo, abrir una cuenta corriente en un banco, abrir un libro, etc. En cada caso se ejecuta una operación diferente aunque tiene el mismo nombre en todos ellos "abrir". El polimorfismo es la propiedad de una operación de ser interpretada sólo por el objeto al que pertenece. Existen diferentes formas de implementar el polimorfismo y variará dependiendo del lenguaje de programación.

3.5.4. EL LENGUAJE DE PROGRAMACIÓN ACTION SCRIPT 3.0

INTRODUCCIÓN A ACTIONSCRIPT 3.0

ActionScript 3 es el lenguaje de programación que utiliza a partir de Flash CS3, hasta Flash CS6. Se trata de un lenguaje de programación orientado a objetos. Inicialmente ActionScript era un lenguaje orientado a eventos, pero poco a poco y debido al potencial de la orientación a objetos esto ha ido cambiando, sirve para crear scripts en Flash.

Gracias a ActionScript 3 podemos estructurar el código de nuestras aplicaciones de forma más clara y sencilla, facilitando la búsqueda de errores y la lectura del programa.

“ActionScript 3.0 ofrece un modelo de programación robusto que resultará familiar a los desarrolladores con conocimientos básicos sobre programación orientada a objetos, y así comunicarse con Flash”. (RICH, y otros, 2007)

VENTAJAS DE AS 3.0

Adobe entrega ahora un lenguaje completamente orientado a objetos, listo para soportar toda especie de carga, hacer largos recorridos y, bajo ciertos casos, mucho más rápido que el anterior.

ActionScript 3.0 aumenta las posibilidades de creación de scripts de las versiones anteriores de ActionScript. Se ha diseñado para facilitar la creación de aplicaciones muy complejas con conjuntos de datos voluminosos y bases de código reutilizables y orientadas a objetos. Aunque no se requiere para el contenido que se ejecuta en Adobe Flash Player 9, ActionScript 3.0 permite introducir unas mejoras de rendimiento que sólo están disponibles con AVM2, la nueva máquina virtual. El código ActionScript 3.0 puede ejecutarse con una velocidad diez veces mayor que el código ActionScript heredado. La versión anterior de la máquina virtual ActionScript (AVM1) ejecuta código ActionScript 1.0 y ActionScript 2.0. Flash Player 9 admite AVM1 por compatibilidad con contenido existente y heredado de versiones anteriores.

INTERFAZ DE FLASH CS6

El interfaz del flash cs6 está dividido en partes o paneles para la mejor disposición y organización del espacio del trabajo, los mismos que pueden ser adaptados a nuestras necesidades.

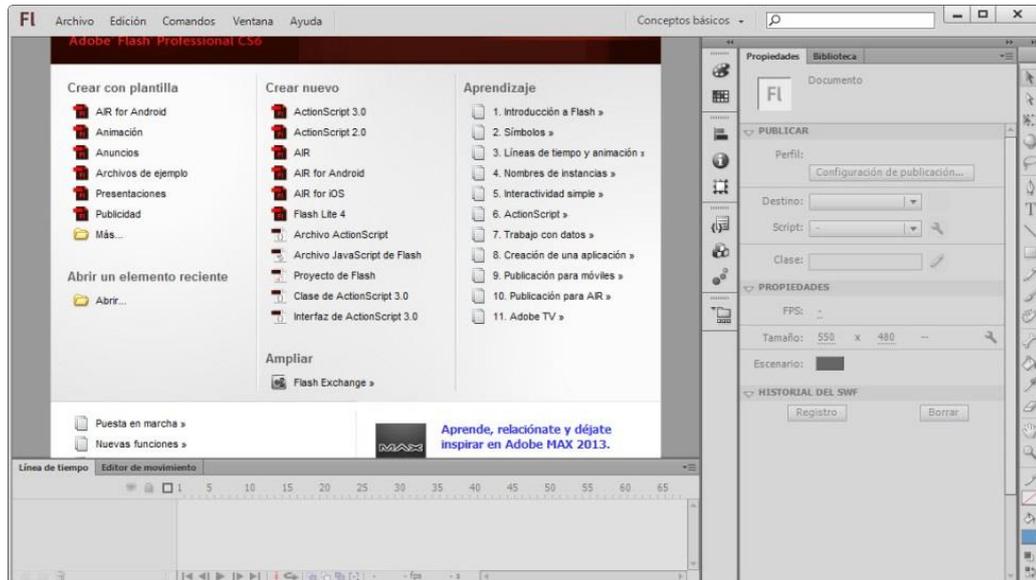


FIGURA III. 267: INTERFAZ FLASH CS6

Fuente: flash cs6

Paneles: De todas las herramientas del adobe flash cs6, es primordial saber desplegar el panel de acciones mismo que es el que permite el ingreso del código AS2.0 y AS3.0, este aparece presionando la tecla **f9** o con clic en el menú **ventana** y **acciones**.

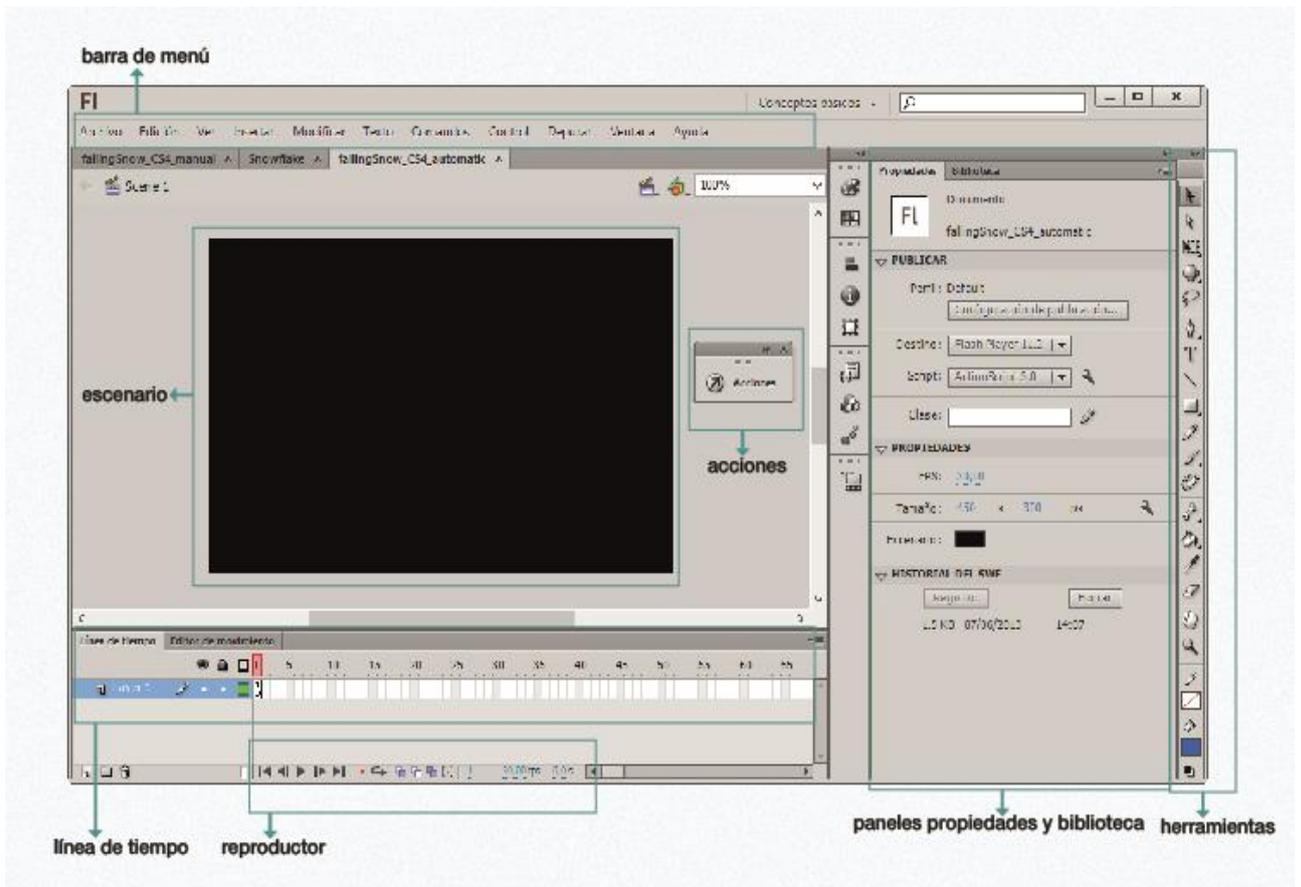


FIGURA III. 28: PANELES DE FLASH CS6

Fuente: flash cs6

También se puede empezar a escribir nuestro código AS3.0 como un archivo independiente sólo de código, abriendo un nuevo documento y a continuación elegimos **Archivo ActionScript**, de esta forma hechos abierto un nuevo archivo .as, único para programación, mismo que puede posteriormente interactuar con los elementos insertados en el escenario de un archivo de tipo .flv



FIGURA III. 29: NUEVO ARCHIVO DE TIPO ACTIONSCRIPT

Fuente: flash cs6

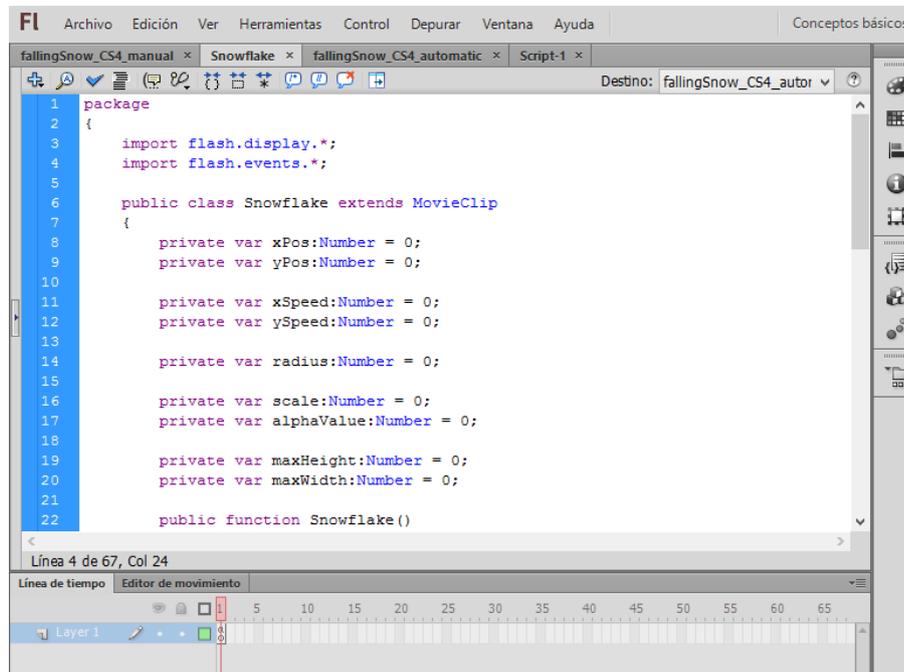


FIGURA III. 270: DOCUMENTO DE TIPO ACTIONSCRIPT

Fuente: flash cs6

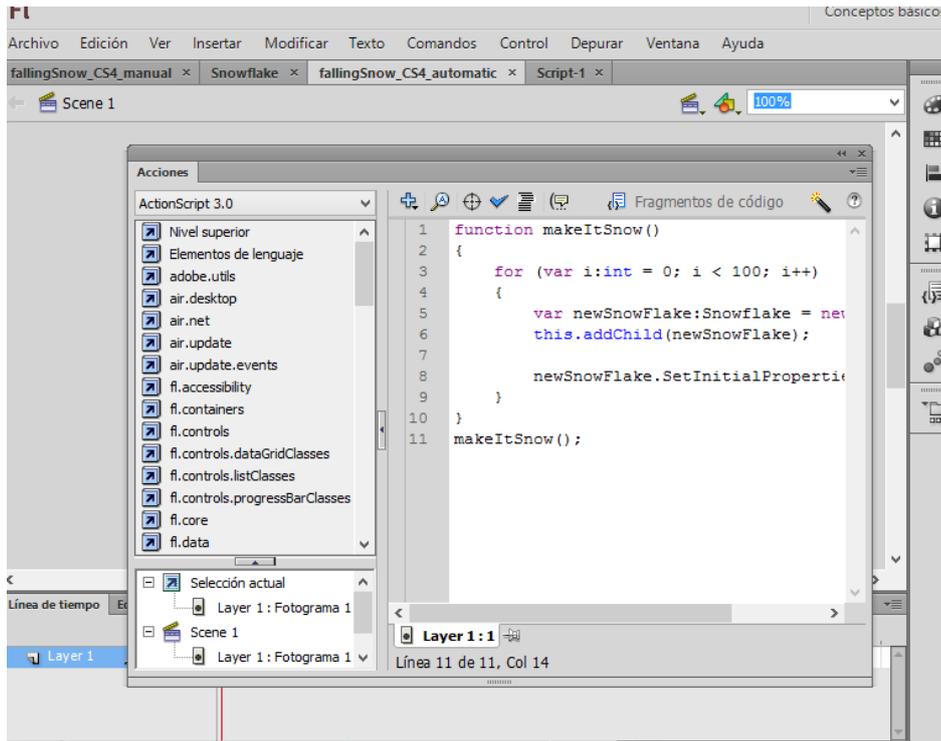


FIGURA III. 281: MANEJO DE CÓDIGO ACTIONSCRIPT3.0 EN EL PANEL DE ACCIONES
Fuente: flash cs6

SINTAXIS DE AS 3.0

Es la estructura adecuada (conjunto de reglas y símbolos), que debe cumplirse al escribir código ejecutable.

Distinción entre mayúsculas y minúsculas.

`var: Num1`  `var: num1`  son dos diferentes variables

Uso del punto y coma al final de cada sentencia:

`trace (result);`

Uso del punto como medio de acceso a las propiedades y métodos de los objetos:

`stage.addEventListener(MouseEvent.CLICK, onClick);`

Uso de paréntesis para expresar orden en las expresiones:

`num=((2+y)-5+j);`

Uso de paréntesis para paso de argumentos:

`trace=("hola");`

Uso de la coma para separar argumentos:

```
trace("hola",x);
```

Uso de comentarios (mensajes explicativos) con el fin de realizar aclaraciones de código:

```
var Num1; // variable para almacenar numero 1
```

```
var num1; /* comentario de línea múltiple,  
variable para almacenar el número 2*/
```

No usar palabras reservadas como identificadores puesto que están reservadas para el uso de ActionScript: *class, if, false, etc.*

Consultar palabras reservadas de ActionScript

Creación de valores constantes a través de la sentencia *const*

```
Const PI:number=3.1416
```

COMPONENTES DE AS3.0 EN ADOBE FLASH CS6

Los componentes de Flash se instalan durante la instalación de Flash CS3.

Los componentes ActionScript 3.0 incluyen los siguientes componentes de interfaz de usuario (IU):

TABLA XI. X: UI COMPONENTS (COMPONENTES DE INTERFÁZ)

UI Components (componentes de interfáz)		
Button	List	TextArea
CheckBox	NumericStepper	TextInput
ColorPicker	RadioButton	TileList
ComboBox	ProgressBar	UILoader
DataGrid	ScrollPane	UIScrollBar
Label	Slider	

Fuente: Adobe Systems Incorporated.

TIPOS DE DATOS DE ACTION SCRIPT 3.0

En ActionScript, hay muchos tipos de datos que pueden utilizarse como el tipo de datos de las variables que se crean.

Algunos de estos tipos de datos se pueden considerar “sencillos” o “fundamentales”:

TABLA XII. XI: TIPOS DE DATOS SENCILLOS O FUNDAMENTALES

Tipos de datos “sencillos” o “fundamentales”	
String	un valor de texto como, por ejemplo, un nombre o el texto de un capítulo de un libro
Numeric	ActionScript 3.0 incluye tres tipos de datos específicos para datos numéricos
Number	cualquier valor numérico, incluidos los valores fraccionarios o no fraccionarios
int	un entero (un número no fraccionario)
uint	un entero sin signo, es decir, que no puede ser negativo
Boolean	un valor true (verdadero) o false (falso), por ejemplo, si un conmutador está activado o si dos valores son iguales

Fuente: Aprendizaje de ACTIONSCRIPT® 3.0

El tipo de datos sencillo representa a un solo elemento de información: por ejemplo, un único número o una sola secuencia de texto. Sin embargo, la mayoría de los tipos de datos definidos en ActionScript son tipos de datos complejos. Representan un conjunto de valores en un solo contenedor, por ejemplo, una variable con el tipo de datos.

Date representa un solo valor (un momento temporal). No obstante, ese valor de fecha se representa en forma de diferentes valores: el día, el mes, el año, las horas, los minutos, los segundos, etc., los cuales son números individuales.

Sin embargo, internamente el equipo lo considera un grupo de varios valores que conjuntamente definen una sola fecha.

Algunos de los tipos de datos complejos que podrían reconocerse son:

TABLA XIII. XII: TIPOS DE DATOS COMPLEJOS

Tipos de datos complejos	
MovieClip	un símbolo de clip de película
TextField	un campo de texto dinámico o de texto de entrada
SimpleButton	un símbolo de botón
Date	información sobre un solo momento temporal (una fecha y hora)

Fuente: Aprendizaje de ACTIONSCRIPT® 3.0

Para referirse a los tipos de datos, a menudo se emplean como sinónimos las palabras clase y objeto. Una clase es simplemente una definición de un tipo de datos. Por otra parte, un objeto es una instancia real de una clase. Por ejemplo, una variable cuyo tipo de datos sea MovieClip se podrá describir como un objeto MovieClip. Se puede decir lo mismo con distintos enunciados:

- El tipo de datos de la variable myVariable es Number.
- La variable myVariable es una instancia de Number.
- La variable myVariable es un objeto Number.
- La variable myVariable es una instancia de la clase Number

OPERADORES

Son funciones especiales que se aplican a uno o más operandos y devuelven un valor.

Se clasifican en unarios, binarios y ternarios en función al número de operandos que intervienen.

TABLA XIII. XIII: OPERADORES

Operadores	Tipo de operadores
* / % + -	Aritméticos
> < = <= >= == != in is	Relacionales
&&	Lógicos
++ -- - +	Unarios

Fuente: Aprendizaje de ACTIONSCRIPT® 3.0

3.5.5. CREAR Y USAR VARIABLES

VARIABLES

Una variable es un nombre que representa un valor en la memoria del equipo.

Cuando se escriben sentencias para manipular valores, se escribe el nombre de la variable en lugar del valor. Ejemplo:

TABLA XVI. XIV: VARIABLES

Variables	Sentencia
S, N1 y N2	S=N1+N2;
Cuando se ejecute la sentencia, el equipo buscará los valores de cada variable y los sumará.	

Fuente: Elaboración propia

En ActionScript 3.0, una variable se compone realmente de tres partes distintas:

- El nombre de la variable.
- El tipo de datos que puede almacenarse en la variable.
- El valor real almacenado en la memoria del equipo.
- Para crear una variable (*declarar*), se utiliza la sentencia var:

```
Var nombre_variable:Number=16;
```

Para crear una instancia de una clase hay que utilizar el operador **new**.

En el ejemplo siguiente se crea una instancia de una clase denominada *CustomClass* y se asigna una referencia a la instancia de clase recién creada a la variable denominada *Itemdecustom*:

```
var Itemdecustom:CustomClass = new CustomClass();
```

Si hubiera que declarar más de una variable, se pueden declarar todas en una línea de código utilizando el operador coma (,) para separar las variables. Por ejemplo, el código siguiente declara tres variables en una línea de código:

```
var a:int, b:int, c:int;
```

También se puede asignar valores a cada una de las variables en la misma línea de código. Por ejemplo, el código siguiente declara tres variables (a, b y c) y asigna un valor a cada una de ellas:

```
var a:int=10, b:int=20, c:int=30;
```

Aunque se puede utilizar el operador coma para agrupar declaraciones de variables en una sentencia, al hacerlo el código será menos legible.

También se utiliza la técnica de almacenar valores al momento de declarar una variable en matrices o en instancias de una clase. Ejm.

Variable en matrices

```
var NumArray:Array=[" zero","one","two"];
```

Instancia de una clase

```
var objeto1:miClase = new miClase();
```

Si una variable no es inicializada, asume un valor por default en función del tipo de dato al que pertenece:

TABLA XVI. XV: TIPOS DE DATOS DE VALORES SIMPLES CON SUS VALORES PREDETERMINADOS

Tipos de datos de valores simples con sus valores predeterminados		
Tipo de datos	Valor	Valor predeterminado
Boolean	True/false	false
int	Valores enteros de 32 bits	0
Number	Números de coma flotante de 64 bits valores enteros de 53 bits	NaN
Object	La clase Object constituye la clase base para todas las definiciones de clase en ActionScript.	null
String	“cadena de caracteres”	null
uint	0 a 2.147.483,647 (321-1)	0
No declarada		undefined
Todas las demás clases, incluidas las clases definidas por el usuario.		null

Fuente: Aprendizaje de ACTIONSCRIPT® 3.0

TABLA XVII. XVI: TIPOS DE DATOS DE VALOR COMPLEJO

Tipos de datos de valor complejo	
Tipo de datos	Valor
Array	permite obtener acceso a matrices y manipularlas [“José”, “Pedro”, “Juan”] o [10,11,12,13]
Date	permite recuperar los valores de fecha y hora relativos a la hora universal <code>Date</code> (yearOrTimevalue: Object , month: Number , date: Number = 1, hour: Number = 0, minute: Number = 0, second: Number = 0, millisecond: Number = 0)
Xml	tiene métodos y propiedades para trabajar con objetos XML XML (value: Object)
XmlList	Un objeto XMLList puede representar uno o varios objetos o elementos XML (incluidos varios nodos o atributos). XMLList (value: Object)
Void	Especifica que una función no puede devolver cualquier valor. functionName():void {}

Fuente: <http://livedocs.adobe.com/flash>

ASPECTOS BÁSICOS DEL ÁMBITO DE VARIABLES

“Variable global está definida en todas las áreas del código, mientras que una variable local sólo está definida en una parte del código” (BRAUNSTEIN, 2006).

En ActionScript 3.0, las variables siempre se asignan al ámbito de la función o la clase en la que se han declarado.

Variable global: Es una variable que se define fuera de una definición de función o clase. Por ejemplo, el código siguiente crea una variable global strGlobal declarándola fuera de las funciones. El ejemplo muestra que una variable global está disponible tanto dentro como fuera de una definición de función.

```
var strGlobal:String = "Global";  
  
function scopeTest()  
{  
    trace(strGlobal); // Global  
}  
  
scopeTest();  
  
trace(strGlobal); // Global
```

Variable local: Se declara dentro de una definición de función. La parte más pequeña de código para la que se puede definir una variable local es una definición de función. Una variable local declarada en una función sólo existirá en esa función. Por ejemplo, si se declara una variable denominada str2 dentro de

una función denominada `localScope()`, dicha variable no estará disponible fuera de la función.

```
function localScope()  
{ var strLocal:String = "local"; }  
  
localScope();  
  
trace(strLocal); // error porque strLocal no está definida  
como global.
```

3.5.6. USAR CONDICIONALES

“Permiten tomar decisiones lógicas, además especificar los términos bajo los cuales una sesión o porción del código debe o no debe ser ejecutada, es una declaración que ejecuta una acción sólo cuando una condición específica se cumple” (SANTOS, s.f).

Ejemplo:

Situación: Ver el contenido de una película flash protegida con contraseña.

Condición: ¿Tiene el usuario la contraseña?

Opción 1: **SI**, Se muestra el contenido de la película.

Opción 2: **NO**, No se puede mostrar el contenido de la película.

AS3 proporciona tres sentencias condicionales básicas:

- La declaración *If*
- La declaración *If...else*
- La declaración *Case o Switch*

CONDICIONAL- DECLARACIÓN IF: Cuando una declaración If es ejecutada, el intérprete comprueba o evalúa la condición. Si el resultado es **True** las sub-declaraciones son ejecutadas.

Cuando se llega a la última sub-declaración, se pasa a la siguiente instrucción fuera del If

Sintaxis del condicional if

```
If (condición) {  
    //Entonces se ejecuta este código  
}
```

Ejemplo:

```
var num;  
num=(Math.random()*1000);  
trace (num);  
if (num <100) {  
    trace ("Resultado menor que 100")  
}
```

Genera un número aleatorio entre 0 a 1000 y pregunta si es menor que 100 si es así mostrar "resultado menor que 100"

FIGURA III. 292: EJEMPLO ESTRUCTURA IF

Fuente: <http://livedocs.adobe.com/flash>

CONDICIONAL- DECLARACIÓN IF... ELSE: Cuando una declaración If es ejecutada, el intérprete comprueba o evalúa la condición. Si el resultado es **True** las sub-declaraciones dentro de If son ejecutadas, si el resultado es **False** las sub-declaraciones de else son ejecutadas.

Sintaxis del condicional if...else

```
If (condición) {  
    /*Si se cumple la condición,  
    se ejecuta este código*/  
}  
else {  
    /*Si NO se cumple la condición,  
    se ejecuta este código*/  
}
```

Ejemplo:

```
var num;  
num=(Math.random()*1000);  
trace (num);  
if (num <100) {  
    trace ("Resultado menor que 100")  
}  
else {  
    trace ("Resultado mayor que 100")  
}
```

Genera un número aleatorio entre 0 a 1000 y pregunta si es menor que 100 si es así mostrar "resultado menor que 100" y sino "mostrar resultado mayor que 100"

FIGURA III. 303: EJEMPLO ESTRUCTURA IF...ELSE,
Fuente: Elaboración propia.

CONDICIONAL: DECLARACIÓN CASE O SWITCH: Cuando existe varios hilos de ejecución que dependen de la misma expresión de condición.

La sentencia switch evalúa una expresión y el resultado utiliza para determinar el bloque de código que debe ejecutarse.

Los bloques de código empiezan con la palabra case y termina con brake.

Sintaxis del condicional switch

```
Var dia:int=1;  
switch(dia)  
{  
    case 0:  
        trace("Domingo");  
        break;  
    case 1:  
        trace("Lunes");  
        Break;  
    default:  
        trace("no existe");  
        break;  
}
```

Ejemplo:

```
var Azar:uint=Math.random()*7;
trace (Azar)
switch(Azar>=0 && Azar<=9)
{
  case Azar==1:
    miclip.gotoAndPlay(1);
    break;
  case Azar==2:
    miclip.gotoAndPlay(11);
    break;
  case Azar==3:
    miclip.gotoAndPlay(31);
    break;
}
```

Genera un número aleatorio entre 0 a 7 y manda a reproducir cierto fotograma en cada uno de los casos.

FIGURA III. 314: EJEMPLO ESTRUCTURA SWITCH

Fuente: Elaboración propia.

3.5.7. USAR LOOPS (CICLOS)

Llamados también Ciclos o Bucles, sirven para ejecutar acciones o tareas repetitivas o incluso tediosas, suponga que deseamos imprimir la serie de números del 1 al 5.

```
1 trace (1);
2 trace (2);
3 trace (3);
4 trace (4);
5 trace (5);
```

FIGURA III. 325: IMPESIÓN DE 5 NÚMEROS

Fuente: Elaboración propia.

¿Qué sucede si necesitamos imprimir la serie de 1 al 500?

Los loops permiten ejecutar un bloque de código tantas veces o tanto tiempo mientras la condición de evaluación se mantenga en **true**

Existen varios tipos de loops:

- Ciclo *while*
- Ciclo *do-while*
- Ciclo *for*
- Ciclo *for -in*

BUCLE - CICLO WHILE: Hacer mientras cumpla la condición, garantiza que bloque de código se repita al menos una vez, a diferencia del condicional If, cuando se llega a la última subdeclaración, el bloque inicia nuevamente, El proceso termina cuando el resultado de la condición se convierte en *False*.

En resumen un loop while contiene 3 partes básicas:

- **Inicialización:** Declaración de variables que se usarán en la expresión de comprobación.
- **Expresión de comprobación:** Condición que debe cumplirse para ejecutar el bloque de subdeclaraciones
- **Actualización:** Modificación de las variables usadas en la expresión de comprobación antes de la siguiente iteración

Sintaxis del ciclo While

```
do
{
sentencias a
ejecutar y/o repetir
}while(condición)
```

Ejemplo:

```
var num=0;
while (num<=500){
    trace (num);
    num=num+1;
}
```

Genera un ciclo de impresión de 0 a 500 inicializado en una variable num y sumando un dígito dentro de la estructura while

FIGURA III. 336: EJEMPLO ESTRUCTURADO WHILE

Fuente: Elaboración propia.

BUCLE: CICLO DO-WHILE: Cuando se utiliza un loop while, puede ocurrir que el bloque de repetición nunca se ejecute, cuando la condición no sea evaluada como true, una declaración do-while, garantiza que el cuerpo del loop o bloque de repetición se ejecute al menos una vez, siendo su estructura es como una declaración while invertida.

En un loop do-while las subdeclaraciones son ejecutadas antes de que la condición sea comprobada.

Sintaxis del ciclo do-while

```
do {
    /*Sub-declaraciones
    o bloque que se desea repetir*/
} while ( condición );
```

Ejemplo:

```
1 var i=1;
2 do {
3 trace (i);
4 i++;
5 }
6 while (i<=500);
7
```

Generar la serie de números de 1

FIGURA III. 347: EJEMPLO ESTRUCTURA DO-WHILE

Fuente: Elaboración propia.

BUCLE: CICLO FOR: Es un sinónimo del loop while, pero con una sintaxis más compacta, incluye en una misma línea: la inicialización, condición y actualización, separando los componentes de la cabecera con punto y coma (;)

Permite repetir una variable para un rango de valores específicos.

Requiere tres expresiones:

- Una variable que determina el valor inicial
- Una sentencia condicional que determina cuando termina el bucle.
- Una expresión que cambia el valor de la variable con cada bucle.

Sintaxis del ciclo do-while

```
for (inicialización; condición; actualización)
{
// subdeclaraciones o sentencias a ejecutar
}
```

Ejemplo:

```
for (var i=1; i<=500; i++){
    trace (i)
}
```

Generar la serie de números de 1 a 500

FIGURA III. 38: EJEMPLO ESTRUCTURA FOR

Fuente: Elaboración propia.

3.5.8. CREAR Y USAR FUNCIONES

Las funciones son bloques de código reutilizables.

Son una serie de instrucciones o expresiones compactadas que ejecutan alguna acción o algún cálculo, estos bloques de código son ejecutados tras el llamado a la función. La misma que posee un nombre que la identifica.

Son bloques de código que realizan tareas específicas y pueden ser reutilizadas.

Existen dos tipos en AS3 (dependido del contexto):

- Métodos;
- Cierres de Función

INVOCACIÓN DE FUNCIONES

Se utiliza su identificador seguido del operador paréntesis, el operador paréntesis se utiliza para escribir parámetros de función.

Ejm. `trace("me permite depurar scripts");`

Si se llama a una función sin parámetros se utiliza parámetros vacíos.

```
Var randomNum:Number=Math.random();
```

Para utilizar una función en *ActionScript 3*, antes es necesario definirla utilizando la palabra clave ***function***.

Empieza con la palabra clave `function` seguido de:

- Nombre de la función
- Los parámetros, en una lista delimitados por comas y cerrados en `()`.
- El cuerpo de la función escrito entre llaves.

Se debe utilizar la sentencia `return` seguido de la expresión o literal que se va devolver, la sentencia `return` finaliza la función, se debe devolver un valor de tipo apropiado para no generar error.

Sintaxis básica de una función:

```
Function nombre_función(lista de argumentos):void {  
  // bloque de código  
  return (valor que devuelve)  
}
```

Las funciones también poseen un identificador o nombre. Éste identificador puede contener letras de la “a” a la “z” (mayúsculas y minúsculas), números del 0 al 9, barra baja (_) y símbolo de dólar (\$). Habitualmente el nombre de la función se define con la primera letra de cada palabra en Mayúscula excepto la primera letra que va en minúsculas, sin dejar espacios entre palabras.

TERMINOLOGÍA DE LAS FUNCIONES

- **Declaración o Definición de funciones:** Significa crear funciones para su futuro uso en un script
- **Invocación o llamado de una función:** Dan lugar a que la función se ejecute. Activan el código de la función
- **Argumentos y parámetros de una función:** Proveen la información o datos que serán manipulados dentro de la función.
- **Término de una función:** Finaliza la ejecución de la función y opcionalmente devuelve el resultado.
- **Alcance de una función:** Determina la disponibilidad y duración de la función y la accesibilidad de las variables incluidas en ella.

Las funciones pueden ser invocadas en cualquier lugar dentro de la película.

Ejemplo:

```
//CON FUNCIONES
function area(largo, ancho)
{
    return(largo*ancho);
}

var t1=area(3,2);
trace ("El área es: ",t1);
var t2=area(3,5);
trace ("El área es: ",t2);
var t3=area(5,2);
trace ("El área es: ",t3);
```

Script que calcule el área de tres rectángulos

FIGURA III. 39: EJEMPLO FUNCIONES PERSONALIZADAS

Fuente: Elaboración propia.

Además de las funciones personalizadas existen las funciones que llegan ya integradas al software de Adobe Flash en Action Script 3.0.

Toda función además de ejecutar código, puede entregar un resultado, el cual es recibido al invocar la función, el resultado se envía con la palabra clave return, y puede ser recibido en una variable, o puede ser utilizado y/o mostrado en pantalla inmediatamente.

Ejemplo:

```
function suma (num1, num2)
{
    var s;
    s=num1+num2;
    return(s)
}

/////recibir el resultado en una variable:
var s1=suma(4,5);
trace (s1);

/////imprimir directamente el resultado
trace (suma(4,5));
```

Función que realiza una suma entregando resultado en una variable y directamente

FIGURA III. 40: EJEMPLO FUNCIONES ENTREGA DE RESULTADOS

Fuente: Elaboración propia.

FUNCIONES INTEGRADAS EN ACTIONSCRIPT

Llamadas funciones internas, intrínsecas, son aquellas funciones pre-construidas, listas para ser utilizadas, su comportamiento es similar a las funciones creadas por el usuario, son invocadas de la misma manera, a través de su nombre y la lista de parámetros de ejecución.

Ejemplo: trace (“Hola”)

Existe una multitud de este tipo de funciones en ActionScript, las cuales sirven para manejar o controlar los distintos objetos:

Entre esos se pueden encontrar funciones de control de sonidos, funciones de control de texto, funciones de control de botones, funciones de control de películas y fotogramas en línea de tiempo a continuación trataremos dos funciones integradas básicas para comprender su funcionamiento.

gotoAndPlay ()

- Permite saltar a un fotograma o a una escena específicos de la película.
- Cuando la película salta a un fotograma, se reproduce a partir del nuevo fotograma

La película también puede saltar a una escena y reproducir un fotograma especificado o el primer fotograma de la escena siguiente o de la anterior.

Posee dos parámetros fundamentales:

- El número de fotograma al que se desea saltar.
- La escena en la que se encuentra el fotograma deseado.

Si no se especifica la escena, se asume que será la misma escena del fotograma que ejecuta la función.

```
gotoAndPlay (10, "Escena 2")
```

gotoAndStop ()

Permite saltar a un fotograma o a una escena específicos de la película y detener la ejecución de la película.

Su uso es similar al de la función `gotoAndPlay ()`

3.5.9. GESTIÓN DE EVENTOS

Toda la interacción entre el usuario y el archivo SWF se considera un evento, los eventos también pueden producirse sin interacción directa con el usuario, los eventos pueden venir desencadenados por el usuario o por la propia ejecución del programa

Ejemplo:

- Cuando se termina de cargar los datos desde un servidor.
- Cuando se activa una cámara conectada
- Se reproduce un fotograma de un clip de película
- Se consume el tiempo de un temporizador
- El usuario pulsa el botón izquierdo del ratón
- El usuario pulsa una tecla del teclado
- El usuario mueve el ratón

La creación de código AS, permite definir el comportamiento de las películas y los objetos inmersos en ella, ya sea a través de instrucciones básicas, Loops, condiciones, funciones, etc, ActionScript no solamente ejecuta esas instrucciones, sino que permanece atento esperando que algo ocurra, ese algo puede estar determinado por el usuario o por acciones propias de la reproducción de la película.

En AS3.0 cada objeto se representa como un objeto de evento (instancia de la clase Event). Los objetos de eventos no contienen información sino también contienen métodos.

Ejemplo:

Cuando flash player detecta un click del ratón, crea un objeto de evento ósea una instancia de la Clase MouseEvent, para representar ese evento de click.

Se puede usar detectores de eventos (funciones o métodos para responder a los distintos eventos)en el código para detectar los objetos de eventos.

Sintaxis Básica De Un Evento:

```
function responderEvento(eventObject:EventType):void
{
//acciones como respuesta al evento
}
origenEvento.addEventListener(EventType.EVENT_NAME, responderEvento)
```

EJECUCIÓN DE EVENTOS

El código AS, no se ejecuta automáticamente o por decisión propia, sino que siempre hay algo que provoca su ejecución, ya sea:

- Por la reproducción sincronizada de una película.
- Por algún evento asincrónico definido por el programador.

Ejecución sincrónica de código: El transmisor debe coordinarse con el receptor antes del envío de datos.

Ejemplo:

- En un nuevo archivo de AS3
- En el fotograma 1 colocamos el código:

```
trace ("Estoy en el fotograma 1");
```

- En el fotograma 10 colocamos el código:

```
trace ("Estoy en el fotograma 10");
```

El intérprete ejecuta el código de él fotograma 1 y la cabeza lectora continúa recorriendo los fotogramas de manera lineal y predecible, cuando llega al fotograma 10 ejecuta el código y repite la película.

Ejecución asincrónica de código: La ejecución asincrónica de código no se realiza en una secuencia determinada, sino en respuesta a algún evento o a alguna acción del usuario:

- Al presionar una tecla,
- Al mover el mouse
- Al redimensionar el tamaño de la película, etc.

La ejecución asincrónica de código, implica que se puede realizar en cualquier momento.

Esto permite hacer películas con mayor dinamismo, a través de la programación de respuesta a *EVENTOS*

Cuando el usuario hace clic en un objeto Botón, por ejemplo, distribuye un evento `MouseEvent.CLICK`.

Cuando termina de cargarse el contenido de una instancia de `UI Loader` y se genera un evento

`Event.COMPLETE`.

TIPOS DE EVENTOS

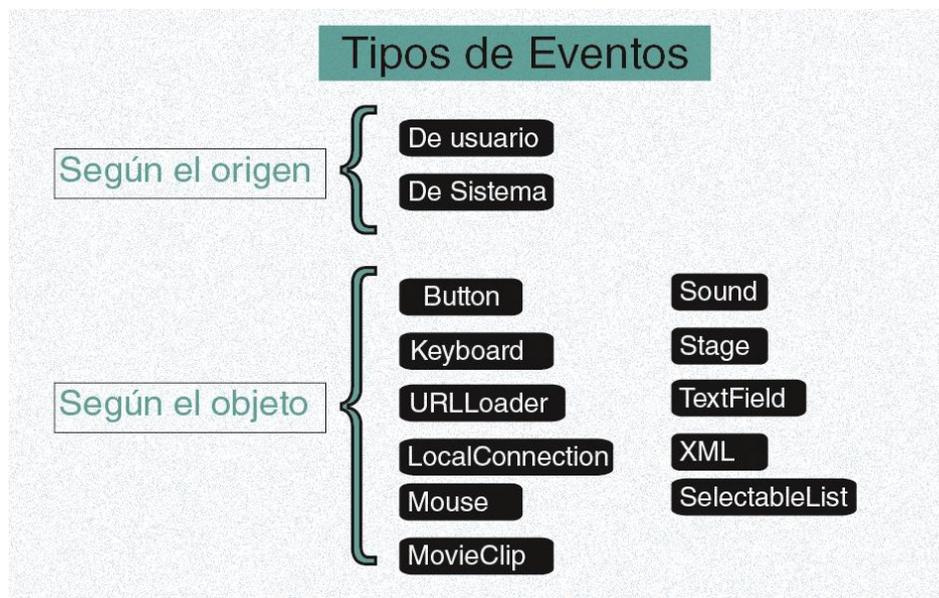


FIGURA III. 41: TIPOS DE EVENTOS

Fuente: Elaboración propia

GESTIÓN DE EVENTOS:

Es la técnica para especificar determinadas acciones que deben realizarse como respuesta a eventos concretos.

Se realiza a través de gestores de eventos, llamados también manejadores o Handler.

Un evento por sí solo no desencadena la ejecución de código, sino que se debe escribir el código que hará que se reaccione ante algún evento (se debe crear el gestor de eventos).

Ejemplo:

- Crear un botón en el escenario
- Ejecutar la película y probar los eventos

El gestor de eventos maneja o controla las acciones en respuesta a los eventos.

Un gestor de eventos permite invocar una función cuando un evento concreto ocurre.

Cuando se escribe el código ActionScript para llevar a cabo la gestión de eventos, se deben identificar tres elementos importantes:

1. El origen del evento.
2. El evento.
3. La respuesta al evento

El origen del evento:

Por ejemplo, ¿en qué botón se hará clic, o qué objeto Loader está cargando la imagen?.

El origen del evento también se denomina objetivo del evento, ya que es el objeto al que Flash Player (donde tiene lugar realmente el evento) destina el evento.

El evento:

¿Qué va a suceder, a qué se va a responder?

Es importante identificar esto porque muchos objetos activan varios eventos.

Ejemplo:

- CLICK,
- DOUBLE_CLICK,
- MOUSE_OVER, etc

La respuesta:

Se refiere a los pasos hay que seguir cuando ocurra el evento.

Sintaxis básica de Gestores de Eventos:

```
objetoOrigen.addEventListener(TipoEvento.NOMBRE_EVENTO,  
respuestaEv) ;  
  
function respuestaEv(variableEvento:TipoEvento):void  
{  
    // Aquí deben incluirse las acciones a realizar en  
    //respuesta al evento.  
}
```

3.5.10. CLASES Y OBJETOS

Una **clase** es una estructura de programación que nos sirve para declarar o definir las características y funcionalidades de un determinado conjunto de elementos.

Un **objeto** es una representación detallada, concreta y particular de un "algo", de una Clase. La instanciación de una clase da origen a un objeto. También es común llamar a un objeto instancia.

En resumen:

- **Objeto:** es una estructura que agrupa propiedades y métodos.
- **Clase:** es una categoría genérica de objetos que comparte una serie de métodos y propiedades comunes.
- **Instancia:** Es una caso específico de esa clase de objeto.

En Flash una clase personalizada se crea en un archivo de ActionScript AS. Hay que crear un archivo AS para cada clase. Así el código está mejor estructurado y se puede reutilizar con más facilidad. Para crear un archivo AS vamos al menú Archivo>Nuevo y elegimos Archivo de ActionScript.

El archivo AS recibe el mismo nombre que la clase que contiene.

Por ejemplo, dentro del archivo NombreDeLaClase.as definiremos la clase de esta manera:

```
//Dentro del archivo NombreDeLaClase.as

package{
    class NombreDeLaClase{
        //Contenido de la clase...
    }
}
```

Package es una palabra reservada que indica que la clase estará comprendida dentro de un paquete, un paquete no es más que un contenedor de clases.

Habitualmente el nombre de la clase se define con la primera letra de cada palabra en mayúscula y sin dejar espacios entre palabras.

INSTANCIA DE UNA CLASE

Por tanto la Clase es como si fuera la definición general del Objeto, sería como un patrón o un plano del objeto. El Objeto, por otro lado, es la realización de un elemento de una Clase.

Para instanciar una clase lo haremos con la palabra clave `new` de esta forma:

```
var objeto:NombreDeLaClase = new NombreDeLaClase();
```

La instanciación del archivo se puede realizar dentro de un archivo AS o dentro de un archivo FLA.

Por lo que acabamos de decir, la instancia de una clase es un objeto. Los objetos o instancias de una clase tienen una serie de características que podríamos separar entre datos y comportamiento.

VINCULACIÓN DE CLASES A LA PELÍCULA

Existe la posibilidad de vincular clases a determinados contenidos multimedia de la película Flash. La vinculación de clases resulta muy útil puesto que nos permite crear Clips de película más complejos, y de esta manera un clip de película puede convertirse en Menú con botones, en Personaje de un juego, una animación interactiva, etc.

Clase de documento (vinculada al archivo FLA de la película)

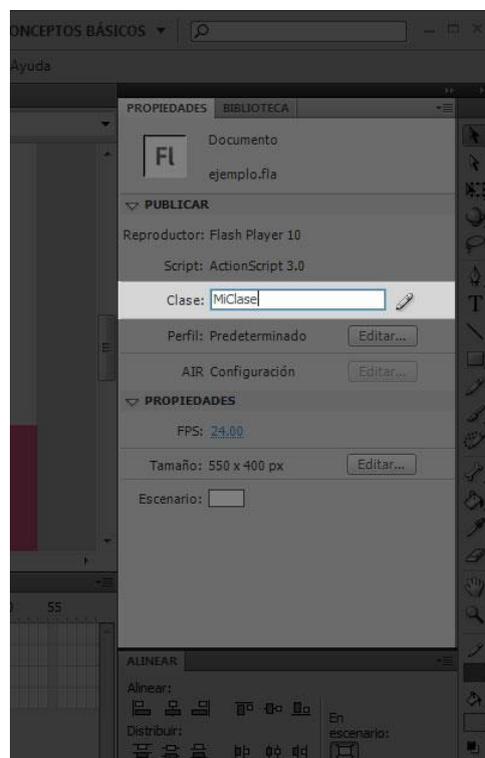


FIGURA III. 352: VINCULACIÓN DE CLASES A LA PELÍCULA
Fuente: <http://circuloycuadrado.com>

Vinculación a Símbolos (contenidos en la biblioteca)

Desde la biblioteca también se pueden asociar clases a Símbolos. Gracias a esto podemos hacer que cualquier elemento de la biblioteca tenga asociada una clase y descienda de esta. De esta manera el símbolo poseerá sus **métodos** y **propiedades**.

Para asociar una clase a un símbolo, lo hacemos desde la biblioteca. Seleccionamos el símbolo y clic derecho sobre él seleccionar del menú desplegable la opción convertir en símbolo. Nos aparece una ventana, seleccionamos el botón Avanzado (si es que estamos en el modo básico), y finalmente en el apartado de vinculación pulsamos sobre la opción: Exportar para ActionScript. En el campo clase (o sino en el campo clase base) indicamos el nombre de la clase que queremos asociar.

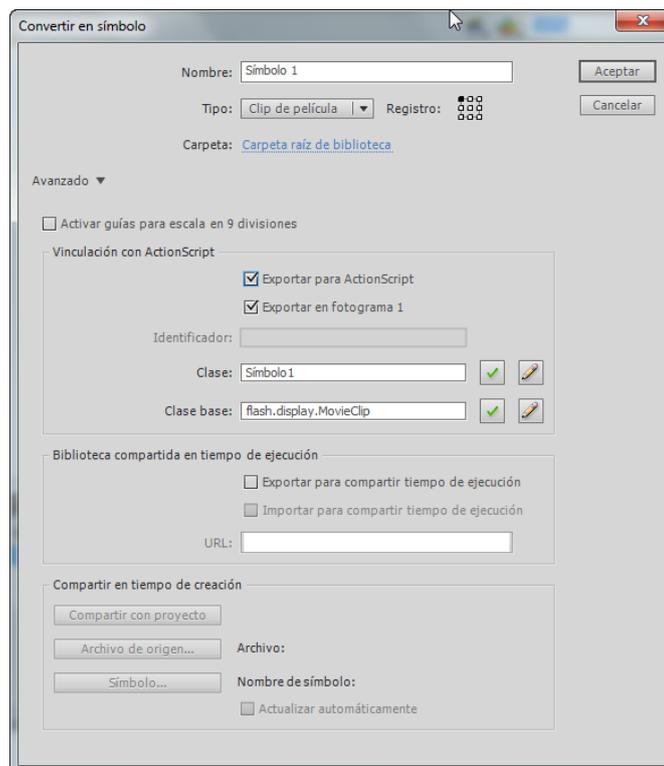


FIGURA III. 363: CONVERTIR EN SÍMBOLO

Fuente: Adobe Flash CS6

3.5.11. OBJETOS DE VISUALIZACIÓN Y CLIPS DE PELÍCULA

Los elementos que aparecen en pantalla en una película *Flash* son **objetos de visualización**. Estos objetos son elementos que tienen una determinada posición y situación en el espacio bidimensional de la pantalla. *Flash* dispone de un variado conjunto de elementos de visualización dotados de diferentes características que nos va a permitir mostrar los diferentes elementos visuales utilizados en las películas *Flash* (CHURA, 2012). Uno de los objetos de visualización que utiliza Flash son los clips de película.

CLASES PRINCIPALES DE VISUALIZACIÓN

El paquete `flash.display` de ActionScript 3.0 contiene clases de objetos visuales que pueden aparecer en Flash Player. En la siguiente ilustración se muestran las relaciones entre las subclases de estas clases principales de objetos de visualización.

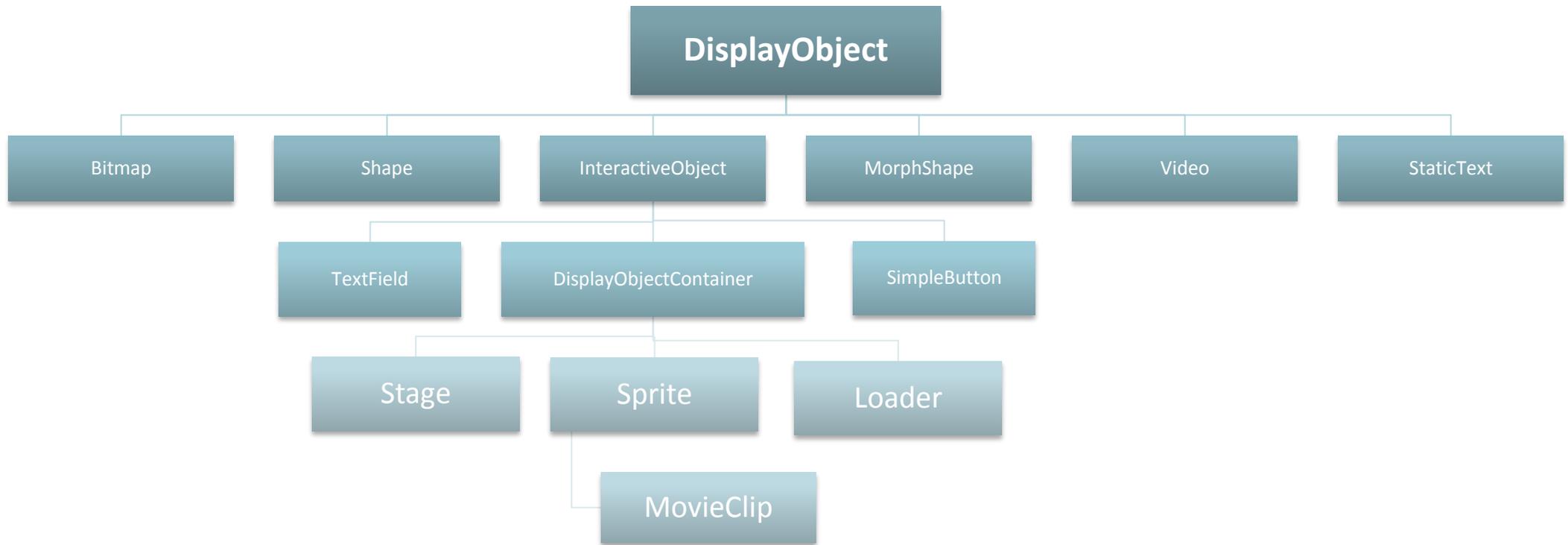


FIGURA III. 37: JERARQUÍA DE CLASES DE VISUALIZACIÓN

Fuente: www.sribd.com

Todas las clases que amplían la clase `DisplayObject` heredan sus métodos y propiedades. Pueden crearse instancias de objetos de las siguientes clases contenidas en el paquete `flash.display`:

- **Bitmap:** la clase `Bitmap` se usa para definir objetos de mapa de bits cargados de archivos externos o representados a través de `ActionScript`. Para cargar mapas de bits de archivos externos, se puede usar la clase `Loader`. Se pueden cargar archivos GIF, JPG o PNG. También es posible crear un objeto `BitmapData` con datos personalizados y crear a continuación un objeto `Bitmap` que utilice dichos datos. Se pueden usar los métodos de la clase `BitmapData` para modificar mapas de bits, tanto si se cargan como si se crean en `ActionScript`.
- **Loader:** la clase `Loader` se usa para cargar activos externos (archivos SWF o gráficos).
- **Shape:** la clase `Shape` se usa para crear gráficos vectoriales como rectángulos, líneas, círculos, etc.
- **SimpleButton:** un objeto `SimpleButton` es la representación en `ActionScript` de un símbolo de botón de Flash. Una instancia de `SimpleButton` tiene tres estados de botón: Arriba, Abajo y Sobre.
- **Sprite:** un objeto `Sprite` puede contener gráficos propios y también objetos de visualización secundarios. La clase `Sprite` amplía la clase `DisplayObjectContainer`.

- **MovieClip:** un objeto MovieClip es la representación en ActionScript de un símbolo de clip de película creado en la herramienta de edición de Flash. En la práctica, un MovieClip es similar a un objeto Sprite, excepto en que tiene además una línea de tiempo

Las siguientes clases, que no se encuentran en el paquete flash.display, son subclases de la clase DisplayObject:

- La clase **TextField**, incluida en el paquete flash.text, es un objeto de visualización para mostrar e introducir texto.
- La clase **Video**, incluida en el paquete flash.media, es el objeto de visualización que se utiliza para mostrar archivos de vídeo.

Las siguientes clases del paquete flash.display amplían la clase DisplayObject, pero no es posible crear instancias de las mismas. En lugar de eso, actúan como clases principales de otros objetos de visualización y combinan la funcionalidad común en una sola clase.

- **AVM1Movie:** la clase AVM1Movie se usa para representar los archivos SWF cargados que se crearon en ActionScript 1.0 y 2.0.
- **DisplayObjectContainer:** las clases Loader, Stage, Sprite y MovieClip amplían la clase DisplayObjectContainer.
- **InteractiveObject:** InteractiveObject es la clase base de todos los objetos y se utiliza para interactuar con el ratón y el teclado. Los objetos SimpleButton, TextField, Video, Loader, Sprite, Stage y MovieClip son subclases de la clase InteractiveObject.

- **MorphShape:** estos objetos se crean al crear una interpolación de forma en la herramienta de edición de Flash. No es posible crear instancias de estos objetos con ActionScript pero se puede acceder a ellos desde la lista de visualización.
- **Stage:** la clase Stage amplía la clase DisplayObjectContainer. Hay una instancia de Stage por aplicación y se sitúa en lo más alto de la jerarquía de la lista de visualización. Para acceder a Stage, debe usarse la propiedad stage de cualquier instancia de DisplayObject.

Además, la clase StaticText del paquete flash.text amplía la clase DisplayObject, pero no es posible crear una instancia de ella en el código. Los campos de texto estático se crean únicamente en Adobe Flash CS3 Professional.

CLASE DISPLAYOBJECT

La Clase DisplayObject recoge las propiedades y métodos que tiene todo objeto que se muestra por pantalla. Entre sus propiedades más importantes encontramos:

Posición del objeto en píxeles en los ejes x e y:

```
var x: Number
```

```
var y: Number
```

Rotación del objeto en grados (0° - 360°):

```
var rotation: Number
```

Tamaño del objeto en píxeles:

```
var width: Number
```

```
var height: Number
```

Escala del objeto. No se mide en píxeles. Cuando la escala es 1 el objeto de visualización se encuentra a tamaño original.

```
var scaleX: Number
```

```
var scaleY: Number
```

Transparencia del objeto. Valor numérico que va desde 0, totalmente transparente, hasta 1, con su transparencia original:

```
var alpha: Number
```

Oculto o muestra el objeto totalmente sin semitransparencias. Hace como si el objeto no existiera en pantalla (El usuario no puede seleccionar botones, textos, etc. sin que estos estén visibles):

```
var visible: Boolean
```

Nombre asociado al objeto de visualización para poder localizarlo en la jerarquía de objetos de visualización:

```
var name: String
```

Los clips de película (MovieClip) son una subclase que desciende de la clase DisplayObject, lo que significa que heredan todas sus propiedades y métodos.

CLASE MOVIECLIP

Los MovieClips son uno de los objetos de visualización (DisplayObject) más utilizados a la hora de programar con Flash. Se trata de un objeto de visualización con las siguientes características:

- Tiene una **línea de tiempos**, lo que permite hacer que el objeto se anime y varíe a lo largo del tiempo.
- Permite **contener otros objetos de visualización** dentro de sí mismo, incluso otros MovieClips. Esto hace que sean útiles para crear elementos complejos como formularios, menús con botones, etc.
- Se trata de un objeto de visualización **interactivo** lo que permite que reaccione ante un evento producido por el ratón.

La clase MovieClip contiene todas las propiedades de los objetos de visualización (DisplayObject): x, y, rotation, width, height, scaleX, scaleY, alpha, visible, name. Amplía la clase DisplayObject con las siguientes propiedades:

Número del fotograma actual de la línea de tiempos:

```
var currentFrame: uint
```

Número total de fotogramas en la línea de tiempo del clip:

```
var totalFrames: uint
```

Además de propiedades la clase MovieClip posee sus propias funciones.

Seguidamente se enumeran algunas de ellas:

Detener la reproducción del clip de película en el fotograma en el que se encuentra (currentFrame):

```
function stop(): void
```

Comenzar a reproducir el clip desde el fotograma en el que se encuentra (currentFrame):

```
function play(): void
```

Detener el clip de película en el fotograma indicado por parámetro, pudiendo ser el número de fotograma o nombre de fotograma:

```
function gotoAndStop(fotograma:uint/String): void
```

Comenzar a reproducir el clip desde el fotograma indicado por parámetro, pudiendo ser el número de fotograma o nombre de fotograma:

```
function gotoAndPlay(fotograma:uint/String): void
```

Referencia de la clase MovieClip:

http://livedocs.adobe.com/flash/9.0_es/ActionScriptLangRefV3/flash/display/MovieClip.html

INSTANCIAR UN CLIP DE PELÍCULA O MOVIECLIP DINÁMICAMENTE

Para instanciar un clip tenemos dos opciones:

- estática: Instanciarlo manualmente arrastrándolo al área de trabajo desde la biblioteca
- dinámica: Instanciarlo por programación con una sentencia de new

Por ejemplo, si tenemos una clase llamada Menú. Primero la vinculamos desde la biblioteca.

Para vincular un símbolo de la biblioteca a una clase:

1. Abrir la biblioteca y seleccionar el símbolo
2. Acceder a sus propiedades (botón propiedades o botón derecho del ratón)
3. Marcar la opción Exportar para ActionScript en el apartado Vinculación

Después utilizamos la sentencia new:

```
var miMenu:Menu = new Menu();
```

Y finalmente realizamos un addChild para que el clip de película se pueda ver en pantalla.

```
addChild(miMenu);
```

Son dos pasos para colocar algo en la escena: primero se lo crea utilizando una sentencia new y luego lo colocamos en la escena utilizando la función addChild.

ACCESO A PROPIEDADES DE UNA CLASE

Toda clase u objeto posee propiedades o atributos, cuyos valores se configuran individualmente en cada instancia creada.

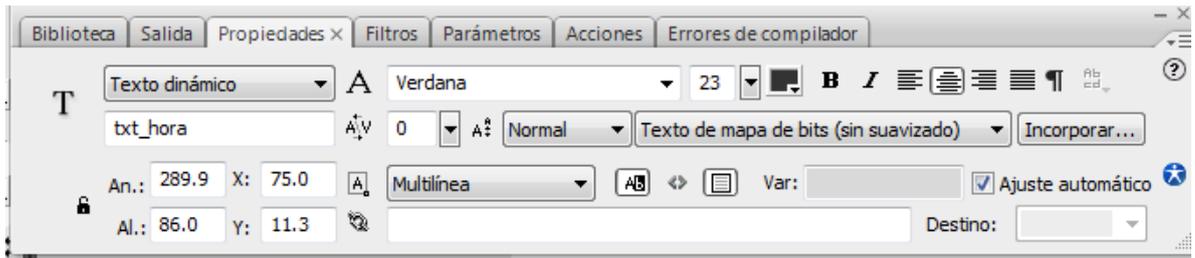


FIGURA III. 385: VENTANA DE PROPIEDADES DE UNA CLASE

Fuente: Adobe Flash CS6

Para acceder a las propiedades de los objetos se utiliza el operador punto (.)

Ejemplo: `txt_hora.text`

Se está accediendo a la propiedad text del objeto texto llamado txt_hora

ACCESO A MÉTODOS:

Los métodos de las clases son accedidos utilizando también el operador punto (.). Y son invocados a través del operador de llamada a función (), los métodos en su invocación pueden necesitar o no parámetros, y además pueden devolver o no valores.

Ejemplo:

```
var h=hora.getHours ();
```

```
this.txt_hora.appendText(h_Total);
```

3.5.12. CARGAR ELEMENTOS EXTERNOS

Flash proporciona mecanismos para comunicarse con otros equipos o servidores. Para ello hace uso de los distintos protocolos de comunicación logrando con ello realizar aplicaciones más dinámicas, la comunicación externa se realiza a través del navegador ejecutando las distintas funciones AS3 para comunicación externa, la herramienta principal de comunicación que brinda AS3 es la función global `navigateToURL` cuyo comportamiento es similar a la etiqueta `<a href >` de html.

La función `navigateToURL` envía una petición al navegador solicitando que cargue una página de una dirección URL específica

`navigateToURL` permite pasar variables con la url, por medio del atributo `data` de `URLRequest`, y elegir la ventana en la que se desea abrir el enlace ("`_blank`", "`_self`", "`_parent`" o "`_top`").

Sintaxis `navigateToURL`

```
navigateToURL (url_pagina, destino);
```

Se pueden pasar URL absolutas y/o URL relativas:

TABLA XVIII. XVII: DESCRIPCIÓN DE URL ABSOLUTA Y RELATIVA

URL Absoluta	Enlace directo a un servidor en internet	<code>http://www.google.com</code>
URL Relativa	Enlace directo a un directorio en el mismo servidor	<code>/noticias/noti1.htm</code>

Fuente: Libro "ActionScript 3.0 Advance" CHURA David

CAPÍTULO IV

4. PROPUESTA

4.1. RECOPIACIÓN DE LA INFORMACIÓN

Para definir los contenidos del curso virtual, es imperativo hacer un reconocimiento del nivel de conocimiento previo sobre la programación en AS3.0, para llegar a los resultados a continuación expuestos se ha definido una encuesta (ver anexo 2) en la que especificamos varias preguntas puntuales sobre temas fundamentales sobre el conocimiento en la programación del lenguaje mencionado, a ser aplicado a nuestro segmento.

En primer lugar se establece si el estudiante conoce sobre la utilidad del lenguaje AS3.0 para determinarlo se ha preguntado sobre los usos posibles del

lenguaje dando varias opciones tanto correctas como incorrectas, dando como resultante un 25% de desconocimiento total mientras el 75% de estudiantes escogieron entre las opciones correctas, la pregunta que se efectuó fue la siguiente:

¿Cuáles de las siguientes opciones considera posibles mediante la aplicación del lenguaje de programación AS3.0 de Adobe Flash?

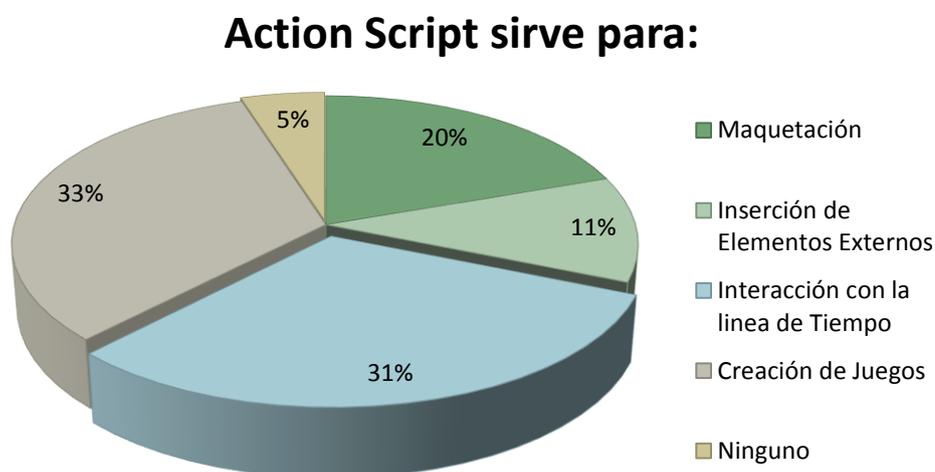


FIGURA IV. 396: CONOCIMIENTOS PREVIOS - USO AS3.0
Fuente: Elaboración propia basado en encuestas

En cuanto a los conocimientos sobre los fundamentos de la POO (Programación Orientada a Objetos) , pocos de los estudiantes demuestran desconocimiento previo siendo que el 31% desconoce de los mismos mientras el 69% tiene conocimiento de aquellos, para definirlos se establecen cuatro opciones, una correcta y 3 erróneas

Fundamentos

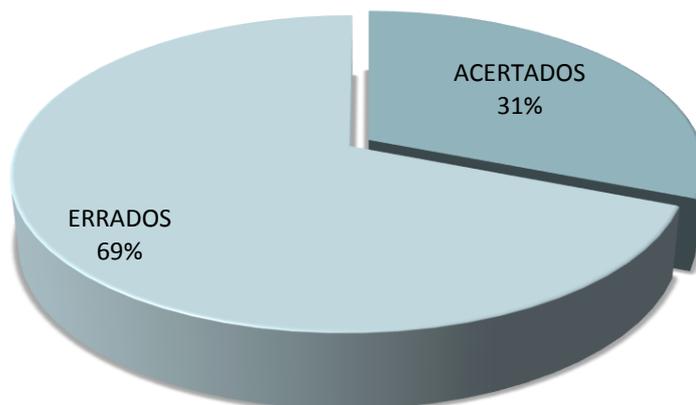


FIGURA IV. 47: CONOCIMIENTOS PREVIOS - FUNDAMENTOS DE LA POO

Fuente: Elaboración propia basado en encuestas

Para establecer el conocimiento previo sobre el uso de las funciones se proponen 5 alternativas, donde las tres primeras son correctas y las dos últimas son las opciones de seleccionar todas o ninguna, en donde el 36% de los estudiantes seleccionan esta última opción resultando en esto un bajo grado de desconocimiento.

Funciones

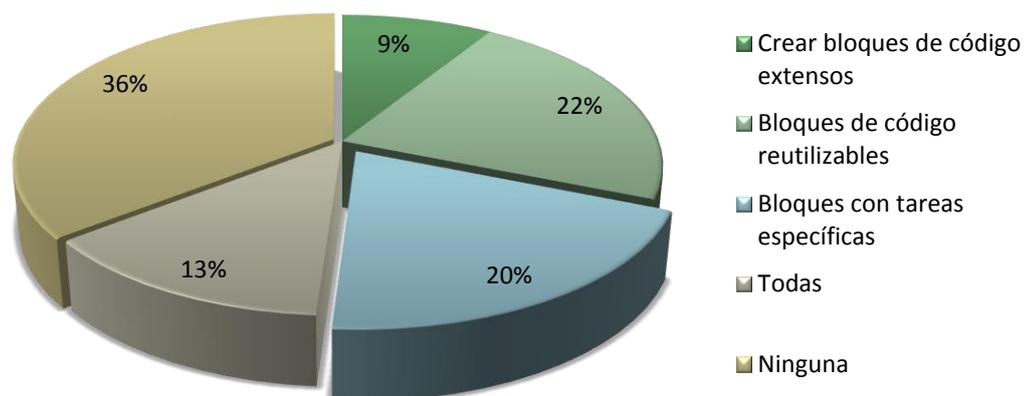


FIGURA IV. 48: CONOCIMIENTOS PREVIOS - FUNCIONES

Fuente: Elaboración propia basado en encuestas

En cuanto al uso de variables el 60% de los estudiantes si tienen previo conocimiento, sobre su uso y declaración.

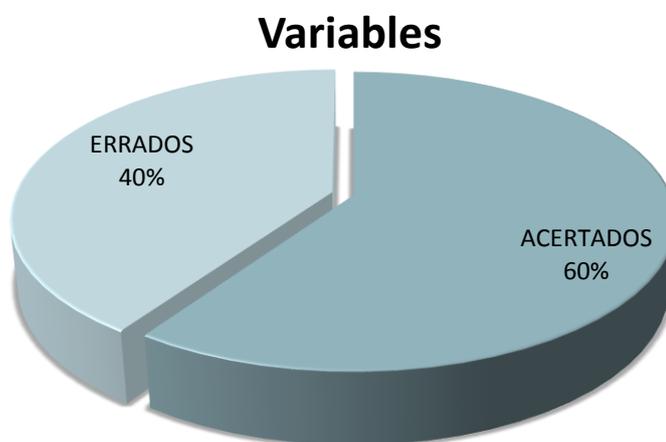


FIGURA IV. 49: CONOCIMIENTOS PREVIOS - VARIABLES

Fuente: Elaboración propia basado en encuestas

Refiriéndonos a la declaración y uso de eventos se observa un notable desconocimiento por parte de los estudiantes dado que apenas un 19% acierta, siendo este un punto importante y a destacarse en nuestro curso virtual.



FIGURA IV. 400: CONOCIMIENTOS PREVIOS – EVENTOS

Fuente: Elaboración propia basado en encuestas

Los Loops y ciclos resultaron casi equitativos los aciertos como los errores, siendo un 41% conocedor de su uso en el lenguaje mientras el 59% no lo conoce, por esto también resultaron ser temas que si bien no tienen el mismo grado de desconocimiento como el punto anterior también merece tener énfasis en su desarrollo, para esto se propone una pregunta de verdadero y falso donde todas las opciones son verdaderas dando como resultados los siguientes:

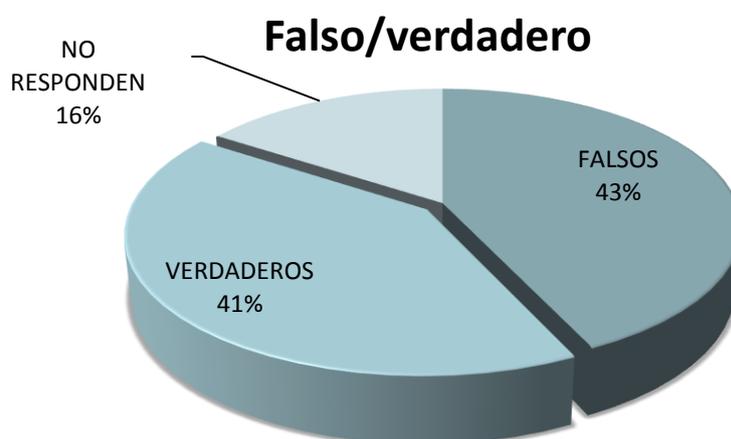


FIGURA IV. 411: CONOCIMIENTOS PREVIOS – CICLOS Y LOOPS

Fuente: Elaboración propia basado en encuestas

Para determinar las capacidades que poseen ya los estudiantes de sexto semestre de la escuela de Diseño Gráfico de la ESPOCH, se aplica un ejercicio con el objetivo de conocer las falencias existentes en el uso del software y en la capacidad de programación en AS3.0.

En este ejercicio se han tomado en cuenta varios parámetros, en primer lugar deseamos determinar el nivel de capacidad en el uso del software para desarrollar actividades elementales que son necesarias para el desarrollo de un

producto multimedia, como son generar botones animados, clips de película y la correspondiente instanciación para la exportación a código AS3.0. Resultando un 75% de nuestra población de estudio errantes en el uso del software para desarrollar las 3 tareas básicas.

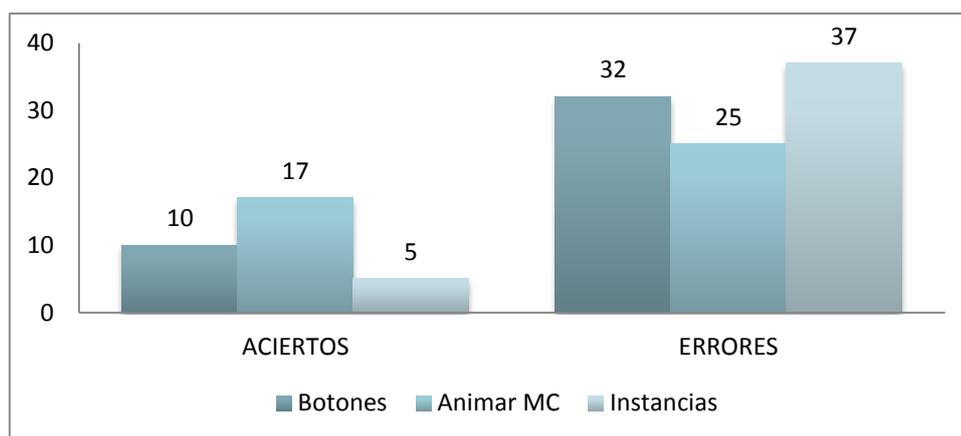


FIGURA IV. 422: CONOCIMIENTOS PREVIOS – ERRORES Y ACIERTOS POR TAREA, EJERCICIO PRÁCTICO EN EL USO DEL SOFTWARE ADOBE FLASH
Fuente: Elaboración propia basado en encuestas

Ejercicio práctico (Uso flash)

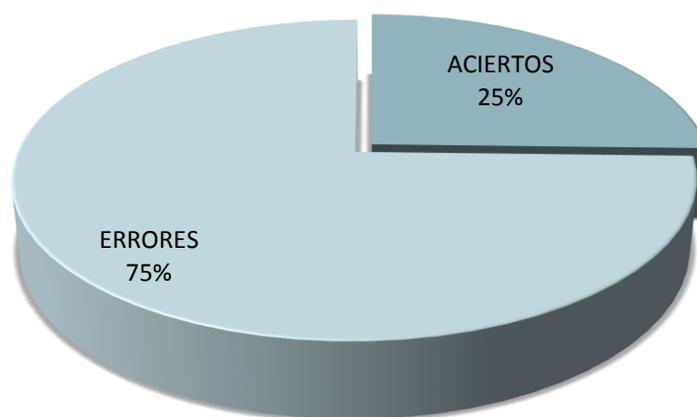


FIGURA IV. 433: CONOCIMIENTOS PREVIOS – ERRORES Y ACIERTOS TOTALES, EJERCICIO PRÁCTICO EN EL USO DEL SOFTWARE ADOBE FLASH
Fuente: Elaboración propia basado en encuestas

De la misma forma para medir la capacidad en el desarrollo de material interactivo con programación AS3.0 se establecen varios parámetros, como declaración de variables, uso de ciclos y funciones, obteniendo un 33% de aciertos y un 67% de errores, a continuación en la siguiente gráfica se detalla los resultados en cada uno de los parámetros evaluados.

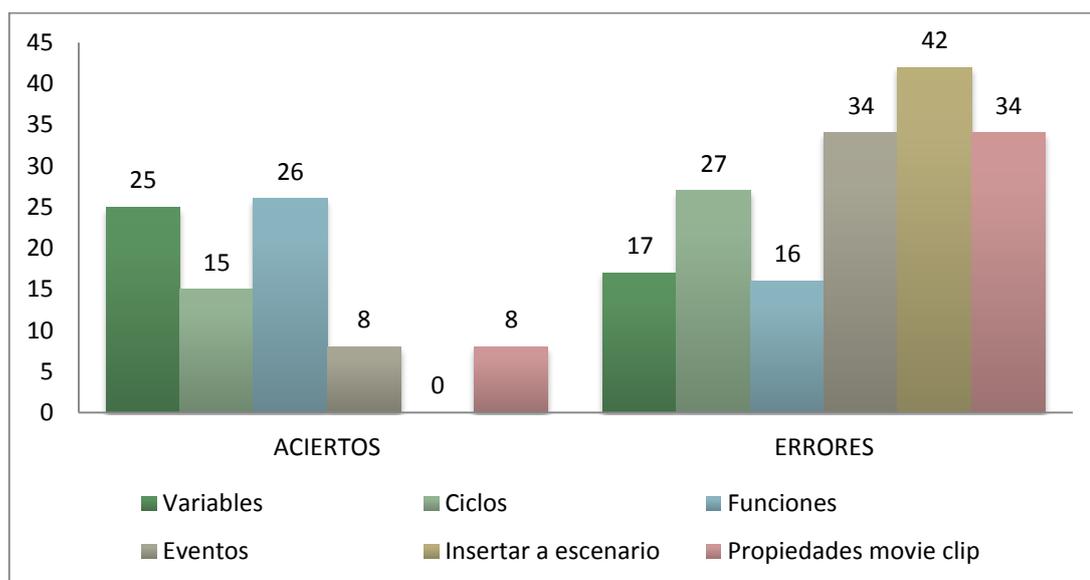


FIGURA IV. 444: CONOCIMIENTOS PREVIOS – ERRORES Y ACIERTOS POR PARÁMETRO, EJERCICIO PRÁCTICO PROGRAMACIÓN EN AS3.0

Fuente: Elaboración propia basado en encuestas

Ejercicio práctico (código AS3.0)

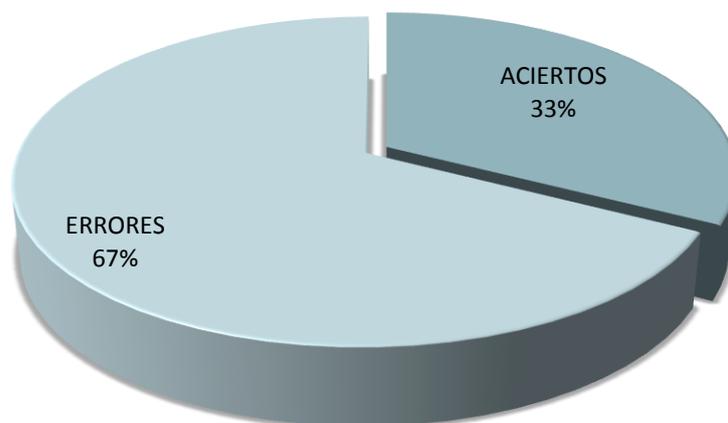


FIGURA IV. 455: CONOCIMIENTOS PREVIOS – ERRORES Y ACIERTOS TOTALES, EJERCICIO PRÁCTICO PROGRAMACIÓN EN AS3.

Fuente: Elaboración propia basado en encuestas

4.2. ANÁLISIS Y ORGANIZACIÓN DE LA INFORMACIÓN.

Entonces encontramos que los estudiantes poseen conocimientos previos sobre la POO, más sin embargo en muchos de los casos no son suficientes para poder desarrollar una aplicación multimedia interactiva, concluimos entonces que nuestro curso virtual deberá reforzar aquellos conocimientos previos partiendo desde las bases de la POO, continuando por la gramática y sintaxis del AS3.0, y dado que la mayor falencia de los estudiantes se demuestra en la parte práctica se determina que los temas deben ser explicados de forma práctica en nuestro curso virtual. A continuación en la siguiente tabla resumimos los resultados obtenidos sobre los conocimientos previos:

TABLA XVIII. XVIII: CONOCIMIENTOS PREVIOS DE LOS ESTUDIANTES

CONOCIMIENTOS PREVIOS		
Conocimiento de la utilidad del AS3.0		75% Conoce % Desconoce 25
Fundamentos de la POO		31% Conoce % Desconoce 69
Conocimiento del uso de funciones		64% Conoce % Desconoce 36
Conocimiento del uso de variables		60% Conoce % Desconoce 40
Conocimiento del uso de Eventos		19% Conoce % Desconoce 81
Condicionales y loops		41% Conoce % Desconoce 59
Capacidad de desarrollo de aplicaciones	Uso del software	Botones 10 Aciertos 32 Errores
		Animar MC 17 Aciertos 25 Errores
	Programación en AS3.0	Generar instancias 5 Aciertos 37 Errores
		Declaración de variables 25 Aciertos 17 Errores
		Manejo de ciclos 15 Aciertos 27 Errores
		Declaración de funciones 26 Aciertos 16 Errores
		Eventos 8 Aciertos 34 Errores
		addChild 0 Aciertos 42 Errores
		Propiedades de movie clip 8 Aciertos 34 Errores
		TOTAL

Fuente: Elaboración propia basado en encuestas

4.3. DISEÑO DEL CURSO VIRTUAL MULTIMEDIA.

Para facilitar el entendimiento de la estructura del curso se ha implementado en un sistema de navegación jerarquizado, el software que se usará para el desarrollo del mismo será adobe flash CS6 mediante su lenguaje de programación AS3.0 por lo que se define como un multimedia que usa el modelo orientado a objetos, ya que nos permite dar la interactividad requerida.

4.3.1. REQUERIMIENTOS BÁSICOS

Una vez establecida la plataforma de diseño de nuestro curso virtual de AS3.0 se determinan los siguientes requisitos:

Ordenador: Usaremos un ordenador con un procesador core i5 y 4 gb en RAM

Software de diseño: Son programas que nos permiten generar gráficos ya sean vectoriales o mapas de bits, así como también videos en este caso utilizaremos para el diseño del curso programas que pertenecen al pack de Adobe Suite, Adobe Illustrator CS6, Adobe Photoshop CS6, para la creación de gráficos e identidad. En el aspecto de edición de video se utilizará Adobe Premiere CS5. Y para la maquetación y elaboración de los prototipos y curso virtual acabado Adobe Flash CS6.

Software adicional: También se requiere software adicional para imprimir pantalla en video y audio para esto se ha utilizado el software Snagit 10 y Audacity 2.0.3

4.3.2. CONTENIDOS MULTIMEDIA

La organización de los contenidos es importante pues no debe generar ningún tipo de confusión en el estudiante, los contenidos tanto del PDF como del Multimedia deben poseer la misma estructura y secuencia, sin embargo el PDF por su formato permite ampliar la información de forma considerable.

Organización de los contenidos Multimedia:

1. Adobe Flash CS6

1.1. Interfaz de Flash CS6

2. Programación Orientada a Objetos

2.1. Ventajas

2.2. Fundamentos POO

2.3. Conceptos POO

3. Introducción al AS 3.0

3.1. Sintaxis de AS 3.0

3.2. Tipos de datos de Action Script 3.0

3.3. Crear y usar variables

3.4. Operadores

3.5. Expresiones

4. Estructuras de AS 3.0

4.1. Condicionales

4.2. Loops (Ciclos)

4.3. Funciones

5. Eventos

5.1. Eventos de mouse

5.2. Eventos de stage

6. Objetos de visualización y Clips de película

6.1. Clase Display Object

6.2. Movie Clip

7. Cargar elementos externos

8. Programación avanzada (ejercicio recopilatorio)

4.3.3. CONTENIDOS HIPERMEDIA INFORMATIVO PDF

Organización de los contenidos PDF:

1. Adobe Flash CS6 1.1. Introducción a ActionScript 3.0

1.2. Interfaz de Flash CS6

2. Programación Orientada a Objetos 2.1. Ventajas

2.2. Fundamentos POO

2.3. Conceptos POO

3. Programación en AS 3.0 3.1. Sintaxis de AS 3.0

3.2. Tipos de datos de ActionScript 3.0

3.3. Crear y usar variables

3.4. Operadores

4. Estructuras de AS 3.0 4.1. Condicionales

4.2. Loops (Ciclos)

4.3. Funciones

5. Eventos 5.1. Gestión de Eventos

6. Objetos de visualización y Clips de película 6.1. Clase Display Object

6.2. Clases y Objetos

6.3. Componentes de AS3.0 en Adobe Flash CS6

7. Cargar elementos externos

8. Contenido Avanzado Adicional 8.1. Matemáticas y ActionScript

8.2. Trabajar con texto

8.3. Creando gráficos, clase Graphics

8.4. Filtros

8.5. Filtro Matriz de colores

8.6. Filtro Convolución

8.7. Programando animaciones

8.8. La Clase Tween

4.3.4. MAPA DE NAVEGACIÓN

La información a formar parte de nuestro curso virtual está organizada de manera jerárquica con los siguientes contenidos:

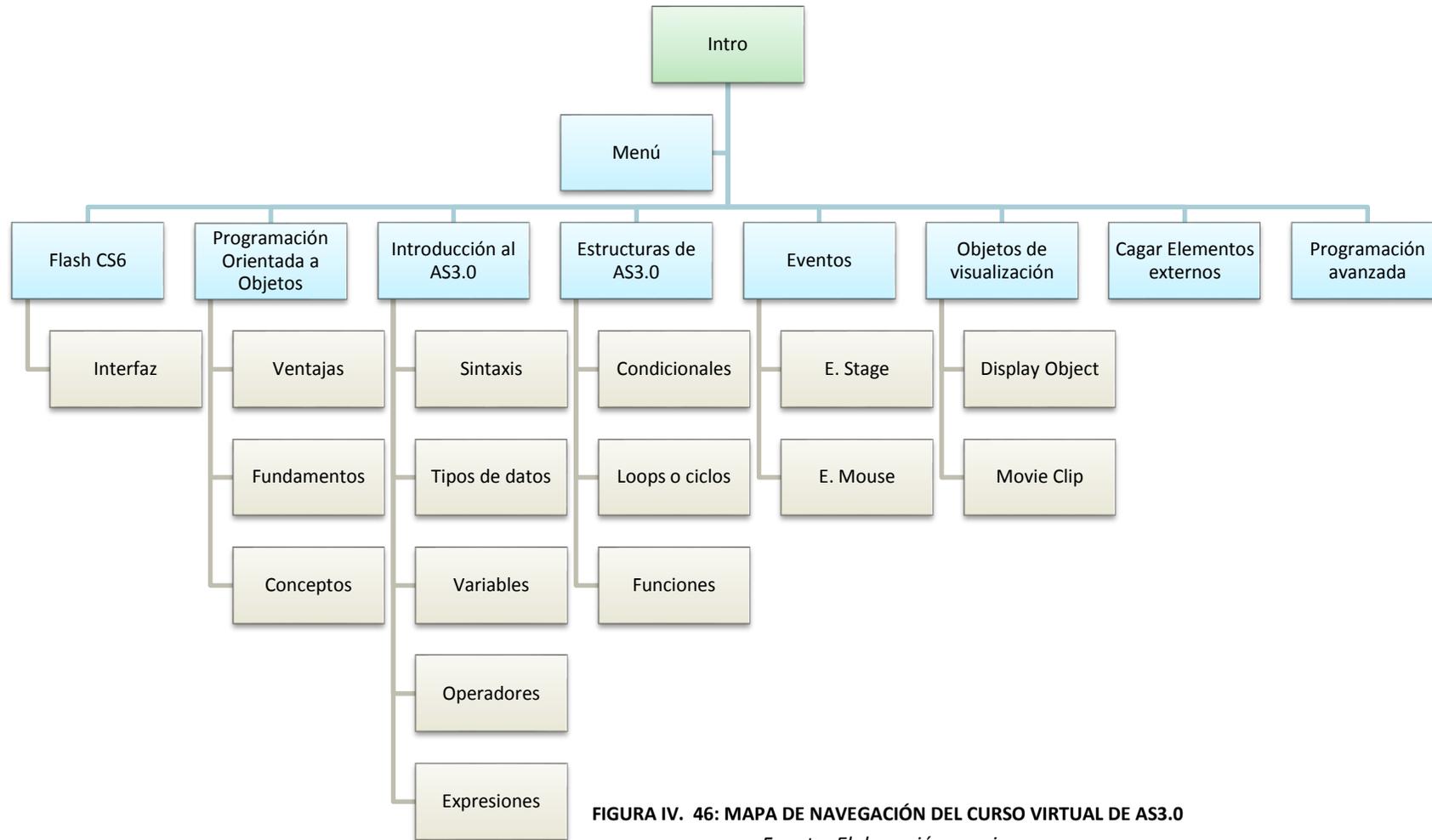


FIGURA IV. 46: MAPA DE NAVEGACIÓN DEL CURSO VIRTUAL DE AS3.0
Fuente: Elaboración propia.

4.3.5. DISEÑO DEL ICONO O LOGO

EL logo fue creado con el principal objetivo identificar al multimedia como material educativo sobre el lenguaje de programación ActionScript 3.0, y reconocerlo como un auxiliar para el aprendizaje del mismo, con direccionamiento a los estudiantes de Diseño Gráfico.

Para la realización del icono se emplearon los conceptos y objetivos del prototipo, los conceptos de su funcionalidad y contenidos fueron definidos para mostrarlos de manera explícita para facilitar el reconocimiento y contenidos del prototipo a simple vista.

OPCIÓN 1



FIGURA IV. 57: POSIBLE ICONO DEL MULTIMEDIA OPCIÓN1

Fuente: Elaboración propia

OPCIÓN 2



FIGURA IV. 58: POSIBLE ICONO DEL MULTIMEDIA OPCIÓN2

Fuente: Elaboración propia

TABLA XIXV. IXX: COMPARACIÓN DE ICONOGRAFÍAS IDENTIFICATIVAS

COMPARACIÓN DE ICONOGRAFÍAS		
PARÁMETROS	OPCIÓN 1	OPCIÓN 2
Estética	45%	55%
Legibilidad	77%	23%
Pregnancia	62%	38%
Reconocimiento	64%	36%

Fuente: Elaboración propia basado en encuestas

Dos propuestas fueron presentadas, sin embargo por el grado de pregnancia y reconocimiento sobre el contenido se determinó la primera iconografía para la identificación del curso.

FONOTIPO:

Para la selección del fonotipo se decidió tomar de forma explícita la denominación el contenido con la finalidad de un reconocimiento inmediato del contenido del curso virtual. Siendo “Curso Virtual AS3.0” fonotipo principal y

“para diseñadores gráficos” como fonotipo secundario, señalando los contenidos el público objetivo del curso.

CÓDIGO CROMÁTICO:

La cromática elegida para el diseño del prototipo se baso en colores cálidos haciendo una conección directa relacional con el software Adobe Flash, donde se trabaja con Action Script 3.0.

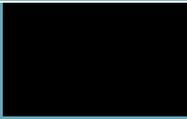
Magenta: Curiosidad, entusiasmo

Naranja: Juventud, dinamismo

Negro: Sofisticación, seriedad

Blanco: Descanso Visual

TABLA XXV. XX: CÓDIGO CROMÁTICO

	RGB	CMYK	PANTONE
	R: 255 G: 255 B: 255	C: 0% M: 0% Y: 0% K: 0%	FFFFFF
	R: 211 G: 6 B: 75	C: 11% M: 99% Y: 55% K: 1%	D3064B
	R: 242 G: 160 B: 0	C: 2% M: 44% Y: 96% K: 0%	F2A000
	R: 0 G: 0 B: 0	C: 100% M: 100% Y: 100% K: 100%	000000

Fuente: Elaboración propia

GRAFÍA:

Se pretende dar un toque de informalidad por lo que se han utilizado curvas y ondas para el multimedia.

TIPOGRAFÍA PRINCIPAL:

Por la legibilidad clara que presenta se ha utilizado la fuente Helvetica 55 Roman

A B C D E F G H I J K L M
N Ñ O P Q R S T U V W X
Y Z a b c d e f g h i j k l m
n ñ o p q r s t u v w x y z

FIGURA IV. 59: HELVETICA 55 ROMAN
Fuente: Elaboración propia

TIPOGRAFÍA SECUNDARIA:

Y se ha tomado la fuente Helvetica27-CondensedUltraLight de tipo ligh de la misma familia como tipografía secundaria.

A B C D E F G H I
J K L M N Ñ O P Q
R S T U V W X Y Z
a b c d e f g h i j k
l m n ñ o p q r s t u
v w x y z

FIGURA IV. 470: HELVETICA27-CONDENSEDULTRALIGHT
Fuente: Elaboración propia

RETÍCULA:



FIGURA IV. 481: RETÍCULA DEL LOGO

Fuente: Elaboración propia

4.3.6. DISEÑO DEL INTERFAZ

Para el diseño del interfaz se tomaron en cuenta los parámetros que establece la teoría de la Gestalt en cuanto a la percepción, y siguiendo la gama cromática cálida característica del software adobe Flash.

El curso virtual está distribuido en 4 bloques principales mismos que facilitan la navegación: el bloque de menú ubicado a la izquierda bajo del ícono identificativo o logo, bajo este los botones de navegación: cerrar, inicio, botón de acceso al pdf, y botón de audio; a la derecha de estos una sección descriptiva misma que mostrará una pequeña síntesis de cada uno de los capítulos del curso. Y por último en la sección de mayor tamaño los videos a mostrarse mismos que son las guías prácticas que describen mediante ejercicios el uso y aplicación de cada uno de los temas de estudio.

PANTALLAS:

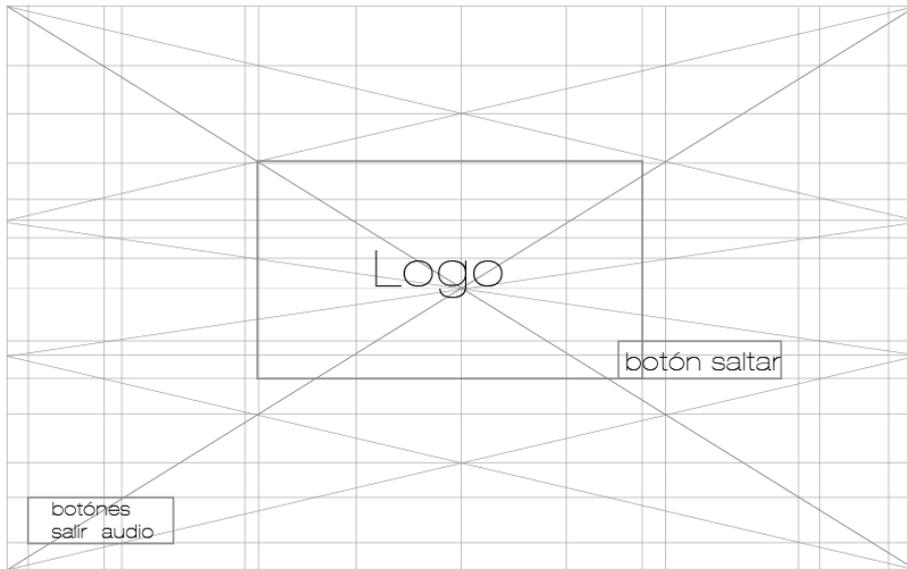


FIGURA IV. 492: RETÍCULA PANTALLA DE BIENVENIDA
Fuente: Elaboración propia



FIGURA IV. 503: PANTALLA DE BIENVENIDA CURSO VIRTUAL
Fuente: Elaboración propia

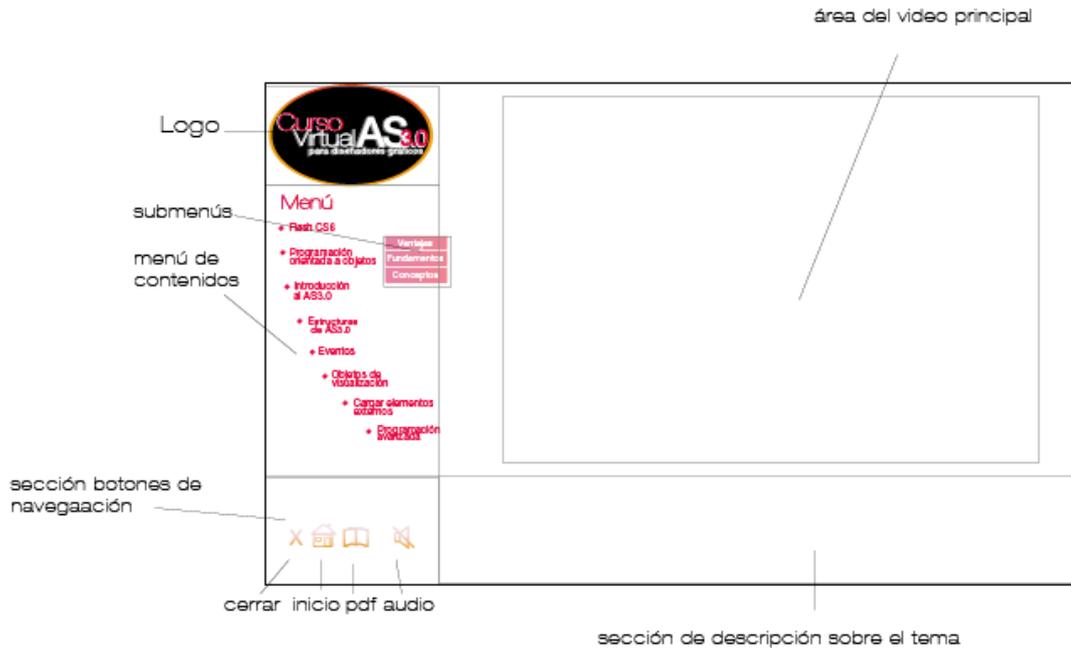


FIGURA IV. 514: DISTRIBUCIÓN DE LAS AREAS DEL MULTIMEDIA

Fuente: Elaboración propia

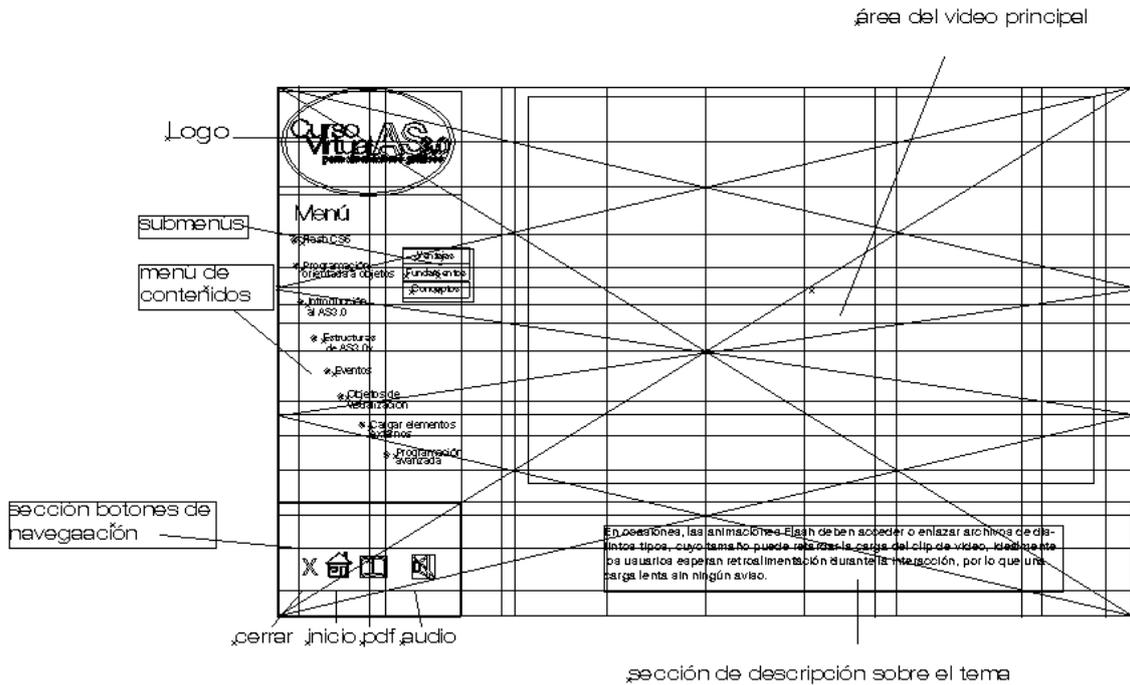


FIGURA IV. 525: RETÍCULA PANTALLA PRINCIPAL

Fuente: Elaboración propia



FIGURA IV. 536: PANTALLA PRINCIPAL CURSO VIRTUAL

Fuente: Elaboración propia

MENÚ:

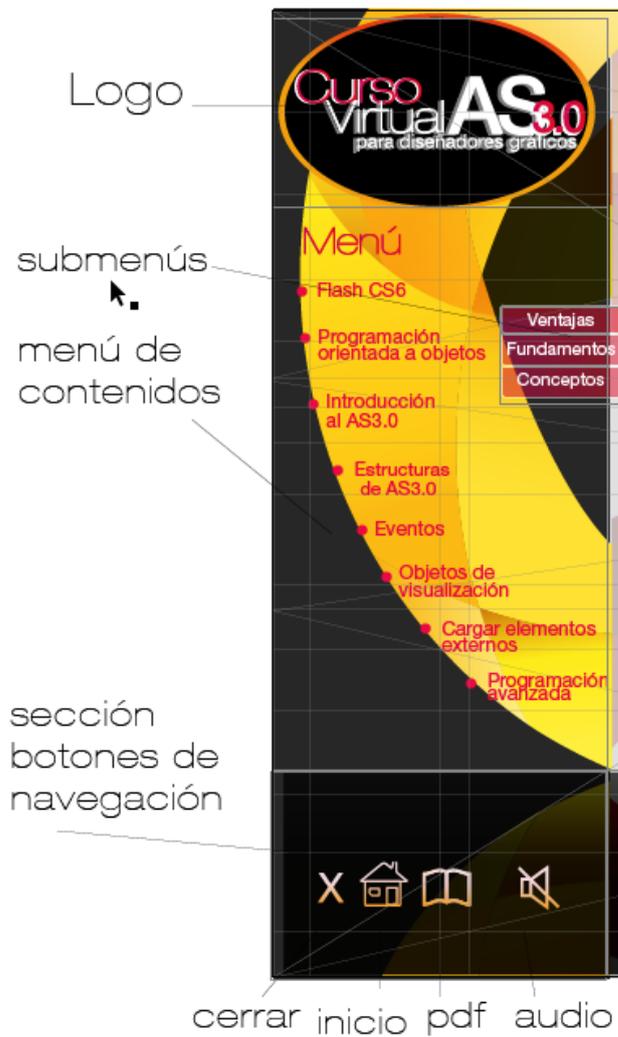


FIGURA IV. 547: DISTRIBUCIÓN DEL MENÚ PRINCIPAL

Fuente: Elaboración propia

4.3.7. DISEÑO DEL PDF INFORMATIVO

Como complementación al multimedia desarrollado se presenta un documento informativo en formato pdf, al que se puede ingresar desde el botón ubicado en el multimedia; este documento facilita la ampliación de la información presentada en el multimedia, este comprende los mismos temas mostrados de forma más amplia por la facilidad que este formato entrega.

Además con la finalidad de facilitar la ampliación de los conocimientos y motivar al estudiante a la investigación el documento posee varios links ubicados estratégicamente de forma que al finalizar cada capítulo, el estudiante pueda consultarlos para ampliar significativamente la información entregada.

En la parte final del documento se encuentra insertado el multimedia, para facilitar el retorno a los contenidos del mismo, además también en la parte final del documento se entrega al estudiante los links de los ejercicios mostrados en los tutoriales en la extensión .fla para ser editados y estudiados en el software propio del lenguaje AS3.0, con el objetivo de impulsarlo a la práctica del lenguaje.

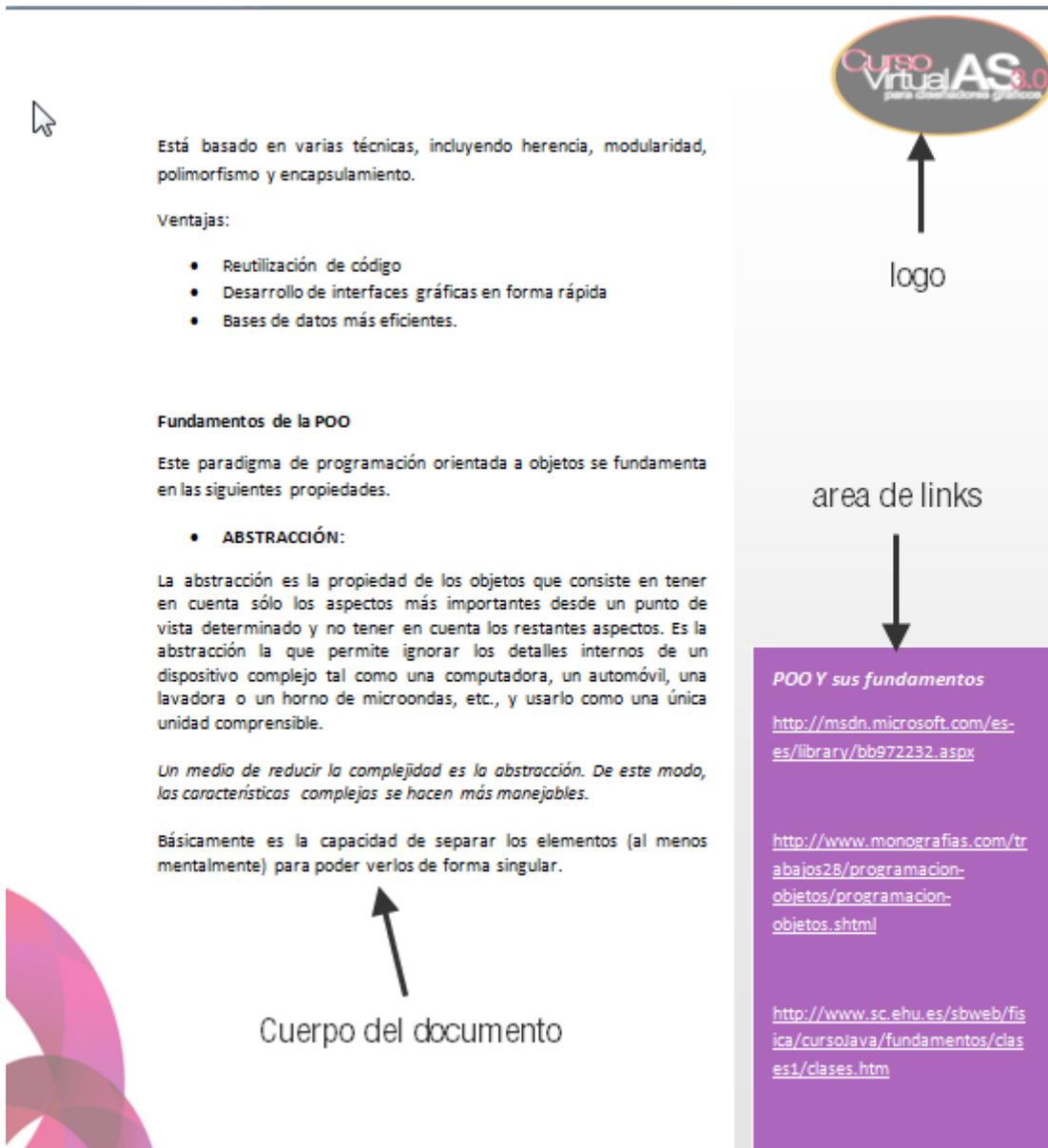


FIGURA IV. 558: ORGANIZACIÓN DEL DOCUMENTO PDF

Fuente: Elaboración propia

4.3.8. PRUEBA DE USABILIDAD

Para establecer la usabilidad del prototipo se realizó una encuesta para definir los parámetros de amigabilidad del interfaz y la calidad de los contenidos abordados.

AMIGABILIDAD DEL INTERFAZ

Al encuestar a los estudiantes sobre la amigabilidad del interfaz se ha establecido que en aspectos como problemas para navegación, sonidos entre otras opciones a continuación descritas no existe mayor dificultad, sin embargo el usuario si presenta un 22% de inconformidad en cuando al tamaño de los iconos, y un 20% posee problemas para identificar con facilidad los niveles de navegación. Y en un análisis más amplio se establece que el 58% encuentra un interfaz amigable, el 22% lo encuentra medianamente amigable, el 13% poco y el 7% presentó varias dificultades en la navegación del prototipo.

TABLA XXIV. XXI: RESULTADO DE USABILIDAD

	SI	MEDIANA MENTE	POCO	NADA
La interface de usuario no opone ninguna resistencia a la navegación	40%	25%	25%	10%
Se puede regresar con facilidad al menú principal	70%	20%	7%	3%
El acceso a los distintos menús es fácil	80%	12%	5%	3%
Los iconos son suficientemente grandes y fáciles de seleccionar	28%	20%	30%	22%
Los aspectos gráficos de las pantallas y del menú son agradables	63%	14%	21%	2%
Los sonidos son adecuados, además de agradables	65%	27%	5%	3%
Se identifican fácilmente los niveles de navegación	39%	20%	21%	20%
EL multimedia permite ir a todo punto de la aplicación en cualquier momento	67%	30%	3%	0%
Puede salir del multimedia cuando lo desee	70%	26%	4%	0%

Fuente: Elaboración propia basado en encuestas

CONTENIDOS ABORDADOS:

En cuanto a la relevancia, organización, relación, y la capacidad para permitir el desarrollo de destrezas el usuario nos entrega respuestas favorables, sin embargo existe un 22% que manifiesta que no se ejercita suficientemente los contenidos para facilitar el afianzamiento de lo aprendido, y un 17% que encuentra cierta monotonía en la información, Pero a pesar de estas falencias a manera general el 52% se manifiesta positivamente hacia la calidad del contenido abordado, el 23% lo hace medianamente, el 18% no están convencidos, mientras el 7% se encuentra en inconformidad.

Dándonos de forma general la apreciación una aceptación positiva hacia los contenidos expuestos en el prototipo, manifestando positivamente sobre la relevancia de estos, y su percepción en relación al conjunto.

TABLA XXIV. XXII: RESULTADO DE CONTENIDOS

	SI	MEDIANA MENTE	POCO	NADA
La información que contiene es relevante	68%	24%	8%	0%
La información que contiene está bien organizada	58%	32%	9%	1%
La información no se transmite de forma monótona	30%	21%	32%	17%
Se percibe la relación entre el conjunto de mensajes	63%	20%	17%	0%
EL multimedia permite desarrollar habilidades y destrezas específicas del tema que trata	62%	15%	13%	10%
Se ejercita suficientemente los contenidos para facilitar el afianzamiento de lo aprendido	25%	21%	32%	22%
Plantea o insinúa estudios de casos o problemas que el usuario podría desarrollar después	57%	26%	13%	4%

Fuente: Elaboración propia basado en encuestas

TABLA XXIII. XXIII: RESULTADO DE USABILIDAD

PARÁMETROS	RESULTADO DE USABILIDAD			
	SI	MEDIANAMENTE	POCO	NADA
Amigabilidad del interfaz	58%	22%	13%	7%
Contenidos abordados de calidad	52%	23%	18%	7%

Fuente: Elaboración propia basado en encuestas

En conclusión los estudiantes manifiestan de forma positiva en un 52% sobre los contenidos abordados, un 23% lo hace medianamente, el 18% no se encuentran convencidos, mientras que el 7% definitivamente no se encuentra convencidos sobre la calidad de los contenidos abordados.

CAPÍTULO V

5. VALIDACIÓN

5.1. ANÁLISIS DE ENCUESTAS

Posterior a la aplicación del curso virtual a los estudiantes, se procedió a aplicar nuevamente la encuesta para determinar los cambios en los conocimientos de los estudiantes, para determinar si existieron o no mejoría en sus capacidades.

El estudiante en la primera pregunta demuestra que ya posee mayor conocimiento sobre la utilidad del lenguaje AS3.0, dando como resultante que un 98% de aciertos, y apenas un 2% erróneo, quedando la opción ninguna con el 0%.

¿Cuáles de las siguientes opciones considera posibles mediante la aplicación del lenguaje de programación AS3.0 de Adobe Flash?

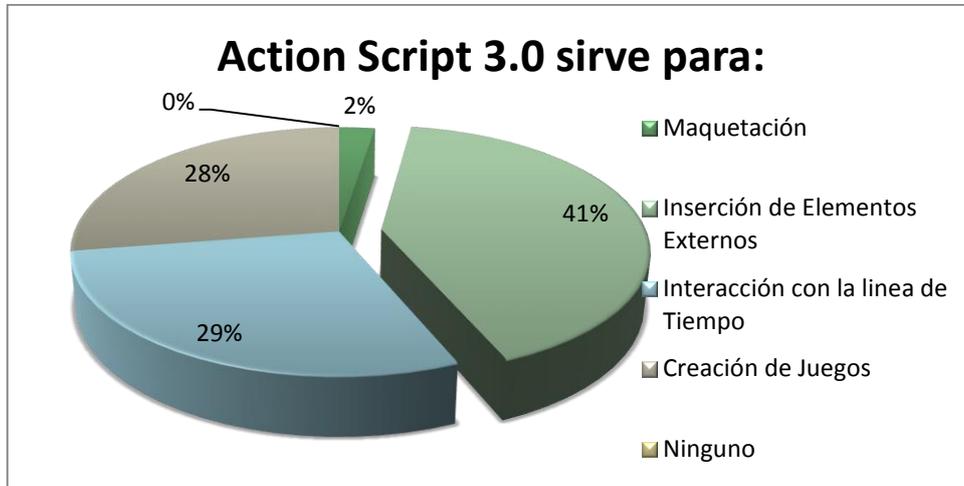


FIGURA V. 569: RESULTADOS DE LA APLICACIÓN- USO AS3.0

Fuente: Elaboración propia basado en encuestas

En cuanto a los conocimientos sobre los fundamentos de la POO (Programación Orientada a Objetos) disminuyendo los errores de un 31% a un 2% posterior a la aplicación del curso virtual.

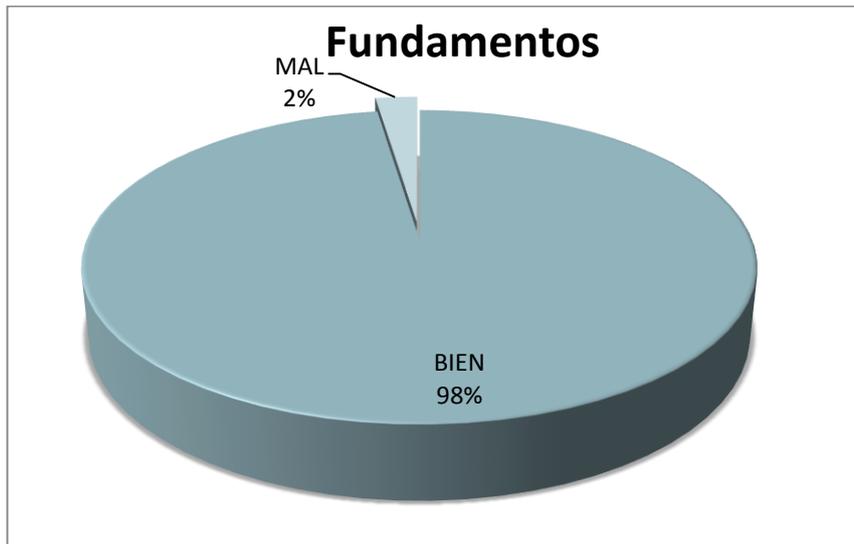


FIGURA V. 70: RESULTADOS DE LA APLICACIÓN - FUNDAMENTOS DE LA POO

Fuente: Elaboración propia basado en encuestas

Con la aplicación se clarifican las dudas existentes sobre las funciones mejorando los resultados notoriamente, ya que el 59% de los estudiantes está

totalmente claro sobre el uso de las funciones, mientras el 49% elige entre las demás opciones ciertas.

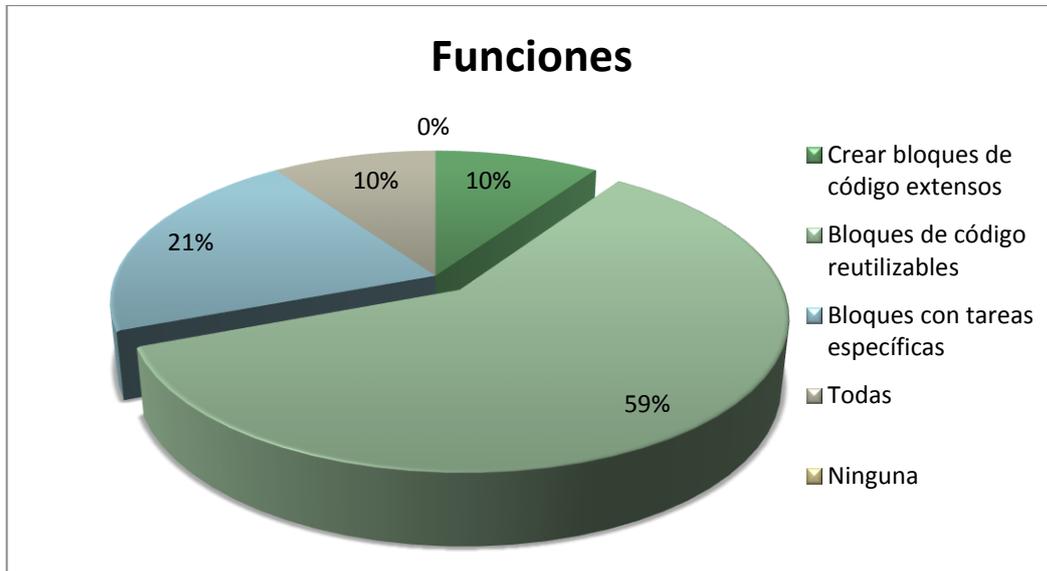


FIGURA V. 71: RESULTADOS DE LA APLICACIÓN - FUNCIONES

Fuente: Elaboración propia basado en encuestas

En cuanto al uso de variables el 60% de los estudiantes poseían conocimientos previos, mismos que se clarifican con la aplicación mejorando de un 60% a un 98%

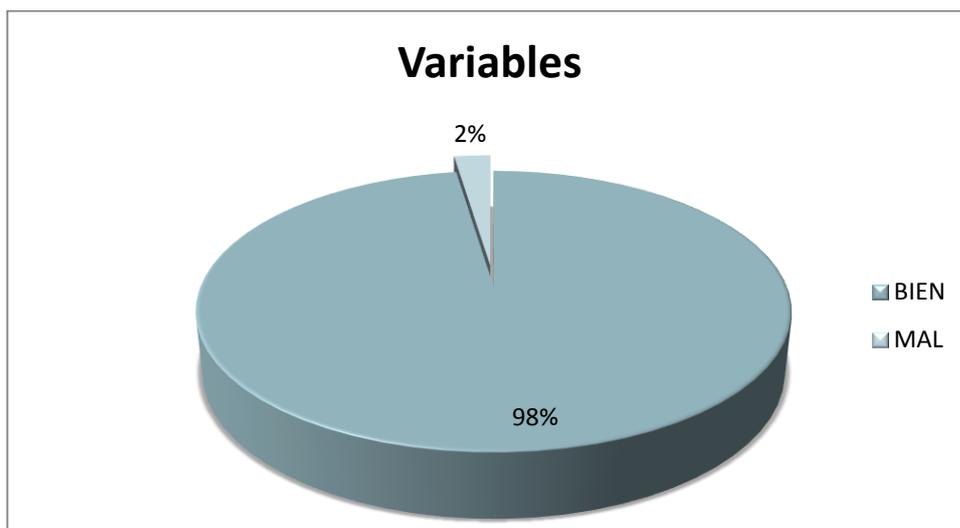


FIGURA V. 72: RESULTADOS DE LA APLICACIÓN - VARIABLES

Fuente: Elaboración propia basado en encuestas

Con la primera encuesta se evidenció que los estudiantes poseían notables falencias en cuanto a uso y conocimiento de eventos dado que apenas un 19% acierto, después de haber sido expuestos los eventos con detenimiento en la aplicación del curso virtual los resultados mejoraron notoriamente a un 95% de aciertos.

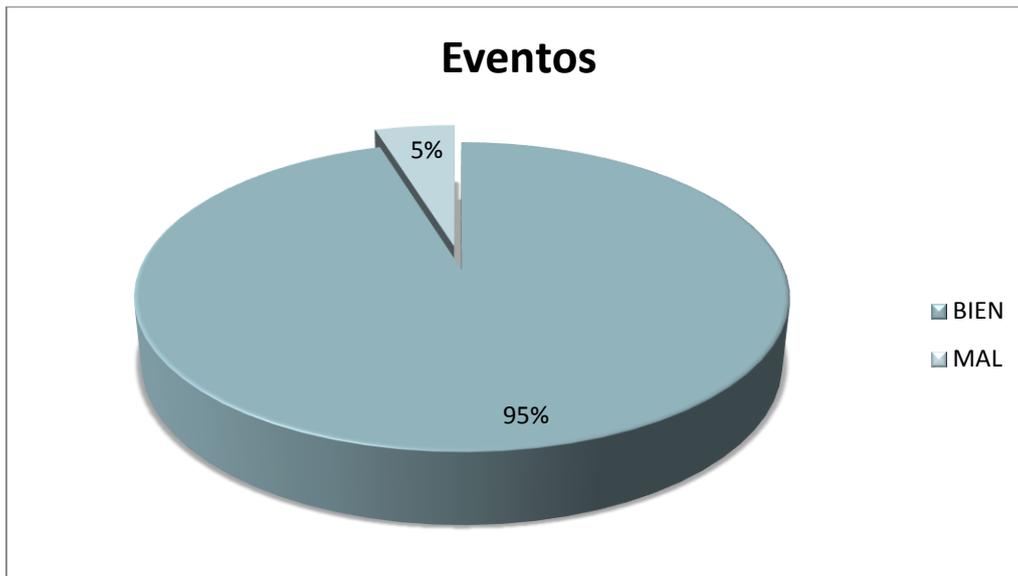


FIGURA V. 73: RESULTADOS DE LA APLICACIÓN – EVENTOS
Fuente: Elaboración propia basado en encuestas

En cuanto a Loops se pasó de un 41% de conocimiento a un 64% mejorando así el conocimiento en cuanto a estructuras condicionales y ciclos:

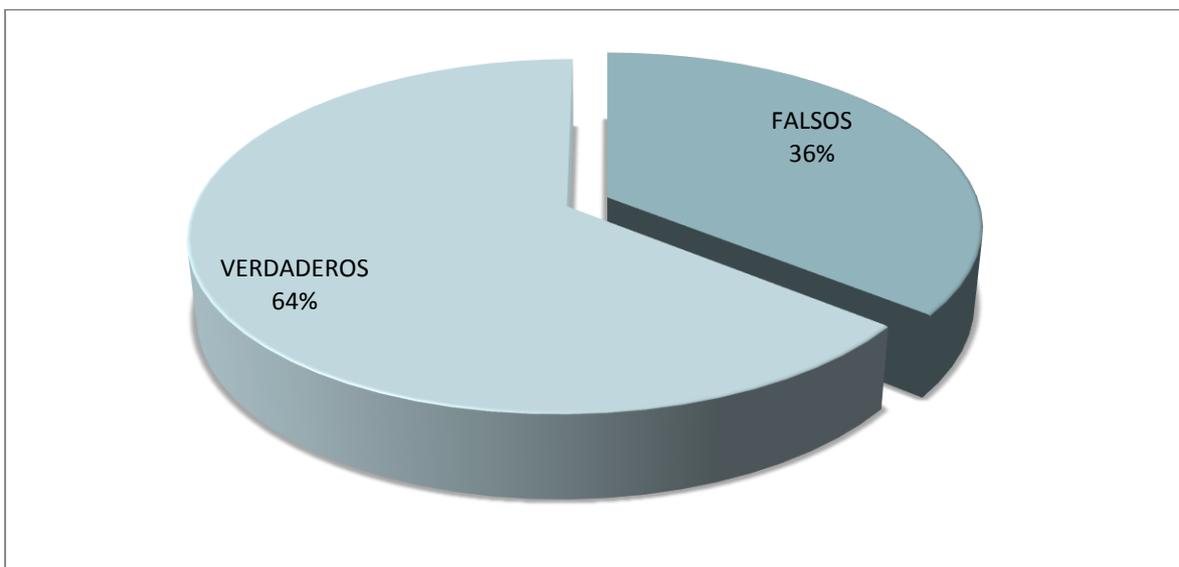


FIGURA V. 74: RESULTADOS DE LA APLICACIÓN – CICLOS Y LOOPS

Fuente: Elaboración propia basado en encuestas

Con la aplicación del ejercicio práctico, se mejoraron las capacidades notablemente la capacidad de programación en AS3.0.

En este ejercicio se nota el cambio en el manejo de los parámetros básicos para crear aplicaciones multimedia como botones animados, clips de película y la correspondiente instanciación para la exportación a código AS3.0. Resultando un 81% de nuestra población de estudio que acertó en el uso del software para desarrollar las 3 tareas básicas.

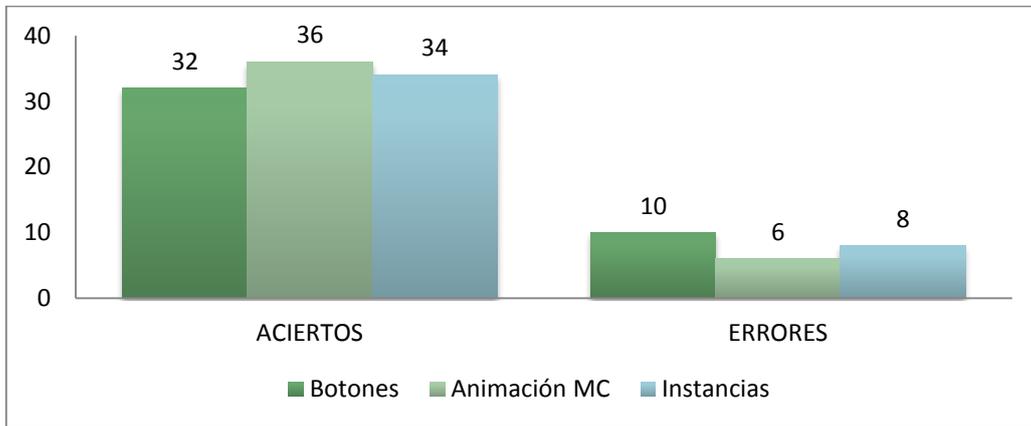


FIGURA V. 75: RESULTADOS DE LA APLICACIÓN – ERRORES Y ACIERTOS POR TAREA, EJERCICIO PRÁCTICO EN EL USO DEL SOFTWARE ADOBE FLASH

Fuente: Elaboración propia basado en encuestas

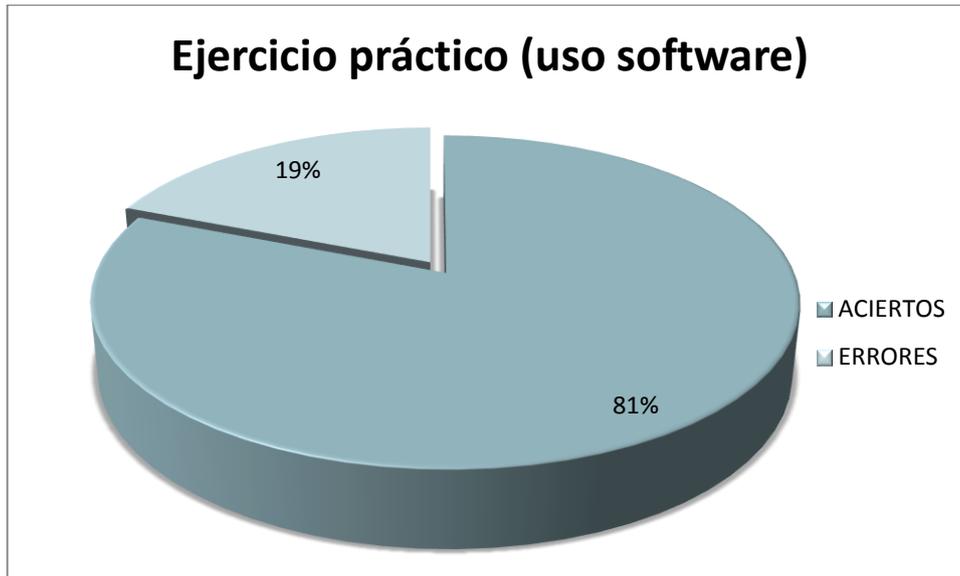


FIGURA V. 76: RESULTADOS DE LA APLICACIÓN – ERRORES Y ACIERTOS TOTALES, EJERCICIO PRÁCTICO EN EL USO DEL SOFTWARE ADOBE FLASH

Fuente: Elaboración propia basado en encuestas

De la misma forma para medir la capacidad en el desarrollo de material interactivo con programación AS3.0 se establecen varios parámetros, como declaración de variables, uso de ciclos y funciones, obteniendo un 89% de aciertos y apenas un 11% de errores, a continuación en la siguiente gráfica se detalla los resultados en cada uno de los parámetros evaluados.

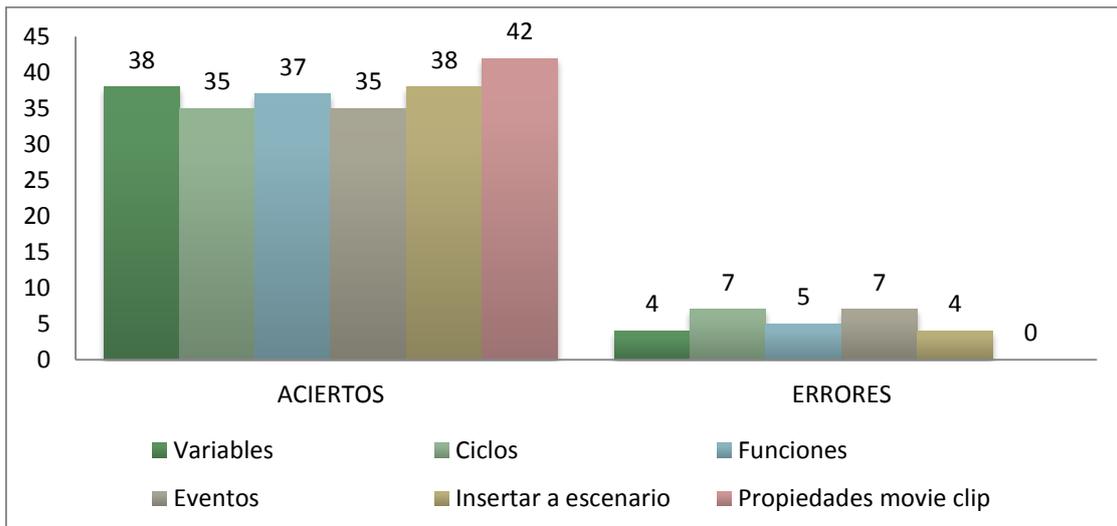


FIGURA V. 77: RESULTADOS DE LA APLICACIÓN – ERRORES Y ACIERTOS POR PARÁMETRO, EJERCICIO PRÁCTICO PROGRAMACIÓN EN AS3.0

Fuente: Elaboración propia basado en encuestas

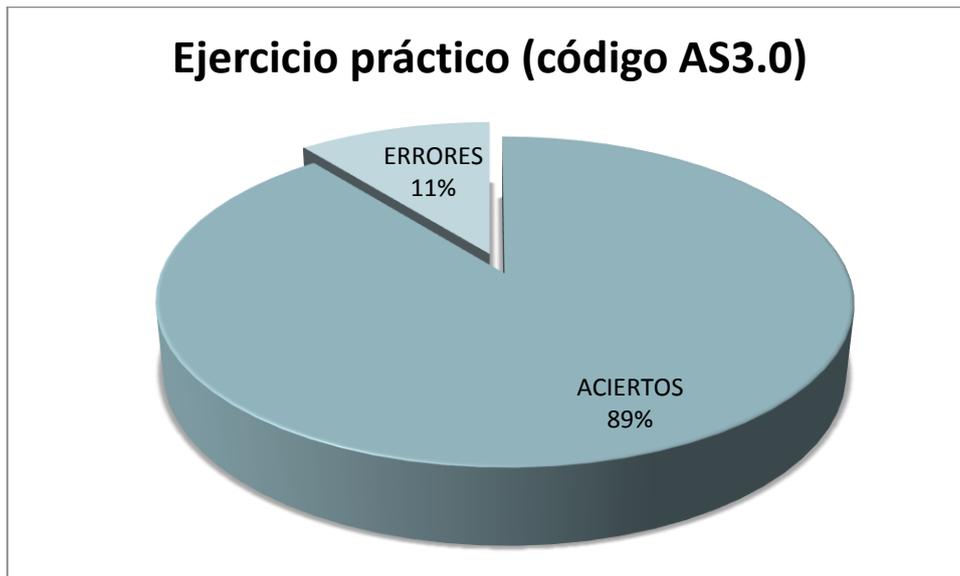


FIGURA V. 578: RESULTADOS DE LA APLICACIÓN – ERRORES Y ACIERTOS TOTALES, EJERCICIO PRÁCTICO PROGRAMACIÓN EN AS3.

Fuente: Elaboración propia basado en encuestas

Dando como resultado después de la aplicación del Curso virtual de AS3.0 las respuestas acertadas pasaron de un 39% a un 85%, demostrando así que se han mejorado notablemente los conocimientos y capacidades de los estudiantes de sexto semestre de Diseño Gráfico de la ESPOCH.

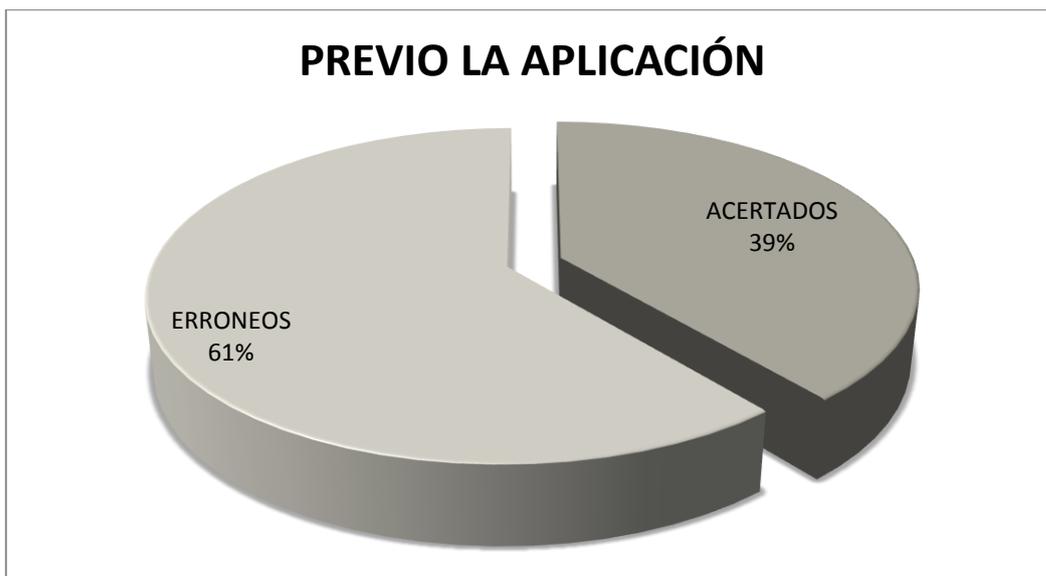


FIGURA V. 79: RESULTADOS PREVIO LA APLICACIÓN DEL CURSO VIRTUAL

Fuente: Elaboración propia basado en encuestas



FIGURA V. 80: RESULTADOS GENERALES POSTERIORES LA APLICACIÓN DEL CURSO VIRTUAL
Fuente: Elaboración propia basado en encuestas

CONCLUSIONES

1. Como resultado de la investigación se descifró que el conocimiento insuficiente sobre la sintaxis, el uso y declaración de variables, loops, funciones, condicionales, eventos y paradigmas de la POO sumados a la falta de experiencia previa en el uso del software y el lenguaje son entre otras las principales limitantes de los estudiantes de sexto semestre de Diseño Gráfico de la ESPOCH, para poder desarrollar aplicaciones multimedia eficientemente, por lo que estas limitaciones nos han dado las pautas para conocer las temáticas más importantes que debieron ser parte del multimedia.
2. El modelo de educación mediante multimedia es una forma innovadora no convencional que brinda innumerables ventajas, que siguiendo los lineamientos de teorías para el desarrollo multimedia como la gestalt, cognitiva y el constructivismo, optimiza el aprendizaje con medios digitales.
3. Una vez identificadas las limitantes de los estudiantes, para definir el contenido teórico y práctico del curso se estudiaron detenidamente los conceptos y paradigmas de la POO y AS3.0, para poder explicarlos de forma clara y sencilla

4. Se determinó así un sistema jerárquico de navegación para una exposición clara de los contenidos tanto del multimedia como del Documento pdf optimizando la comprensión, mostrado en una interfaz amigable y relacionada cromáticamente a Adobe Flash, software que usa ActionScript3.0

5. Mediante la encuesta se pudo verificar que las capacidades de los estudiantes de sexto semestre de la escuela de diseño gráfico mejoraron notablemente, ya que se obtuvo un 46% en la mejoría de conocimientos del y capacidades para programar ejercicios prácticos en lenguaje ActionScript 3.0, esto como resultado de la aplicación del curso virtual.

RECOMENDACIONES

1. La educación presencial puede apoyarse con materiales didácticos multimedia, beneficiando al estudiante al tener material complementario que apoye su aprendizaje, permitiendo que pase a ser parte pasiva del proceso educativo a una activa.
2. Se recomienda para futuras aplicaciones multimedia educativas, realizar previamente el estudio y análisis de las necesidades educativas del público al que va dirigido, para poder discernir eficientemente los contenidos y el esquema de la aplicación, asegurando así la optimización de los contenidos, para crear una interfaz y navegación eficientes.
3. El estudio de los paradigmas de la POO (Programación Orientada a Objetos) y el uso del lenguaje de programación AS3.0 son de gran utilidad; para el desarrollo de aplicaciones multimedia interactivas, por permitir la creación de películas o animaciones con altísimo contenido interactivo, ya que es imperativo para el diseñador gráfico poder desarrollar presentaciones útiles y atractivas.

RESUMEN

Se diseñó un curso virtual de Action Script 3.0 como apoyo del proceso enseñanza aprendizaje hacia los estudiantes de sexto semestre de la escuela de diseño gráfico, Escuela Superior Politécnica de Chimborazo, teniendo como finalidad mejorar capacidades de desarrollo para aplicaciones flash multimedia.

Aplicando el método descriptivo, se encuestaron y observaron a todos los miembros de sexto semestre de la escuela de diseño gráfico, para conocer las características y analizar el grado de aceptación del grupo de estudio ante un posible curso virtual que permitiera entregar material para el desarrollo multimedia, obteniendo como resultado datos cuantitativos y cualitativos, mismos que fueron utilizados diseño del multimedia y sus contenidos; para ello fueron útiles equipos como computador y micrófono, el software utilizado fue: Snagit, Audacity, Illustrator, Photoshop, Premiere, y Flash.

Al ser implementado el curso virtual, se concluyó que el grupo de estudio pudo ampliar sus conocimientos sobre el tema, comprobando que existe mejoría en

sus capacidades y conocimientos para desarrollar aplicaciones multimedia en Action Script 3.0 pasando de un conocimiento previo sobre el 39% a un 85% como resultado de la aplicación del curso virtual; dado que presenta los conceptos y paradigmas de la Programación Orientada a Objetos, y específicamente del Action Script 3.0, ya que fue realizado bajo la aplicación de los métodos adecuados para la enseñanza-aprendizaje para multimedia y los paradigmas de educación virtual.

Se recomienda el uso del lenguaje Action Script 3.0 para desarrollar aplicaciones flash multimedia, por las múltiples utilidades que presenta, además la implementación de material complementario virtual educativo, para facilitar el aprendizaje del estudiante de Diseño Gráfico.

ABSTRACT

It was designed a virtual course of ActionScript 3.0 as a supportive mean of the teaching-learning process for the students of sixth semester of the Graphic Design School belonging to The Escuela Superior Politécnica de Chimborazo. This study is directed in order to improve the development capabilities for multimedia flash applications.

All members studying at sixth semester of the graphic design school were surveyed and observed by following the descriptive method in order to recognize the characteristics and determine the acceptance degree by the study group in case of a possible virtual course that would provide material for multimedia development, resulting in quantitative and qualitative data, used for the multimedia design and it's contents. To make this possible, useful equipment such as computer and microphone, were necessary as well as the software to be used was: Snagit, Audacity, Illustrator, Photoshop, Premiere, and Flash.

Having implemented the virtual course, it was concluded that the study group was able to expand their knowledge on the subject, proving that there is an

improvement in their skills and knowledge to develop multimedia applications in ActionScript 3.0, increasing from a prior knowledge over the 39% to 85% as a result of the implementation of the virtual course, since it shows the concepts and paradigms of the Object Oriented Programming and the ActionScript 3.0 specifically, as it was carried up under the application of appropriate methods for teaching-learning for multimedia and paradigms of virtual education.

It is recommended to use ActionScript 3.0 language to develop multimedia flash applications due to the several utilities it offers, also the implementation of virtual complementary educational materials to use the learning process of the students of Graphic Design.

GLOSARIO

ActionScript: Lenguaje de programación utilizado en el software Adobe Flash, para el desarrollo multimedia.

Frames: fotograma de la película de flash.

Hipermedias: Es el término con el que se designa al conjunto de métodos o procedimientos para escribir contenidos que integren soportes como texto, imagen, video, audio, mapas y otros soportes de información.

Hipertexto: Sistema que permite que un texto contenga enlaces con otras secciones del documento o con otros documentos:

Interactivo: Permite una interacción, a modo de diálogo, entre un ordenador y el usuario.

Icono: Pequeña imagen que se muestra en la pantalla de un ordenador y que representa una opción que puede ser elegida por el usuario

Interfaz: En informática, se utiliza para nombrar a la conexión física y funcional entre dos sistemas o dispositivos de cualquier tipo dando una comunicación entre distintos niveles.

Metodología: Parte de la lógica que estudia los métodos del conocimiento.

Método: Conjunto de reglas y ejercicios destinados a enseñar una actividad, un arte o una ciencia

Multimedia: La multimedia consiste en el uso de diversos tipos de medios para transmitir, administrar o presentar información.

Prototipo: Primer ejemplar que se fabrica de una figura, un invento u otra cosa, y que sirve de modelo para fabricar otros iguales, o molde original con el que se fabrica

Usuario: Persona que hace uso de un ordenador.

ANEXOS

(ANEXO 1)

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

ESCUELA DE DISEÑO GRÁFICO

El objetivo de la presente encuesta es determinar el segmento de mercado, requerido para el desarrollo de un proyecto multimedia en beneficio de los estudiantes de la EDG-ESPOCH.

Se recomienda llenar los siguientes casilleros de forma clara.

Género: FEMENINO: _____ **Edad:** _____ años
MASCULINO:

1.- Califique su desempeño como estudiante de la escuela de diseño, marcando el que corresponda con una X.

Excelente Muy bueno Bueno
Regular Malo

2.- ¿Cómo calificas tu capacidad para el desarrollo de aplicaciones multimedia flash?
(selecciona las opciones más acordes)

Excelentes Buenas Pueden mejorar
Regulares Malas

3.- Dando un valor de 0 a 3 señale el nivel de interés en cada una de estas actividades

- Diseño publicitario _____
- Desarrollo aplicaciones multimedia _____
- Diseño de identidad corporativa _____
- Diseño de medios impresos _____
- Diseño de presentaciones web _____
- Diseño de juegos flash _____
- Creación de presentaciones interactivas _____

4.- En el caso de tener la oportunidad de aprender de forma extracurricular con un curso virtual ¿cuáles son los beneficios que espera obtener? (marque con una X)

- Facilidad de ingreso a la información
- Fuentes de información extras
- Ejercicios prácticos
- Auto-evaluación
- Espacio para discusión sobre el tema de aprendizaje.
- Presentación atractiva y clara de los contenidos

5.- ¿Considera usted que el lenguaje de programación ActionScript 3.0 es muy útil para el desarrollo de aplicaciones y presentaciones multimedia?

- Siempre En algunas ocasiones No es útil Desconozco del lenguaje

(ANEXO 2)

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO ESCUELA DE DISEÑO GRÁFICO

El objetivo del presente test es determinar el nivel de conocimientos previos sobre el lenguaje ActionScript 3.0 de los estudiantes de la EDG – ESPOCH.

Se recomienda llenar los siguientes casilleros de forma clara.

1.- ¿Cuáles de las siguientes opciones considera posibles mediante la aplicación del lenguaje de programación AS3.0 de Adobe Flash? (seleccionar con una X)

Maquetación y Diagramación Interactividad con la línea de tiempo
Inserción de elementos externos Creación de juegos Ninguno

2.- Señale cuales son los fundamentos de la POO

ABSTRACCIÓN
CAPSULACIÓN
FUNCIONES
JERARQUÍA

EVENTUALIDAD
ENCAPSULACIÓN
MÉTODOLÓGÍA
JERARQUÍA

ABSTRACCIÓN
ENCAPSULACIÓN
MODULARIDAD
JERARQUÍA

PROPIEDADES
FUNCIONES
MÉTODOS
VARIABLES

3.- La correcta declaración de una variable en ActionScript 3.0 es: (seleccionar con una X)

Var a:int= 6; var:int=8:a; var a=6;

4.-Los eventos flash se declaran de la siguiente forma:

```
objetoOrigen.addEventListener(TipoEvento.NOMBRE_EVENTO, respuestaEvento);  
function respuestaEvento(variableEvento:TipoEvento):void  
{  
    // Acciones a realizar en  
    //respuesta al evento.  
}
```

```
objetoOrigen.addEventListener(NOMBRE_EVENTO TipoEvento);  
function NOMBRE_EVENTO(variableEvento:TipoEvento):  
{  
    // Acciones a realizar en  
    //respuesta al evento.  
}
```

5.-Las funciones en AS3 sirven para:

- a.) Crear bloques de código extensos entre llaves
- b.) Crear bloques de código reutilizables
- c.) Bloques con tareas específicas
- d.) Todas
- e.) Ninguna

6.-Responda con V (verdadero) y F (falso)

ActionScript3.0 utiliza Loops (ciclos de repetición) _____

ActionScript3.0 tiene como estructuras condicionales *if, if else, switch (case)*_____

ActionScript3.0 posee normas respecto a palabras reservadas _____

Se pueden programar filtros para modificación interactiva de una imagen _____

7.-Desarrolle el siguiente ejercicio, con el uso de AS3.0

Mediante un botón inserte cuatro movieclips animados de una misma clase, mismos que se animen cuando el mouse se ubique sobre ellos.

Creación de un botón y sus estados

Creación del objeto mc animado

Crear el objeto mc e instanciarlo

Código:

Declaración de variables

Manejo de ciclos

Declaración de funciones

Declaración de eventos

Manejo addChild

Manejo Propiedades del mc

(ANEXO 3)

**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
ESCUELA DE DISEÑO GRÁFICO**

El objetivo de la presente encuesta es determinar el nivel usabilidad del prototipo de multimedia presentado a usted, con los contenidos del ActionScript 3.0

Se recomienda llenar los casilleros de acuerdo a su valoración hacia el multimedia entregado.

1.- AMIGABILIDAD DE LA INTERFACE DE USUARIO:

	SI	MEDIANA MENTE	POCO	NADA
La interface de usuario no opone ninguna resistencia a la navegación				
Se puede regresar con facilidad al menú principal				
El acceso a los distintos menús es fácil				
Los iconos son suficientemente grandes y fáciles de seleccionar				
Los aspectos gráficos de las pantallas y del menú son agradables				
Los sonidos son adecuados, además de agradables				
Se identifican fácilmente los niveles de navegación				
EL multimedia permite ir a todo punto de la aplicación en cualquier momento				
Puede salir del multimedia cuando lo desee				

1.- CONTENIDOS ABORDADOS:

	SI	MEDIANA MENTE	POCO	NADA
La información que contiene es relevante				
La información que contiene está bien organizada				
La información no se transmite de forma monótona				
Se percibe la relación entre el conjunto de mensajes				
EL multimedia permite desarrollar habilidades y destrezas específicas del tema que trata				
Se ejercita suficientemente los contenidos para facilitar el afianzamiento de lo aprendido				
Plantea o insinúa estudios de casos o problemas que el usuario podría desarrollar después				

(ANEXO 4)

CÓDIGO FUENTE:

BOTONES PANEL INFERIOR

```
1  import flash.events.MouseEvent;
2  import flash.display.Stage;
3  import flash.display.MovieClip;
4  import flash.media.Sound;
5  import flash.net.URLRequest;
6  import flash.display.StageDisplayState;
7
8  saltar1_btn.addEventListener(MouseEvent.CLICK, saltar);
9
10 function saltar(e:MouseEvent):void
11 {
12     gotoAndPlay("finintro");
13 }
14
15 audio1_btn.addEventListener(MouseEvent.CLICK, irAcancion);
16 cancion();
17 var mus:Boolean=false;
18
19 function irAcancion (e:MouseEvent):void
20 {
21     if (mus==true){
22         cancion();
23         trace ("false");
24         mus=false;
25     }
26     else{
27         SoundMixer.stopAll();
28         trace ("true");
29         mus=true;
30     }
31 }
32
33 function cancion ():void
34 {
35     var ruta:URLRequest=new URLRequest();
36     ruta.url="Space.mp3";
37     var musicaFondo:Sound=new Sound();
38     musicaFondo.load(ruta);
39     musicaFondo.play();
40 }
41
42 salir1_btn.addEventListener(MouseEvent.CLICK, salir)
43 function salir(evento:MouseEvent):void
44 {
45     fscommand("quit")
46 }
```

CÓDIGO MENÚ DE CONTENIDOS

```
1 import fl.motion.MotionEvent;
2 import flash.display.Sprite;
3 import flash.display.MovieClip;
4 import flash.events.MouseEvent;
5 import fl.video.FLVPlayback;
6 stop();
7
8 // _____ flash _____
9 flash1_btn.addEventListener(MouseEvent.CLICK, interfaz);
10 function interfaz(e:MouseEvent):void
11 {
12     gotoAndPlay("a1");
13     anim1_mc.gotoAndPlay(2);
14     SoundMixer.stopAll();
15 }
16 flash1_btn.addEventListener(MouseEvent.MOUSE_OUT, fuera);
17 function fuera(e:MouseEvent):void
18 {
19     anim1_mc.gotoAndPlay(16);
20 }
21 /* _____ Botón POO _____ */
22 pool1_btn.addEventListener(MouseEvent.CLICK, interfaz2);
23 function interfaz2(e:MouseEvent):void
24 {
25     gotoAndPlay("a2");
26     anima2_mc.gotoAndPlay(2);
27     SoundMixer.stopAll();
28 }
29 pool1_btn.addEventListener(MouseEvent.MOUSE_OUT, fuera2);
30 function fuera2(e:MouseEvent):void
31 {
32     anima2_mc.gotoAndPlay(16);
33 }
34 // _____ Intro a AS3 _____
35 introducl_btn.addEventListener(MouseEvent.CLICK, interfaz3);
36 function interfaz3(e:MouseEvent):void
37 {
38     gotoAndPlay("a3");
39     SoundMixer.stopAll();
40     anima3_mc.gotoAndPlay(2);
41 }
42 introducl_btn.addEventListener(MouseEvent.MOUSE_OUT, fuera3);
43 function fuera3(e:MouseEvent):void
44 {
45     anima3_mc.gotoAndPlay(16);
46 }
```

```

47 // _____ Estructuras _____
48 estructuras1_btn.addEventListener(MouseEvent.CLICK, interfaz4);
49 function interfaz4(e:MouseEvent):void
50 {
51     gotoAndPlay("a4");
52     SoundMixer.stopAll();
53     anima4_mc.gotoAndPlay(2);
54 }
55 estructuras1_btn.addEventListener(MouseEvent.MOUSE_OUT, fuera4);
56 function fuera4(e:MouseEvent):void
57 {
58     anima4_mc.gotoAndPlay(16);
59 }
60 // _____ Eventos _____
61 eventos1_btn.addEventListener(MouseEvent.CLICK, interfaz5);
62 function interfaz5(e:MouseEvent):void
63 {
64     gotoAndPlay("a5");
65     SoundMixer.stopAll();
66     anima5_mc.gotoAndPlay(2);
67 }
68 eventos1_btn.addEventListener(MouseEvent.MOUSE_OUT, fuera5);
69 function fuera5(e:MouseEvent):void
70 {
71     anima5_mc.gotoAndPlay(16);
72 }
73 // _____ Objetos _____
74 objetos1_btn.addEventListener(MouseEvent.CLICK, interfaz6);
75 function interfaz6(e:MouseEvent):void
76 {
77     gotoAndPlay("a6");
78     SoundMixer.stopAll();
79     anima6_mc.gotoAndPlay(2);
80 }
81 objetos1_btn.addEventListener(MouseEvent.MOUSE_OUT, fuera6);
82 function fuera6(e:MouseEvent):void
83 {
84     anima6_mc.gotoAndPlay(16);
85 }
86 // _____ Cargar elementos externos _____
87 cargar1_btn.addEventListener(MouseEvent.CLICK, externos)
88 function externos(e:MouseEvent):void
89 {
90     gotoAndPlay("a7");
91     SoundMixer.stopAll();
92 }

```

```

93 //_____Prog Avanzada_____
94 Progav_btn.addEventListener(MouseEvent.CLICK,Avanzada)
95 function Avanzada(e:MouseEvent):void
96 {
97     gotoAndPlay("a8");
98     SoundMixer.stopAll();
99 }
100 //_____HOME_____
101 home1_btn.addEventListener(MouseEvent.CLICK,home)
102 function home(evento:MouseEvent):void
103 {
104     gotoAndPlay("intro");
105     SoundMixer.stopAll();
106 }
107 //_____PDF_____
108 pdf_btn.addEventListener(MouseEvent.CLICK,pdf)
109 function pdf (evento:MouseEvent):void
110 {
111     navigateToURL(new URLRequest("PDF AS3.pdf"), "_blank");
112 }

```

AS3 : 120

CÓDIGO PANTALLA DE BIENVENIDA:

```

1  var texto1:tex1_txt=new tex1_txt();
2
3      SoundMixer.stopAll();
4
5      var video:Video = new Video(700,480);
6      addChild(video);
7      video.x=300;
8      video.y=10;
9
10     var ncc:NetConnection = new NetConnection();
11     ncc.connect(null);
12
13     var nss:NetStream = new NetStream(ncc);
14     video.attachNetStream(nss);
15
16     var listener:Object = new Object();
17     listener.onMetaData = function(evt:Object):void {};
18     nss.client = listener;
19
20
21     reproductor.source="INSERTAR/vid1.flv";
22     reproductor.fullScreenTakeOver = false;
23
24     texto1.x=600;
25     texto1.y=550;
26
27     addChild(texto1);
28

```

CÓDIGO PANTALLA BOTÓN FLASH CS6:

```
1 import fl.motion.MotionEvent;
2 import flash.net.URLLoader;
3 import flash.net.URLRequest;
4 import flash.events.Event;
5 import flash.text.TextField;
6 import flash.events.MouseEvent;
7
8 interfaz1_btn.mouseChildren = true;
9 var textoll:tex2_txt=new tex2_txt();
10 interfaz1_btn.addEventListener(MouseEvent.ROLL_OVER, quieto);
11 function quieto(e:MouseEvent):void
12 {
13     gotoAndStop(15);
14 }
15 interfaz1_btn.addEventListener(MouseEvent.MOUSE_OUT, fuera2);
16 function fuera2(e:MouseEvent):void
17 {
18     gotoAndPlay(16);
19 }
20 interfaz1_btn.addEventListener(MouseEvent.CLICK, video);
21 function video (e:MouseEvent):void
22 {
23     SoundMixer.stopAll();
24     var vid:Video = new Video(700,480);
25     addChild(vid);
26     vid.x=170;
27     vid.y=-190;
28     var nc:NetConnection = new NetConnection();
29     nc.connect(null);
30     var ns:NetStream = new NetStream(nc);
31     vid.attachNetStream(ns);
32     var listener:Object = new Object();
33     listener.onMetaData = function(evt:Object):void {};
34     ns.client = listener;
35     reproductor.fullScreenTakeOver = false;
36     reproductor.source="INSERTAR/vid2.flv";
37 }
38 }
39 interfaz1_btn.addEventListener(MouseEvent.CLICK, texto);
40 function texto (e:MouseEvent):void
41 {
42     textoll.x=480;
43     textoll.y=370;
44     addChild(textoll);
45 }
```

CÓDIGO PANTALLA BOTÓN STANDARD

```
1 import fl.video.FLVPlayback;
2 ventajas_btn.mouseChildren = true;
3 fundamentos_btn.mouseChildren= true;
4 conceptos_btn.mouseChildren= true;
5
6 function quieto(e:MouseEvent):void
7 { gotoAndStop(15); }
8 function fuera(e:MouseEvent):void
9 { gotoAndPlay(16); }
10 // _____CONCEPTOS_____
11 var texto14:tex4_txt=new tex4_txt();
12 function video3(e:MouseEvent):void
13 {
14     SoundMixer.stopAll();
15     var vid3:Video = new Video(700,480);
16     addChild(vid3);
17     vid3.x=30;
18     vid3.y=-220;
19     var nc3:NetConnection = new NetConnection();
20     nc3.connect(null);
21     var ns3:NetStream = new NetStream(nc3);
22     vid3.attachNetStream(ns3);
23     var listener3:Object = new Object();
24     listener3.onMetaData = function(evt:Object):void {};
25     ns3.client = listener3;
26     reproductor.source="INSERTAR/vid5.flv";
27     reproductor.fullScreenTakeOver = false;
28     texto14.x=380;
29     texto14.y=360;
30     addChild(texto14);
31
32     function borrar3 (e:MouseEvent)
33     {
34         removeChild (texto14);
35         removeChild (vid3);
36     }
37 ventajas_btn.addEventListener(MouseEvent.CLICK, borrar3);
38 fundamentos_btn.addEventListener(MouseEvent.CLICK, borrar3);
39 }
40 conceptos_btn.addEventListener(MouseEvent.ROLL_OVER, quieto);
41 conceptos_btn.addEventListener(MouseEvent.MOUSE_OUT, fuera);
42 conceptos_btn.addEventListener(MouseEvent.CLICK, video3);
```

BIBLIOGRAFÍA

1. **ÁLVAREZ, O.**, La enseñanza virtual en la educación superior., 3er. Ed., Bogotá – Colombia., Secretaría General-Procesos Editoriales., 2002., Pp. 5-7.
2. **BARTOLOMÉ, A.**, Medios y recursos interactivos., 4ª. Ed., Madrid - España., Marfil., 1995 Pp. 20-27.
3. **BRAUNSTEIN, R. Y OTROS.**, Action Script 3.0 Bible., 1ra Ed., Indiana – Estados Unidos de América., Wiley Publishing., 2006., Pp. 50-90.
4. **CHURA, D.**, Action Script 3.0 advance., 1ra Ed., Lima – Perú., Empresa Editora Macro., 2010., Pp. 34-35.
5. **CORDÓN, O.**, Enseñanza virtual: Fundamentos, perspectivas actuales y visión., 1ra Ed., Granada – España., Universidad de Granada., 2005., Pp. 20-27.

6. **HERAS, C. Y OTROS.**, ActionScript 3.0 para Flash CS3., 2da Ed., Madrid – España., Editorial Anaya., 2007., Pp. 215-270.

7. **JOYANES, L. y SÁNCHEZ L.**, Programación en C++., 1ra Ed., Madrid – España., McGrawHill., 2006., Pp. 30-35.

8. **KASHIWAMOTO, J. Y OTROS.**, Curso Programación Orientada a Objetos., 1ra Ed., México – México., Universidad Autónoma de México., 2004., Pp. 11-48.

9. **MESTRE, U. Y OTROS.**, Entornos virtuales de aprendizaje., 1era Ed., Ciudad de las Tunas – Cuba., Editorial Universitaria., 2007., Pp. 9-17.

10. **OLEA J. y PONSODA, V.**, Test adaptativos informatizados., 1era Ed., Madrid-España., Ediciones UNED., 2003., Pp. 13-20.

11. **OÑATE L.**, La metodología PACIE., 2da Ed., Quito – Ecuador., Fundación para la Actualización Tecnológica de latinoamérica., 2009., Pp. 4-11, 60-65.

- 12. ORIHUELA, J. y SANTOS, M.,** Introducción al diseño digital., 4ta Ed., Madrid – España., Editorial Anaya., 1999., Pp. 20-34.

- 13. POLEE, B.,** Tecnología Educativa - Aportes de la multimedia al proceso de enseñanza y aprendizaje en el módulo de salud integral., 1era Ed., Barcelona – España., McGraw-Hill., 1999., Pp. 56-60.

- 14. RICH, Sh. y ZEVAN, R.,** Learning ActionScript 3.0., 2da Ed., Ontario – Canadá., O’Reilly., 2007., Pp. 3-111.

- 15. UNIGARRO, M.,** Educación virtual, encuentro formativo con el ciberespacio., 2da Ed., Bucaramanga – Colombia., UNAB., 2001., Pp. 27-118.

- 16. VAUGHAN, T.,** Multimedia Making It Work., 8va Ed., Los Angeles – USA., McGraw-Hill., 2010., Pp. 1-10; 196-240.

- 17. ADELLE, J.,** Internet en el aula., Revista Electrónica de Tecnología Educativa., Castellón de la Plana – España., Vol. 1., Pp. 1-16.

18. **LEFLORE, D.**, Theory supporting design guidelines for web-based instruction Beverly Abbey., Instructional and Cognitive Impacts of Web-Based Education., Hershey, PA – USA., Idea Group Publishing., 2000., N° 9., Vol. 1., Pp. 10-15.

19. **MARTÍNEZ, R.**, Importancia relativa de algunos factores del rendimiento educativo., Edicación, Eficiencia y equidad., Santiago de Chile – Chile., Ediciones SUR., 2008., N° 3., Vol. 1., Pp. 7.

20. **MIRAVALES, M. Y OTROS.**, Revista de Universidad y Sociedad del Conocimiento., De la educación a distancia la educación virtual., Barcelona – España., Vol. 2., N° 4., UOC., 2006., Pp. 3-8.

21. **NUÑEZ, T.**, Eductec-e. Revista Electrónica de Tecnología Educativa., Entornos virtuales de enseñanza aprendizaje (EVEA)., Maracay – Venezuela., Vol. 2., N° 7., Universidad Nacional Experimental Politécnica de la FUerza Armada., 2011., Pp. 6-11.

22. **MANCERO, P. y TOAPANTA, M.**, Sistema virtual de tutoría académica para la materia Programación II., Facultad de Ingeniería de Sistemas., Escuela de Ingeniería en Sistemas Informáticos y de Computación., Escuela Politécnica Nacional., **TESIS.**, Quito – Ecuador., 2010., Pp. 1-23.

23. ACTION SCRIPT 3.0

- <https://sites.google.com/site/as3gamedev/Home/actionscript-3-api>

2013 – 05 – 25.

- <http://circuloycuadrado.com/programacion-basica-actionscript-3>

2013 – 06 – 07.

- http://help.adobe.com/es_ES/as3/learn/as3_learning.pdf

2013 – 06 – 05.

24. DISEÑO GRÁFICO

- <http://www2.fe.ccoo.es/andalucia/docu/p5sd7586.pdf>

=

2013 – 05 – 25.

- <http://www.monografias.com/trabajos84/fundamentos-del-diseno-grafico-preparacion-tecnicos-medios>

2013 – 04 – 25.

25. ENTORNOS VIRTUALES DE EDUCACIÓN

- <http://www.uv.es/bellochc/pdf>

2013 – 02 – 23.

- <http://antia.fis.usal.es/sharedir/TOL/herramientasAutor/index.html>

2013 – 03 – 17.

- [http://www.ugr.es/~sevimeco/revistaeticanet/Numero2/Articulos/Posibilidades+y+limitaciones+de+Internet\[1\].pdf](http://www.ugr.es/~sevimeco/revistaeticanet/Numero2/Articulos/Posibilidades+y+limitaciones+de+Internet[1].pdf)

2013 – 04 – 10.

26. MULTIMEDIA

- <http://www.aulafacil.com/cursosenviados/cursomulti-mediaaprendizaje/Lecc-2.htm>.

2013 – 05 – 25.

- <http://dis.um.es/~jfernand/0405/tsm/tema1.pdf>

2013 - 06 - 04.

- http://www.imaginar.org/webmaster/index_archivos/Guia/guia_usabilidad.pdf.

2013 – 05 – 12.

27. PROGRAMACIÓN ORIENTADA A OBJETOS

- <http://luis.izqui.org/resources/ProgOrientadaObjetos.pdf>.

2013 – 04 – 26.

- <http://www.monografias.com/trabajos-pdf/apuntes-fundamentos-programacion-uno>

2013 – 04 – 08.

28. TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN (TIC)

- http://www.dgde.ua.es/congresotic/public_doc/pdf

2012 – 13- 19.

- <http://www.explored.com.ec/noticias-ecuador/las-cifras-del-uso-de-internet-crecen-en-el-paa-s-535452.html>

2013 – 01 – 17.

