

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

Facultad de Contaduría y Administración

Facultad de Ciencias Químicas e Ingeniería

Maestría en Tecnologías de la Información y la Comunicación



Desarrollo de Objetos de Aprendizaje Móviles para Apoyar la Enseñanza de Programación Estructurada

Tesis para obtener el grado de:

MAESTRO EN TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN

Presenta:

MÁRQUEZ GARCÍA YADHIRA ISABEL

Bajo la dirección de:

Dra. Carelia Guadalupe Gaxiola Pacheco

Codirección:

M.C Marco Antonio Pinto Ramos

Tijuana, Baja California, Noviembre del 2016

Hoja en blanco para sustituirla por la aprobación de impresión de tesis...

Agradecimientos

A Dios por haberme dado salud y guía para poder llegar hasta este punto.

A mi esposo Daniel por su ayuda y apoyo incondicional para poder cumplir esta meta. Por su gran amor que ayudo a motivarme y seguir adelante en los momentos que estuve a punto de desistir.

A mis padres por haberme apoyado en todo momento. Por sus lecciones de vida; con esfuerzo, mucha perseverancia y confianza en uno mismo, se pueden lograr las metas que uno se propone en la vida y ser cada día mejor persona.

A mi directora de tesis la Dra. Carelia Gaxiola, por su guía y conocimiento. Me ha hecho fácil lo difícil. Ha sido un privilegio contar con su apoyo durante este proceso.

A las coordinadoras de la MTIC por su atención, amabilidad y apoyo en todo lo referente a mi vida como alumno de la maestría. A los maestros por compartir sus experiencias y conocimientos para hacer de mí un mejor profesionalista.

A mi familia y a todos aquellos que siguen cerca de mí y que le regalan a mi vida algo de ellos permitiéndome ser mejor persona día con día.

Resumen

El desarrollo y uso de las aplicaciones móviles ha incrementado exponencialmente en los últimos años, debido principalmente, a la reducción del costo de adquisición del equipo y a la facilidad de uso de sus aplicaciones, lo que permite que un mayor número de usuarios utilicen el dispositivo móvil no solo para comunicarse. Incorporar aspectos educativos a los objetos de aprendizaje permite ofrecer nuevas alternativas didácticas, por lo que el presente trabajo de tesis explora esta área de oportunidad y brinda 7 objetos de aprendizaje móviles desarrollados en una aplicación para dispositivos móviles con sistema operativo Android. Estos objetivos de aprendizaje móviles podrán ser utilizados por los alumnos de la Facultad de Ciencias Químicas e Ingeniería, para apoyar su proceso de enseñanza-aprendizaje de la materia de Programación Estructurada.

Índice General

Capítulo 1.- Introducción	9
1 Introducción	10
1.1 Motivación	10
1.2 Objetivo General	11
1.3 Objetivos Específicos.....	11
1.4 Alcances de la Tesis.....	12
1.5 Organización de la Tesis	12
Capítulo 2.- Marco Teórico	14
2 Marco Teórico	15
2.1 Proceso de enseñanza-aprendizaje de Programación en la Ingeniería	15
2.1.1 Tecnologías de la información y la comunicación.....	18
2.2 Los objetos de aprendizaje	23
2.2.2 Recursos educativos: Objetos de aprendizaje	25
2.3 Ingeniería del Software	27
2.3.1 Definición de la Ingeniería de Software.....	27
2.3.2 Objetivos de la Ingeniería del Software	30
2.3.3 Ciclos de vida.....	31
2.4 Dispositivos y Aplicaciones móviles	38
2.4.1 Tipos de dispositivos móviles	39
2.5 Edición e implementación de objetos de aprendizaje móviles.....	41
2.5.1 Software con licencia	41
2.5.2 Software libre.....	43
2.5.3 Características de hardware para las aplicaciones móviles	46
2.5.4 Seguridad para aplicaciones móviles	47
2.5.5 Software de desarrollo más populares.....	48
Capítulo 3.- Metodología	52
3 Metodología para el desarrollo de Objetos de Aprendizaje y su adaptación en entornos móviles.....	53
3.1 Análisis	54

3.2	Diseño	54
3.3	Desarrollo.....	54
3.4	Implementación y Evaluación.....	55
Capítulo 4.– Desarrollo.....		56
4	Caso de estudio: Objetos de Aprendizaje en entornos móviles para el apoyo a la enseñanza de Programación Estructurada	57
4.1	Análisis	57
4.2	Diseño	57
4.3	Desarrollo.....	58
4.4	Implementación y Evaluación.....	65
Capítulo 5.– Pruebas y Resultados		66
5	Funcionamiento de los prototipos de objetos de aprendizaje	67
Capítulo 6.– Conclusiones		77
6	Conclusiones	78
Referencias.....		79
Anexo A – Reactivos de Examen Colegiado		88
Anexo B – Diagramas UML		103
Anexo C – GameMaker: Studio.....		106

Índice de Tablas

Tabla 1 - Diferencias entre entornos centrados en el profesor y centrados en el alumno. Tomada de (UNESCO, 2004, pág. 27)	17
Tabla 2 - Cambios en el rol del docente. Tomada de (UNESCO, 2004, pág. 28)	17
Tabla 3 - Comparación entre modelos de ciclo de vida (SCHACH, 2006, pág. 59)	38
Tabla 4 - Tabla de resultados y reactivos de examen colegiado de Programación del ciclo 2012-2	89

Índice de Figuras

Figura 1 - 4'p de la Ingeniería de Software (BRAUDE, 2003, pág. 3)	28
Figura 2 - Estratos de la Ingeniería del Software (PRESSMAN, 2006, pág.24)	29
Figura 3 - Modelo en Cascada (BRAUDE, 2003, pág.25).....	32
Figura 4 - Modelo en cascada con más detalle.(BRAUDE, 2003, pág. 26).....	33
Figura 5 - Modelo del ciclo de vida de espiral (WEITZENFELD, 2005, Pág.53)	34
Figura 6 - Modelo del ciclo de vida de codificación y ajuste.(SCHACH, 2006, pág.48).....	35
Figura 7 - Modelo del ciclo de vida de elaboración rápida de un prototipo.(SCHACH, 2006, pág. 51)....	36
Figura 8 – Smartphone o teléfono inteligente http://www.clipartpanda.com/clipart_images/you-can-download-the-clip-art-1104135)	39
Figura 9 - Venta de teléfonos inteligentes a usuarios finales desde el primer cuarto del 2009 al tercer cuarto del 2014 (Global smartphone sales to end users from 1st quarter 2009 to 3rd quarter 2014, by operating system (in million units),2015)	49
Figura 10 - Concepto del Modelo ADDIE (MARIBE, 2009).....	53
Figura 11 - Proceso de Metodología SCRUM (https://commons.wikimedia.org/wiki/File:Ficha_scrum.jpg).....	58
Figura 12 – GameMaker: Objetos principales: el Fox, el Robot, el Diskette (Fuente Propia)	60
Figura 13 – GameMaker: Programación del objeto Robot (Fuente Propia)	61
Figura 14 – GameMaker: Programación de movimiento (Fuente Propia).....	61
Figura 15 – GameMaker: Programación de objetos (Fuente Propia).....	62
Figura 16 – GameMaker: Interfaz gráfica de menú principal (Fuente Propia)	62
Figura 17 – GameMaker: Diseño de nivel 1 (Fuente Propia)	63
Figura 18 – GameMaker: Nivel 1 Secuencia de Instrucciones (Fuente Propia)	63
Figura 19 – GameMaker: Menú para elegir el nivel a jugar (Fuente Propia)	64
Figura 20 – GameMaker: Menú principal (Fuente Propia).....	64
Figura 21 – GameMaker: Seleccionar el tipo de sistema operativo Android (Fuente Propia)	65
Figura 22 – GameMaker: Guardar archivo en formato .apk (Fuente Propia)	65
Figura 23 – GameMaker: Archivo ejecutable (.apk) (Fuente Propia).....	65
Figura 24 - Información del teléfono inteligente utilizado para pruebas (Fuente Propia)	67
Figura 25 - Información de la tableta electrónica utilizado para pruebas (Fuente Propia)	67
Figura 26 - Descarga de aplicación en el teléfono inteligente Motorola modelo Moto G (Fuente Propia).....	68

Figura 27 - Instalación de aplicación en el teléfono inteligente Motorola modelo Moto G (Fuente Propia)	68
Figura 28 - Pantalla principal (Fuente Propia)	69
Figura 29 – Pantalla de Ayuda: Instrucción Avanza (Fuente Propia)	69
Figura 30 - Pantalla de Ayuda: Instrucción Rota Derecha (Fuente Propia)	70
Figura 31 - Pantalla de Ayuda: Instrucción Rota Izquierda (Fuente Propia)	70
Figura 32 - Pantalla de Ayuda: Instrucción Recoge Disquete (Fuente Propia)	70
Figura 33 - Pantalla de Ayuda: Instrucción Activa Robot (Fuente Propia)	71
Figura 34 – Pantalla de información de la aplicación (Fuente Propia)	71
Figura 35 – Menú para seleccionar el nivel a jugar (Fuente Propia)	72
Figura 36 - Nivel 1 : Secuencia de Instrucciones (Fuente Propia)	72
Figura 37 – Nivel 2 : Nivel 3 : Secuencia de Instrucciones (Fuente Propia)	73
Figura 38 - Nivel 3 : Secuencia de Instrucciones (Fuente Propia)	73
Figura 39 – Nivel 3 : Lógica de Instrucciones (Fuente Propia)	74
Figura 40 – Nivel 6 – Ciclo For (Fuente Propia)	74
Figura 41 – Nivel 6 : Ciclo For con instrucciones (Fuente Propia)	75
Figura 42 – Nivel 6 : Ciclo While (Fuente Propia)	75
Figura 43 – Nivel 6 : Ciclo Do (Fuente Propia)	75
Figura 44 – Nivel 7: Procedimiento (Fuente Propia)	76
Figura 45 – Nivel 7 : Procedimiento con instrucciones (Fuente Propia)	76
Figura 46 - Formato para elaborar especificaciones de reactivos – Reactivo #7	91
Figura 47 - Formato para elaborar especificaciones de reactivos – Reactivos #14 y 15	92
Figura 48 - Formato para elaborar especificaciones de reactivos – Reactivo #20	93
Figura 49 - Formato para elaborar especificaciones de reactivos – Reactivo #21	95
Figura 50 - Formato para elaborar especificaciones de reactivos – Reactivo #26	97
Figura 51 - Formato para elaborar especificaciones de reactivos – Reactivo #28	99
Figura 52 - Formato para elaborar especificaciones de reactivos – Reactivo #29	102
Figura 53 - Diagrama de casos de uso para aplicación móvil El Fox & Robot (Fuente Propia)	103
Figura 54 - Diagrama de casos de uso de niveles de la aplicación móvil El Fox & Robot (Fuente Propia)	104
Figura 55 – Diagrama de Casos de uso de como jugar la aplicación móvil El Fox & Robot (Fuente Propia)	105
Figura 56 - Diagrama de casos de uso para ver la información de la aplicación móvil El Fox y el Robot (Fuente Propia)	105

Figura 57 - Sprites y Objetos utilizados en la aplicación móvil El Fox & Robot (Fuente Propia)	106
Figura 58 - Objeto Main: Código de evento “crear” (Fuente Propia).....	107
Figura 59 - Objeto Main: Código de procedimiento en evento “crear” (Fuente Propia)	107
Figura 60 - Objeto Main: Código de evento “paso” (Fuente Propia).....	108
Figura 61 - Objeto Main: Código de procedimiento en evento “paso” (Fuente Propia).....	108
Figura 62 - Objeto Fox: Programación de función del objeto en evento “crear” (Fuente Propia).....	109
Figura 63 - Objeto Fox: Código de procedimiento en evento “paso” (Fuente Propia).....	109
Figura 64 - Objeto Robot: Código de procedimiento en evento “paso” (Fuente Propia)	110
Figura 65 - Objeto Obstáculo: Programación de función del objeto en evento “Objeto Fox” (Fuente Propia).....	110
Figura 66 - Objeto Diskette: Programación de función del objeto en evento “Objeto Fox” (Fuente Propia).....	111
Figura 67 - Objeto Tutorial: Programación de función del objeto en evento “paso” (Fuente Propia).....	111
Figura 68 - Selección de Sistema Operativo para exportar la aplicación (Fuente Propia).....	112
Figura 69 - Archivo para dispositivos con Sistema Operativo Andorid (*.apk) (Fuente Propia).....	112

Capítulo 1.- Introducción

1 Introducción

El uso de la tecnología implementada en el proceso de enseñanza – aprendizaje, ha estado funcionando desde hace varios años en México y en el mundo. Se han desarrollado plataformas e-learning que permiten a los catedráticos colocar el contenido de sus cursos, que los estudiantes pueden acceder desde su computadora personal e interactuar con sus actividades a distancia, lo que permite a las instituciones educativas cubrir un mayor número de estudiantes. (SOTELO, 2008).

Uno de los mayores retos en el desarrollo de Objetos de Aprendizaje, es que estos sean comprendidos, aceptados y principalmente utilizados por los estudiantes en cualquier parte del mundo lo que conlleva aparte el uso de una metodología instruccional adecuada para el desarrollo de contenidos incluir una metodología multimedia bajo los principios de usabilidad que permita crear aplicaciones que estén acordes a los estilos de aprendizaje, personalidad y habilidades de los estudiantes (GALEANA, 2012).

1.1 Motivación

La Facultad de Ciencias Químicas e Ingeniería se ha visto en la necesidad de implementar dispositivos tecnológicos y aplicaciones web aplicados al área de conocimiento para brindar mayor soporte al aprendizaje del alumnado, tal es el caso de la pizarrones electrónicos, proyectores y el apoyo de algunas aplicaciones en línea.

La mayoría de los alumnos toman notas en libretas, pero no es de sorprender que cada vez más alumnos prefieran utilizar la computadora portátil para este caso, o también que tomen fotografías de los pizarrones con la información impartida en clase en lugar de ser escrita a mano, inclusive hay quienes prefieren grabar la clase para su posterior consulta y estudio.

El uso de las aplicaciones móviles ha ido incrementado exponencialmente (INEGI, 2013), por lo que el presente trabajo de tesis explora esta área de oportunidad y 7 objetos de aprendizaje móviles dentro de una aplicación móvil que pueden ser utilizados por los alumnos de la Facultad de Ciencias Químicas e Ingeniería que utilicen dispositivos móviles con sistema operativo

Android. Los objetos de aprendizaje están diseñados de tal manera que apoyen el aprendizaje de los temas con mayor índice de reprobación de la asignatura de Programación. De acuerdo a los exámenes colegiados realizados desde el año 2012 al final del semestre a todo el alumnado de Programación, los temas con mayor índice de reprobación son: algoritmos, funciones, estructuras de control de selección y estructuras de control de iteración.

Para la realización de este trabajo se tomó como base el estudio previamente realizado donde se comprueba una aceptación del 82% de los 17 alumnos encuestados sobre su experiencia con los objetos de aprendizaje desarrollados previamente en la tesis (Palacios, 2008).

1.2 Objetivo General

El objetivo general de ésta tesis consiste en el diseño e implementación de objetos de aprendizaje móviles para apoyar el proceso de enseñanza-aprendizaje de las estructuras de control básicas de la Programación.

1.3 Objetivos Específicos

- Diseñar e implementar una aplicación móvil de la asignatura de Programación que apoye a reforzar los temas de: Algoritmos, Funciones, Estructuras de Control de Selección y Estructuras de Control de Iteración, que se imparten a los alumnos en dicha asignatura.
- Generar una herramienta que sirva al maestro como apoyo al impartir el curso de Programación.

1.4 Alcances de la Tesis

El avance en la tecnología ha permitido que la comunicación y realización de actividades de la vida diaria, laboral y educativa sean ejecutadas en menor costo y tiempo. Al igual que la tecnología va desarrollándose, la forma de impartir y adquirir el conocimiento ha evolucionado, adecuándose al medio y a los recursos, por ésta razón se considera importante que la Facultad de Ciencias Químicas e Ingeniería de la Universidad Autónoma de Baja California se encuentre a la vanguardia con los recursos educativos.

El presente trabajo de tesis aporta un conjunto de objetos de aprendizaje móviles desarrollados específicamente para la asignatura de Programación, que apoya a profesores y alumnos en el proceso de enseñanza-aprendizaje en temas contenidos en dicha asignatura, se presenta una alternativa de estudio innovador que ayuda a mejorar la competencia de los alumnos.

1.5 Organización de la Tesis

Este documento de tesis está organizado en 6 capítulos:

- El primer capítulo presenta la introducción, donde se explica la motivación de este trabajo, el objetivo, las metas, el alcance y la estructura de este documento.
- En el capítulo dos se ofrece el marco teórico de la investigación realizada, iniciando con el proceso de enseñanza-aprendizaje de Programación en la Ingeniería, después con definición y objetivos de la ingeniería de Software, siguiendo con las aplicaciones móviles que existen para Programación en Ingeniería, y finalizando con la edición e implementación de objetos de aprendizaje móviles.
- En el capítulo tres se presenta la metodología propuesta para el desarrollo de los objetos de Aprendizaje y su adaptación en entornos móviles, se explica el funcionamiento de cada una de sus fases.
- En el capítulo cuatro se expone un caso de estudio, es una aplicación móvil con 7 diferentes ejercicios de lógica que los estudiantes pueden utilizar como apoyo para el aprendizaje de Programación Estructurada.

- En el capítulo cinco se expone el funcionamiento de la aplicación en un teléfono móvil en un teléfono inteligente modelo Moto G (2da generación) con sistema operativo Android versión Lollipop (5.0.2) y los resultados obtenidos.
- En el capítulo seis se presenta una discusión acerca de la conclusión de la presente investigación.

Capítulo 2.– Marco Teórico

2 Marco Teórico

En la Facultad de Ciencias Químicas e Ingeniería de la Universidad Autónoma de Baja California actualmente se imparten 6 programas educativos a nivel licenciatura y un Tronco Común del área de Ingeniería, la asignatura de Programación se imparte a los alumnos de Tronco Común de Ciencias de la Ingeniería.

A continuación se analizarán las formas de enseñanza-aprendizaje aplicadas a la Programación y también algunos problemas presentados para el estudio de la misma.

2.1 Proceso de enseñanza-aprendizaje de Programación en la Ingeniería

Es de fundamental importancia describir las terminologías correspondientes a la enseñanza-aprendizaje.

- Enseñar: “Instruir, doctrinar, amaestrar con reglas o preceptos: enseñar a leer, enseñar filosofía.” (QUILLET, DICCIONARIO ENCICLOPÉDICO QUILLET, 1988, pág. 40)
- Enseñanza: “Acción y efecto de enseñar. Sistema y método de dar instrucción. Ejemplo que sirve de experiencia.” (QUILLET, pág. 40) Aprender: “Adquirir el conocimiento de una cosa por medio del estudio o de la experiencia. Tomar algo en la memoria.” (QUILLET, pág. 383)
- Aprendizaje: “Acción de aprender algún arte u oficio.” (QUILLET, pág. 383)

Los seres humanos aprendemos por naturaleza, y cada persona aprende a su ritmo y manera, el aprendizaje se lleva a cabo en un buen entorno que proporcione los materiales necesarios y con los profesores que apoyen y estimulen a los alumnos. Los estudiantes hacen uso de distintas formas de aprendizaje para adquirir el conocimiento, a continuación se mencionarán algunas formas aplicadas en el nivel superior:

- Aprendizaje basado en problemas

Esta forma de aprendizaje busca primero plantear una problemática que permite identificar las necesidades del aprendizaje, así como la información necesaria para resolver el problema. La forma de trabajar es colaborativa y en pequeños grupos de alumnos, que comparten su experiencia, practican y desarrollan diversas habilidades. Este es un método que puede ser adoptado por una gran gama de disciplinas. (MONTERREY, 2000, pág. 3)

- Aprendizaje cognitivo

En el aprendizaje cognitivo los alumnos construyen su propio conocimiento y lo comparten con sus compañeros para enriquecerse mutuamente con experiencias propias. Las tecnologías de la información y comunicación sirven como herramientas para apoyar este estilo de aprendizaje, que permite diversas actividades como lo es el aprendizaje a distancia y el compartir materiales y objetos de aprendizaje.

Con el estilo de aprendizaje de forma cognitiva se da al alumno un rol activo en el proceso del aprendizaje, se logra el aprendizaje significativo y se desvía el aprendizaje memorístico. (UNESCO, 2004, pág. 33)

- Aprendizaje auto-regulado

Tiene un papel importante debido a que requiere de habilidades de uso cotidiano en el entorno académico, la forma de aprendizaje auto-regulada se ha utilizado recientemente para describir la generación y seguimiento de reglas que definen un comportamiento propio, a lo que conlleva la autorregulación del contexto académico. Se podría describir como la capacidad que un alumno posee para generar y regular su conocimiento propio construyendo su propio aprendizaje. (CASTRO, LANDA DURAN, & VEGA VALERO, pág. 5)

De acuerdo con (UNESCO, 2004, págs. 26-28), en el siglo XXI cambió el entorno de aprendizaje con la incorporación de las tecnologías de la información en la educación, en la Tabla 1 se muestran las diferencias entre un entorno que se centra en el profesor y otro que se centra en el alumno.

Tabla 1 - Diferencias entre entornos centrados en el profesor y centrados en el alumno. Tomada de (UNESCO, 2004, pág. 27)

	<i>Entorno de aprendizaje centrado en el docente</i>	<i>Entorno de aprendizaje centrado en el alumno</i>
<i>Actividades de clase</i>	<i>Centradas en el docente Didácticas</i>	<i>Centradas en el alumno. Interactivas.</i>
<i>Rol del profesor</i>	<i>Comunicador de hechos. Siempre experto.</i>	<i>Colaborador. A veces aprende de sus alumnos.</i>
<i>Énfasis instrucciones</i>	<i>Memorización de hechos.</i>	<i>Relacionar, cuestionar e inventar.</i>
<i>Concepto de conocimiento</i>	<i>Acumulación de hechos. Cantidad.</i>	<i>Transformación de hechos.</i>
<i>Demostración de aprendizaje efectivo</i>	<i>Seguir las normas como referencia.</i>	<i>Nivel de comprensión del alumno.</i>
<i>Evaluación</i>	<i>Múltiple opción.</i>	<i>Pruebas con criterio de referencia. Carpetas de trabajo y desempeño.</i>
<i>Uso de Tecnología</i>	<i>Repetición y práctica.</i>	<i>Comunicación, acceso, colaboración y expresión.</i>

Este tipo de aprendizaje en donde muestra al alumno más activo, implica cambio de roles entre profesores y alumnos, así proporcionar un papel más activo a los alumnos y a los profesores se les deja de ver como los transmisores de conocimiento y se convierten en orientadores y facilitadores del aprendizaje; con el uso de las nuevas tecnologías se ha apoyado este giro en donde los alumnos tienen la responsabilidad de su propio aprendizaje, para reforzar los conocimientos, habilidades que no se han asimilado o acomodado en el aula. En la Tabla 2 se muestran los cambios en los roles entre los profesores y los alumnos.

Tabla 2 - Cambios en el rol del docente. Tomada de (UNESCO, 2004, pág. 28)

<i>Cambios en el rol del docente</i>	
<i>Cambio de:</i>	<i>Cambio de:</i>
<i>Transmisor de conocimiento, fuente principal de información, experto en contenido y fuente de todas respuestas.</i>	<i>Facilitador del aprendizaje, colaborador, entrenador, tutor, guía y participante del proceso de aprendizaje.</i>
<i>El profesor controla y dirige todos los aspectos del aprendizaje</i>	<i>El profesor permite que el alumno sea más responsable de su propio aprendizaje y le ofrece diversas opciones.</i>
<i>Cambios en el rol del alumno</i>	
<i>Receptor pasivo de información</i>	<i>Participante activo del proceso de aprendizaje.</i>
<i>Reproductor de conocimiento</i>	<i>El alumno produce y comparte el conocimiento, a veces participando como experto.</i>
<i>El aprendizaje es concebido como una actividad individual.</i>	<i>El aprendizaje es una actividad colaborativa que se lleva a cabo con otros alumnos.</i>

Con base en lo anterior, se sabe que el desarrollo de la tecnología ha facilitado en gran medida la nueva forma de concebir el proceso de enseñanza-aprendizaje y ahora este es centrado en el alumno.

Al avanzar la tecnología, el área de la educación se ve obligada a incorporar métodos de enseñanza-aprendizaje, con materiales digitales que lo ayuden a comprender mejor algún tema.

Los alumnos por su parte, con el uso de las Tecnologías de la Información y la Comunicación (TICs) pueden recurrir a tutoriales y cursos en línea, a compartir conocimientos, obtener libros digitales, hacer uso de Internet para consultar información específica, objetos de aprendizaje, entre muchas otras actividades que ayudan a enriquecer su aprendizaje.

A continuación se describe de manera sucinta a los objetos de aprendizaje los cuales son parte de las TICs.

2.1.1 Tecnologías de la información y la comunicación

En esta sección se mencionan las TICs así como los diferentes tipos de ambientes de aprendizaje que existen, enfocándose principalmente en el nivel superior.

La tecnología tiene impacto sobre todas las cosas así como en el ambiente educativo, de acuerdo con (ROSARIO, 2005, pág. s/p) el uso de la tecnología en el aprendizaje plantea paradigmas nuevos que revolucionan la forma tradicional de enseñar y aprender.

Con el uso de Internet ahora se cuenta con mayor alcance del conocimiento y facilita el acceso a fuentes ricas en información como lo son museos, bibliotecas digitales, tutoriales en línea, textos, documentos y diversas páginas web.

Inclusive existe la modalidad de estudiar una carrera en línea. Se le llama así ya que se debe contar con una computadora personal y con el acceso a Internet. Cada vez son más las universidades que incorporan este método de enseñanza-aprendizaje en la actualidad.

Lo denominado tecnologías de la información y comunicación corresponde a: *“El conjunto de tecnologías que permiten la adquisición, producción, almacenamiento, tratamiento, comunicación, registro y presentación de informaciones, en forma de voz, imágenes y datos contenidos en señales de naturaleza acústica, óptica o electromagnética”*(ROSARIO, 2005, pág. s/p).

De acuerdo con (LÓPEZ GUZMÁN & GARCÍA PEÑALVO, 2004, pág. s/p) en nuestra sociedad la introducción de la tecnología obliga a las instituciones educativas establecer y renovar sus medios de enseñanza para poder formar profesionistas competentes en el mercado laboral. Es por esta razón que la educación superior se transforma y adopta nuevos esquemas que usa las tecnologías de la información y la comunicación.

2.1.1.1 Ambientes de aprendizaje

Un ambiente de aprendizaje es un concepto que involucra tanto a las personas y el aprendizaje. Está constituido por las dinámicas establecidas en los procesos educativos, materiales, actitudes y experiencias de las personas participantes del ambiente y la infraestructura necesaria.

Según (PINEDA, 2012, pág. s/p) el ambiente educativo se define como: *“la organización del espacio, la disposición y la distribución de los recursos didácticos, el manejo del tiempo y las interacciones que se dan en el aula”*

De acuerdo con (D. BRANSFORD, L. BROWN, & R. COCKING, 2007, págs. 12-33) existen varios tipos de ambientes de aprendizaje, los cuales se mencionan a continuación:

- *Ambientes centrados en el conocimiento:* Este tipo de ambiente se enfoca en la información y actividades que ayudan a los estudiantes a comprender las diferentes disciplinas.
- *Ambientes centrados en la evaluación:* En este ambiente se busca fortalecer el conocimiento proporcionando oportunidades de retroalimentación y revisión mediante la evaluación constante. Utiliza la evaluación de dos formas: 1) como recurso para medir de alguna manera lo que los estudiantes han aprendido y 2) como fuente de retroalimentación para mejorar el aprendizaje.
- *Ambientes centrados en la comunidad:* Se buscan soluciones de aprendizaje de forma grupal, por ejemplo, se pueden tomar las evaluaciones educativas como motivacionales para fomentar la competencia y superación del alumnado. Los individuos contribuyen a

la comunidad con sus fortalezas. Existen diferencias entre unas comunidades a otras y depende de la cultura y la región en donde sea aplicado el ambiente.

2.1.1.2 Presencial

Como se mencionó anteriormente en el aprendizaje centrado en el docente. El modelo presencial requiere de la asistencia a las clases, donde los actores son alumnos y profesores, y los profesores son los que facilitan los conocimientos y hacen uso de recursos educativos como un pizarrón, libros, libretas, lápices, etc.

Según (GALLARDO, 2012, pág. s/p) menciona que en el ambiente presencial es el profesor el que tiene el uso de la palabra y explica todo el tema a detalle y el alumno es simplemente el oyente y receptor de la información. Esta modalidad educativa es también llamada la “educación tradicional”.

- Ventajas :
 - Trato directo con el profesor
 - Fortalece el trabajo en equipo
 - Compañerismo
 - Incrementa la sociabilidad entre compañeros de clase
- Desventajas:
 - Se tiene un horario determinado
 - Requiere de la presencia física del alumno
 - Se tiene que cumplir con la asistencia constante a clases

2.1.1.3 Ambiente de aprendizaje Virtual

Un ambiente virtual, según Ávila et. al., es el espacio en donde las tecnologías se integran al entorno escolar favoreciendo el conocimiento y a los procesos pedagógicos. Se llama virtual debido a que el espacio de estudio no se lleva a cabo en un lugar predeterminado y los recursos se mantienen de forma digitalizada. (ÁVILA & BOSCO, 2001, págs. 1-2)

El ambiente virtual está conformado por:

- El estudiante
- El asesor
- Los contenidos educativos
- El espacio
- La evaluación
- Los medios de información

En este entorno de aprendizaje la tecnología juega un papel muy importante, ya que es necesaria la incorporación de los dispositivos tecnológicos para poder aplicarse. Debido a que todos los recursos y materiales educativos se encuentran de forma virtual. Se enfoca principalmente en población que tiene dificultades temporales para poder asistir a clases de la forma tradicional.

De acuerdo con Rosario las ventajas y desventajas que presenta este ambiente de aprendizaje son:

- Ventajas :
 - El horario de estudio se puede adaptar a un nivel personal
 - El alumno juega un papel activo
 - Mayor ahorro de dinero y tiempo
 - Descansa sobre el alumno la responsabilidad del proceso formativo
 - Brinda la oportunidad de aprender a todas las personas que no tienen la posibilidad de acceder a clases presenciales
- Desventajas :
 - Las clases se interrumpirán al haber algún tipo de falla técnica
 - Existe diferencia en cuanto al contacto de persona a persona
 - Se requiere de responsabilidad y disciplina por parte del estudiante
 - Requiere de capacitación y actualización por parte del personal docente para combinar la tecnología y la enseñanza

Un ejemplo de este tipo de ambiente de aprendizaje es la enciclopedia virtual llamada Wikipedia, esta enciclopedia puede ser editada por cualquier persona que tenga los conocimientos y desee compartirlos por este medio. En este tipo de ambiente de aprendizaje se sugiere incorporar los objetos de aprendizaje móviles. (ROSARIO, 2005, pág. S/P)

2.1.1.4 Blendedlearning

El concepto de Blendedlearning es el siguiente: “*Modo de aprender que combina la enseñanza presencial con la tecnología no presencial*” (BARTOLOMÉ, 2004, pág. 5)

Es también denominado como un modelo virtual-presencial de enseñanza, en donde las tecnologías presenciales y virtuales se unen para optimizar el aprendizaje y el profesor tiene el rol tradicional pero se apoya en el uso de las tecnologías. Por ejemplo: el profesor imparte una clase normal y también tutorías a distancia, la forma en que se lleve cabo la enseñanza-aprendizaje dependerá del curso en cuestión.

Se plantea a *Blendedlearning* como el buen uso de las tecnologías para la educación, supera a *E-learning* modalidad que solo abarcaba el uso de Internet para la adquisición del conocimiento y habilidades, su fracaso se debió a diversos factores como que el alumno no contara con ese contacto humano, y también que las personas no tuvieran la iniciativa propia para estudiar ni capacidad de organización aunado a la falta de constancia en el estudio (MARTÍNEZ, 2002, págs. 2-3).

- Ventajas :
 - Proporciona mejoras en la calidad del aprendizaje
 - El papel del estudiante es más activo
 - Mayor beneficio al disponer de más material y recursos educativos
 - Aprovecha el material existente en la red de Internet
 - El alumno puede enriquecer su aprendizaje por cuenta propia al utilizar diversos recursos tecnológicos

- Desventajas :
 - Su efectividad depende de los materiales y recursos que se tengan
 - Los alumnos deben contar con el conocimiento suficiente para poder utilizar los recursos tecnológicos
 - Se debe buscar información de fuentes confiables al hacer uso de la red de Internet (BARTOLOMÉ, 2004, págs. 7-9)

2.2 Los objetos de aprendizaje

De acuerdo con (ALVARES, 2004, pág. 11) que cita al Comité de Estándares de Tecnologías de Aprendizaje (del inglés: *Learning Technologies Standard Commite*), donde un objeto de aprendizaje se define como:

“Una entidad digital o no digital, que puede ser usada, re-usada o referenciada durante el aprendizaje apoyado por la tecnología. Ejemplos de aprendizaje apoyado por tecnología incluyen a los sistemas a entrenamiento basado en computador, ambientes de aprendizaje interactivos, sistemas inteligentes de instrucción asistida por computador, sistemas de aprendizaje a distancia y ambientes colaborativos de aprendizaje. Ejemplos de objetos de aprendizaje, incluyen contenidos multimediales, contenidos instruccionales, objetivos de aprendizaje, Software instruccional y herramientas de Software, y personas, y organizaciones y eventos referenciados durante el aprendizaje apoyado por tecnología”

La ventaja de los objetos de aprendizaje es que pueden ser reutilizados en diversas asignaturas, ayudar a los alumnos y profesores a tener un nivel grupal de conocimientos.(RODRÍGUEZ, 2009, pág. 1)

Los objetos de aprendizaje deben ser escalables y reusables, es decir, la capacidad que tienen los objetos de aprendizaje para agregarse a estructuras más complejas y que se puedan utilizar nuevamente en otros cursos respectivamente. Además, deben cumplir con la autocontención conceptual que hace referencia a la capacidad que tienen para auto explicarse y facilitar el aprendizaje.(GONZÁLEZ, 2009, pág. 26)

2.2.1 Tipos de Objetos de Aprendizaje

Todos los objetos de aprendizaje tienen ciertas cualidades, es la diferencia en la forma en la cual los objetos exhiben tales cualidades que los hacen ser un objeto de aprendizaje distinto de otro. La siguiente taxonomía diferencia entre cinco tipos de objetos de aprendizaje. Los ejemplos de estos cinco tipos de objetos se mencionan enseguida seguidos por la taxonomía, la cual explica sus diferencias y similitudes. La distinción entre los tipos de objetos de aprendizaje es cosa de identificar la forma en que el objeto a clasificar presenta ciertas características. Estas características son los atributos críticos y son estables aún en ambientes separados, por ejemplo, las propiedades permanecen igual sin importar si un objeto reside en una librería digital. El propósito de la taxonomía es diferenciar los posibles tipos de objetos de aprendizaje (WILEY, 2000).

2.2.1.1 Fundamental.

Un recurso digital individual sin combinar con otro, el objeto de aprendizaje fundamental es, generalmente, una ayuda visual que presenta una función. Por ejemplo, una imagen JPEG de una mano tocando un acorde o un teclado de piano.

2.2.1.2 Combinado-Cerrado.

Un pequeño número de recursos digitales combinados en el momento de diseño del objeto, los recursos que constituyen el objeto no están accesibles para su reutilización en forma individual. Un video clip ejemplifica esta definición, como las imágenes fijas y el audio están combinados en una forma en la cual las piezas constituyentes del objeto no son recuperables (o al menos son difíciles de recuperar). Los objetos de aprendizaje combinados cerrados pueden contener lógica limitada (por ejemplo, la habilidad para revisión de hojas de respuestas) pero no deben contener lógica interna compleja (por ejemplo, la capacidad para evaluar un conjunto de casos) dada esta capacidad el objeto no sería reutilizable en otros objetos de aprendizaje. Los objetos de aprendizaje combinados cerrados son generalmente de un solo propósito, esto es, proporcionan instrucción o práctica. Por ejemplo, el video de una mano tocando un acorde arpegiado en un teclado de piano, acompañado por audio.

2.2.1.3 Combinado-Abierto.

Un gran número de recursos digitales combinados por una computadora en tiempo real cuando se hace una petición al objeto, cuyos objetos constituyentes son directamente accesibles para su reutilización. Una página Web ejemplifica esta definición, cuando sus componentes imágenes, video clips, etc., existen en un formato reutilizable y son combinados en un objeto de aprendizaje en tiempo de ejecución. Los objetos combinado abierto frecuentemente combinan instrucción y práctica relacionados, con el fin de crear una unidad instruccional completa. Por ejemplo, una página Web combinando dinámicamente la imagen JPEG anteriormente mencionada y un archivo Quicktime con texto relacionado.

2.2.1.4 Generativo-Presentación.

Es un objeto que tiene la lógica y estructura para combinar o generar y combinar objetos de aprendizaje de bajo nivel (Tipos Fundamental y combinado cerrado). Los objetos de aprendizaje generativo presentación pueden ser por ejemplo, un applet de Java capaz de generar un conjunto notas musicales, posicionándolas apropiadamente para presentar un problema de identificación de un acorde al estudiante.

2.2.1.5 Generativo instruccional.

Son objetos que instruyen y proporcionan ejercicios de práctica para cualquier tipo de procedimiento, por ejemplo, el proceso de identificar la raíz, calidad e inversión de un acorde.

2.2.2 Recursos educativos: Objetos de aprendizaje

Los recursos educativos son aquellos materiales con los que se cuenta para fortalecer el aprendizaje. Por ejemplo: Una presentación con diapositivas, objetos de aprendizaje, entre otros y pueden ser incorporados en diversos ambientes de aprendizaje.

En los medios digitales se habla de “objetos de aprendizaje”, según (LÓPEZ GUZMÁN & GARCÍA PEÑALVO, 2004, pág. 5) mencionan que: *“La IEEE (2001) dice que los objetos de*

aprendizaje son una entidad, digital o no digital, que puede ser utilizada, reutilizada y referenciada durante el aprendizaje apoyado con tecnología”

A continuación se mencionan algunos recursos educativos utilizados comúnmente:

- Tutoriales y video tutoriales: Se trata de un curso generalmente breve que enseña los conocimientos necesarios de un tema en específico. Puede ser también en la modalidad de video, Ejemplos: tizaymouse.com, youtube.com, videotutoriales.com, tutoriales.com, entre otros.
- Blogs: Es un portal de Internet en donde uno o más autores escriben información relevante de algún tema en específico, y son actualizados constantemente. Posee la característica que los lectores pueden dejar comentarios y ser contestados por el autor.
- Foros: Son páginas web en donde se discuten diversas temáticas, los usuarios participantes puede interactuar y expresar sus opiniones y dudas respecto a un tema en especial.
- Plataformas virtuales: Son Sistemas de gestión de contenidos de aprendizaje y además proporcionan ayuda para crear comunidades, por ejemplo: Moodle, ATutor, entre otros.
- Diccionarios y enciclopedias virtuales: Son sitios web destinados a dar definiciones o explicaciones específicas de algún concepto o tema, puede ser escrito por una institución formal como tal es el caso del diccionario de la real academia española o puede ser escrito por los usuarios adscritos al portal como por ejemplo Wikipedia.

2.3 Ingeniería del Software

2.3.1 Definición de la Ingeniería de Software

Antes de definir el término de Ingeniería de Software, primero se describirán las terminologías correspondientes a lo que es Ingeniería y Software por separado.

Ingeniería:

Definición 1: “*f. Arte de aplicar los conocimientos científicos a la invención, perfeccionamiento o utilización de la técnica industrial en todas sus determinaciones.*” (QUILLET D. E., 1985, pág. 178)

Definición 2: “*f. Conjunto de conocimientos y técnicas que permiten aplicar el saber científico a la utilización de la materia y de las fuentes de energía, mediante invenciones o construcciones útiles para el hombre.*”(ESPAÑOLA D. D., 1984, pág. 772)

Software:

Definición 1: “*Instrucciones para una computadora. Una serie de instrucciones que realizan una tarea en articular se llama programa o programa de Software. Las dos categorías principales son software de sistemas y de aplicaciones. Según (FREEDMAN, 1993, pág. 717) el Software de sistemas se compone de programas de control, Software de comunicaciones y administrador de base de datos. El Software de aplicaciones es cualquier programa que procesa datos para el usuario*”.

Definición 2: “*m. Inform. Conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora.*” (ESPAÑOLA R. A., 2012, pág. s/p) Una vez descritas las definiciones ahora se procederá a definir por completo el término compuesto de Ingeniería de Software.

Con esto tenemos la siguiente definición “*Ingeniería del Software es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del Software*” según (IEEE, 1993).

El Software se forma con las instrucciones o programas que al ejecutarse brindan las características, funciones y el nivel de desempeño deseados; las estructuras de datos permiten a los programas manipular información de manera adecuada; y por último los documentos que especifican y describen el uso y la operación de dichos programas.

El Software es un elemento lógico, no físico y posee las siguientes características:

1. El Software se desarrolla o construye.
2. El Software no sufre desgaste como el hardware.
3. La mayor parte del Software aún se construye a la medida y no por componentes.

Un componente de Software es un código que se diseña e implementa de forma que pueda utilizarse en diversos programas diferentes.(PRESSMAN, 2006, págs. 5-8).

La Ingeniería del Software es la disciplina que tiene por objetivo la creación de Software para ser entregado a tiempo sin que el costo rebase el presupuesto, además que el Software se encuentre exento de fallas y a su vez satisfaga las necesidades de los clientes, además de incluir que éste pueda modificarse conforme a las nuevas necesidades de los usuarios (SCHACH, 2006, pág. 4).

Según (BRAUDE, 2003, págs. 2-3), señala que dicha disciplina involucra a personas, procesos, proyectos y productos; motivo por el cual se le llaman las 4 “P” de la Ingeniería del Software (ver Figura 1).

Personas	• Quién lo hace
Procesos	• La manera en que se hace
Proyectos	• La realización
Productos	• La aplicación de artefactos

Figura 1 - 4'p de la Ingeniería de Software (BRAUDE, 2003, pág. 3)

A continuación se describen las 4'p de la Ingeniería del Software:

- **Personas** quienes llevan a cabo la realización de un proyecto.
- **Procesos** la forma en que se hacen los proyectos.
- **Proyectos** se muestran los roles y actividades de cada uno de los ingenieros que intervienen en la realización del trabajo.
- **Productos** en donde además de consistir en el código fuente y código objeto también incluye documentación, los resultados de las pruebas así como también las medidas de productividad.

La Ingeniería del Software involucra herramientas, métodos, un proceso y el enfoque de calidad (Figura 2).



Figura 2 - Estratos de la Ingeniería del Software (PRESSMAN, 2006, pág.24)

La base de la Ingeniería del Software es el **enfoque de calidad**, la gestión de la calidad fomenta la cultura de mejora continua del proceso.

El **proceso** es el elemento que mantiene juntos los estratos de la tecnología y permite el desarrollo a tiempo del Software.

Los **métodos** proporcionan la forma técnica para construir el Software. Abarca tareas como comunicación, análisis de requisitos, el modelado del diseño, la construcción del programa, la realización de pruebas y el soporte.

Las **herramientas** proporcionan el soporte automatizado y semi-automatizado para el proceso y los métodos.

2.3.2 Objetivos de la Ingeniería del Software

Una vez revisada la definición de Ingeniería del Software a continuación se abundará sobre los objetivos que tiene esta disciplina.

En 1968 durante la conferencia de Ingeniería de Software de la OTAN (Organización del Tratado del Atlántico Norte) que se llevó a cabo en Garmisch, Alemania; se apoyó y discutió que el hecho de realizar Software es realmente un trabajo meramente ingenieril ya que la Ingeniería del Software debe usar los métodos y paradigmas que las disciplinas de Ingeniería establecen para solucionar problemas. (SCHACH, INGENIERÍA DE SOFTWARE CLÁSICA Y ORIENTADA A OBJETOS, 2006, pág. 5)

En el caso de la Ingeniería del Software el problema al que se enfrenta frecuentemente es a lo que se denomina la *crisis del Software*.

Uno de los objetivos de la Ingeniería del Software es superar esta crisis del Software que se refiere a cuando un producto no es de calidad ni cumple con ser entregado a tiempo, no cumple las necesidades del cliente y además excedió los presupuestos previstos para su realización.

De acuerdo con (SCHACH, 2006, pág. 6) a continuación se presentan algunas consecuencias financieras a causa de la crisis del Software según la encuesta realizada por el CutterConsortium [2002].

- Un 78% de las organizaciones tecnológicas se vieron involucradas en disputas que finalizaron en litigio.
- En un 67% de los casos la funcionalidad o el desempeño de los productos entregados no cumplió con los términos ofrecidos por parte de los programadores.
- Un 56% de los casos la fecha de entrega fue aplazada varias veces.
- En un 45% de los casos existían fallas muy graves en los productos que volvieron el Software inutilizable.

“Los buenos ingenieros de Software evitan repetir los errores de proyectos anteriores documentando y manejando sus procesos de desarrollo” (BRAUDE, 2003, pág. 16).

Según (BRAUDE, 2003, pág. 18) todos los proyectos requieren documentarse desde el inicio hasta el final y pueden haber bastantes cambios en los documentos durante el desarrollo del Software. Es por ello que se debe llevar un control de los cambios realizados tanto en los documentos como en el código. A este control se le denomina ***administración de la configuración***.

La administración de la configuración se encarga de administrar por una parte la acumulación de partes nuevas en el proyecto y también de las versiones sucesivas de dichas partes. Se debe ser cuidadoso en cuanto a no perder la orientación en las versiones, saber exactamente donde se encuentran todas y cada una de las partes del proyecto y cuáles son las que deben ir juntas. (BRAUDE, 2003, pág. 51)

2.3.3 Ciclos de vida

Existen *modelos prescriptivos* y *modelos incrementales*, en donde los modelos prescriptivos indican el conjunto de elementos del proceso como son: las actividades del marco de trabajo, acciones de Ingeniería del Software, productos del trabajo, aseguramiento de la calidad y mecanismos de control de cambio para cada proyecto. Así como también se prescribe un flujo de trabajo el cual indica la forma en que se irán desarrollando los elementos del proyecto en sí.

En cuanto a los modelos incrementales producen Software como una serie de entregas de incrementos de Software, aplican secuencias lineales de forma escalonada conforme avanza el tiempo del proyecto. (PRESSMAN, 2006, págs. 52,73)

- *Modelo del ciclo de vida en cascada.*

Es también llamado el *ciclo de vida clásico*, este paradigma es el más antiguo y tiene un enfoque sistemático y secuencial para desarrollar Software. Es muy útil cuando hay situaciones en donde los requerimientos son fijos y el trabajo se realiza hasta su conclusión de una forma lineal. (PRESSMAN, 2006, págs. 50- 51).

Según (BRAUDE, 2003, págs. 25-26), el modelo en cascada se compone de las siguientes etapas (ver Figura 3):

1. **Análisis de requerimientos:** En donde se agrupan las necesidades del producto.
2. **Diseño:** Se describe la estructura interna del producto, se acostumbra representar con diagramas y texto.
3. **Implementación:** En esta fase se procede la programación del código, representa el código en sí.
4. **Integración:** Se refiere al proceso en el cual se reúnen todas las partes que conformaran el proyecto y así poder completar el producto.
5. **Pruebas:** Donde se incluyen los reportes de las pruebas y la descripción de sus defectos.

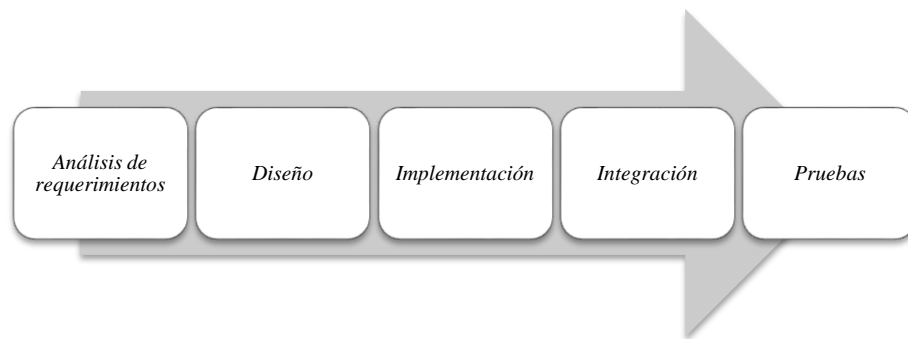


Figura 3 - Modelo en Cascada (BRAUDE, 2003, pág.25)

Existe una versión más completa del modelo de proceso en cascada que incluye las siguientes etapas (ver Figura 4):

- *Análisis conceptual:* en donde se define la idea global del proyecto
- *Análisis orientado a objetos:* se establecen las clases importantes
- *Pruebas unitarias y del sistema:* Se prueban cada una de las partes que integran al proyecto y el proyecto como un total

- *Mantenimiento*: Se realiza al final del proceso y consiste en modificar y reparar la aplicación para que siempre sea útil

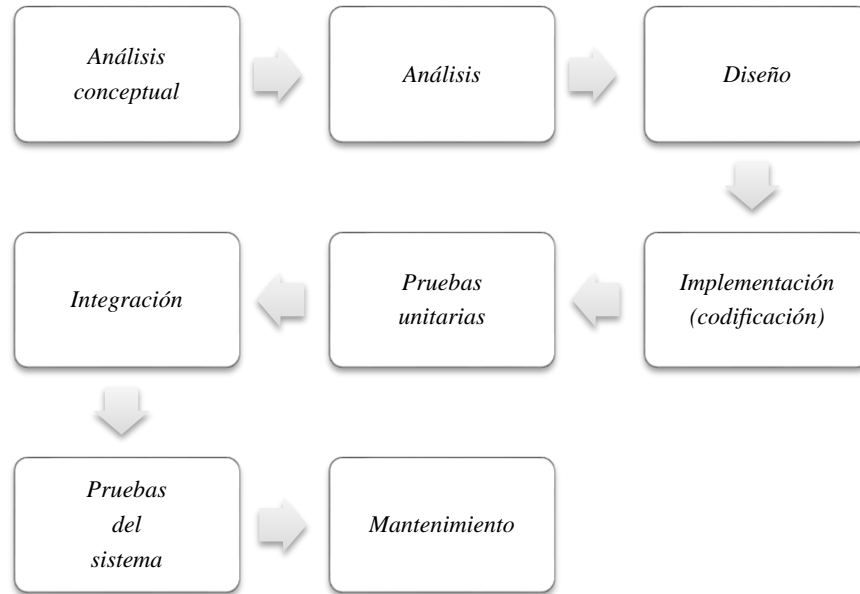


Figura 4 - Modelo en cascada con más detalle.(BRAUDE, 2003, pág. 26)

Las desventajas que presenta el modelo en cascada son:

- Es muy poco común que los proyectos reales sigan un flujo secuencial
- Frecuentemente se dificulta establecer los requisitos del sistema de manera explícita al cliente, ya que el modelo en cascada lo requiere
- El proyecto tendrá una versión funcional para cuando esté muy avanzado el desarrollo del mismo, por esta razón un error será catastrófico sino es detectado pertinentemente

También la naturaleza lineal del modelo genera “estados de bloqueo” en donde cada miembro del proyecto debe esperar a otro para poder realizar su trabajo ya que las tareas son dependientes y el tiempo de espera puede superar al tiempo productivo.(PRESSMAN, 2006, pág. 50)

- *Modelo del ciclo de vida en espiral*

Este modelo se compone de cuatro etapas (ver Figura 5) las cuales se repiten constantemente, las etapas son las siguientes: *requerimientos*, *diseño*, *implementación* y *pruebas*.

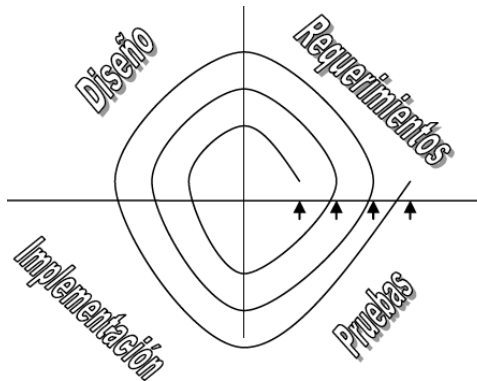


Figura 5 - Modelo del ciclo de vida de espiral (WEITZENFELD, 2005, Pág.53)

Las razones por las que se renuevan las etapas constantemente son principalmente porque se tiene la necesidad de eliminar riesgos, se desea construir una versión parcial preliminar del producto la cual podrá ser mostrada al cliente de una forma inicial y obtener una retroalimentación y finalmente se evita la integración masiva de código.

Al utilizar el modelo en espiral cabe destacar que requiere de una administración mucho más cuidadosa del que se requiere en el modelo en cascada, la documentación debe ser consistente siempre que se cumpla una iteración completa. Surgen ocasiones en las que se necesita iniciar una iteración antes de terminar la anterior con el mero propósito de optimizar la producción.(PRESSMAN, 2006, pág. 53)

El modelo en espiral presenta algunas creencias:

- Cuando se entienden los objetivos y los riesgos que se involucran en el proyecto es cuando comienza una actividad.
- Se utilizan aquellas herramientas que sirvan para reducir los riesgos basándose en la evaluación de las diferentes soluciones alternas.

- Todas las personas involucradas en el proyecto deben revisar cada actividad comprometiéndose con las siguientes actividades.
- Al incrementarse en cada etapa el desarrollo del Software se obtienen prototipos consecutivos del producto.(WEITZENFELD, 2005, pág. 53)

- *Modelo de ciclo de vida de codificación y ajuste*

En este tipo de modelo los programadores lanzan los códigos y los reajustan de acuerdo a las necesidades del cliente tantas veces como sea necesario (ver Figura 6); carece de requerimientos, especificaciones y diseño, puede funcionar bien para proyectos que son pequeños pero no lo es así para proyectos robustos.

Esta es la forma más sencilla de construir Software y por lo tanto la peor, ya que es difícil mantener un producto sin documentos de especificaciones o de diseño. (SCHACH, 2006, pág. 48).

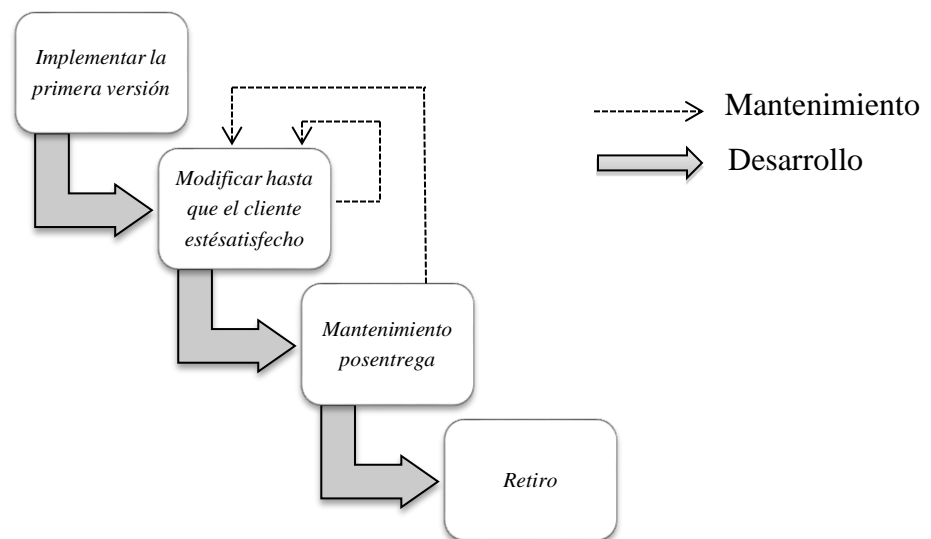


Figura 6 - Modelo del ciclo de vida de codificación y ajuste.(SCHACH, 2006, pág.48)

- *Modelo de elaboración rápida de un prototipo*

“Un prototipo rápido es un modelo de trabajo cuya función es equivalente a un subconjunto del producto”

En este ciclo de vida se construye un prototipo rápido y se permite que el cliente y usuarios interactúen con él. Cuando el cliente está convencido de lo que ofrece el prototipo es eso lo que necesita los programadores proceden a redactar la documentación correspondiente al proyecto.

Una de las principales fortalezas de este modelo es que la creación del producto es lineal, inicia desde el prototipo rápido y termina hasta que se entrega el producto. (SCHACH, 2006, pág. 51). El ciclo de vida en modelo de elaboración rápida de un prototipo se muestra en la Figura 7.

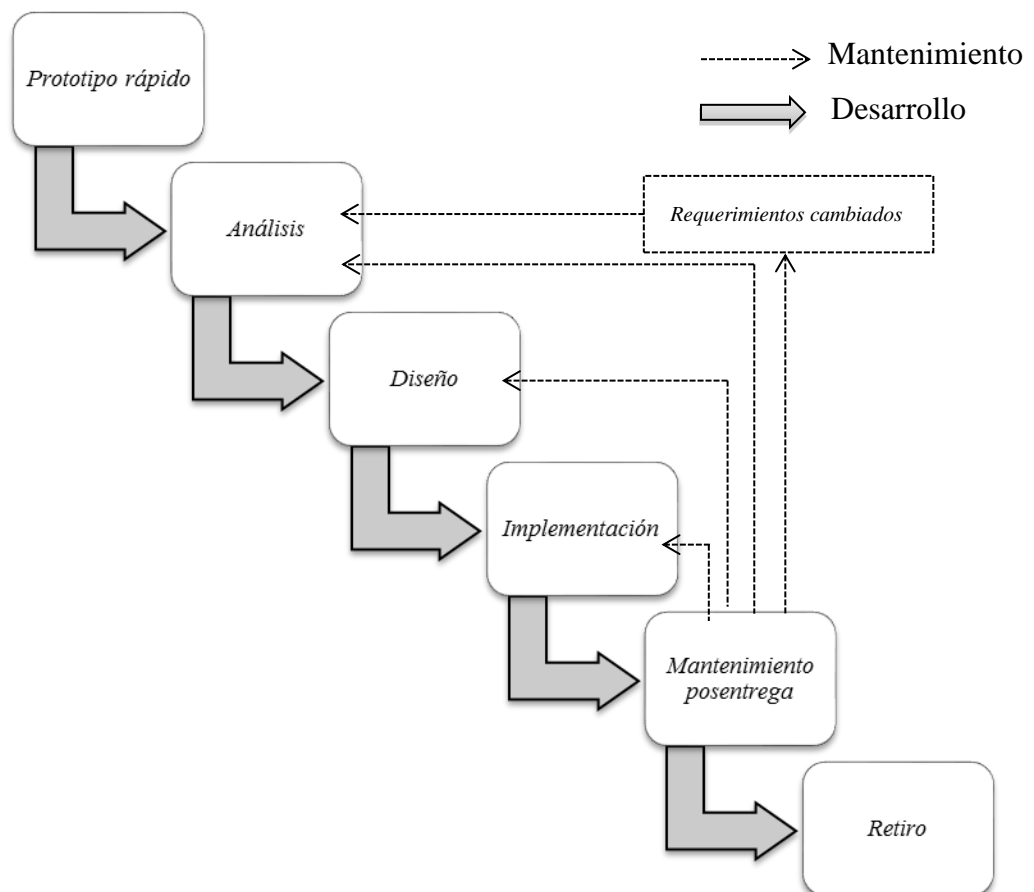


Figura 7 - Modelo del ciclo de vida de elaboración rápida de un prototipo.(SCHACH, 2006, pág. 51)

- *Modelo de ciclo de vida de sincronización y estabilización*

El modelo de ciclo de vida de sincronización y estabilización es utilizada por la empresa Microsoft Inc. para desarrollar la mayoría de sus paquetes, este ciclo de vida es una versión más del modelo iterativo e incremental. En su fase de análisis de requerimientos se hacen una serie de entrevistas a las personas que podrían ser clientes potenciales para comercializar el paquete, como resultado de esas entrevistas se elaboran listas con las características de más prioridad para los clientes. Después se llena un documento de especificaciones.

Se divide el trabajo en estructuras, la primera contendrá las características más importantes y así sucesivamente de mayor a menor importancia. Se forman equipos pequeños y trabajan en forma paralela para desarrollar las tareas asignadas. Al final de una jornada se reúnen los equipos y eliminan errores de los productos resultantes.

La *estabilización* se realiza al final de cada estructura, y es en esta fase donde se arreglan las deficiencias detectadas y al final ya no se realizan más cambios a las especificaciones.

En la fase de sincronización se asegura que los componentes trabajen de manera conjunta.(SCHACH, 2006, pág. 54)

En la Tabla 3 se muestran las principales fortalezas y debilidades de cada uno de los modelos mencionados anteriormente.

Tabla 3 - Comparación entre modelos de ciclo de vida (SCHACH, 2006, pág. 59)

<i>Modelo de ciclo de vida</i>	<i>Fortalezas</i>	<i>Debilidades</i>
<i>De codificación y ajuste</i>	<i>Es bueno para programas cortos que no requieren mantenimiento.</i>	<i>Totalmente insatisfactorio para programas complejos.</i>
<i>De cascada</i>	<i>Enfoque con disciplina manejado con documentos.</i>	<i>El producto resultante puede no satisfacer las necesidades de los clientes.</i>
<i>De elaboración rápida de prototipo</i>	<i>Se asegura de que el producto resultante satisfaga las necesidades de los clientes.</i>	<i>Todavía no hay ninguna debilidad comprobada</i>
<i>De sincronización y estabilización</i>	<i>Cumple con las necesidades futuras del cliente. Se asegura de que se puedan integrar bien otros componentes.</i>	<i>No se ha usado ampliamente, sólo en Microsoft.</i>
<i>En espiral</i>	<i>Orientado a riesgos.</i>	<i>Solo se puede usar para productos locales a gran escala. Los desarrolladores deben ser competentes en el análisis de los riesgos y su solución.</i>

2.4 Dispositivos y Aplicaciones móviles

Los dispositivos móviles son aparatos electrónicos que brindan los beneficios a los usuarios como: mantenerlos comunicados, organizar sus agendas, tomar fotografías, hablar por teléfono, leer documentos, visitar páginas de Internet, escribir correos electrónicos, mandar mensajes de texto, guardar y procesar gran cantidad de información, utilizar redes sociales, descargar programas e instalarlos, funcionar como módem inalámbrico, manejar una amplia gama de conectividad mediante las tecnologías: Bluetooth, WiFi, 3G , y otras funciones más. No todos los dispositivos móviles presentan estas funcionalidades algunos pueden ser más poderosos que otros y presentar características tanto hardware y software mucho más sofisticadas.

También existen algunos dispositivos que fueron especialmente diseñados para realizar solamente algunas funciones, tal es el caso de los reproductores de música portátil. Por otra parte, existe una gran diversidad de aplicaciones móviles de toda índole, en el presente trabajo de tesis se mencionaran de forma breve algunas aplicaciones para Programación que ayuden a los estudiantes de Ingeniería en sus tareas y actividades cotidianas.

2.4.1 Tipos de dispositivos móviles

A continuación se presentara una breve recopilación de algunos dispositivos móviles que son los de mayor uso común; los cuales están clasificados por tipo de dispositivo móvil y por sus características generales.

- Tipos de teléfonos celulares
 - Celular sencillo: Es un tipo de dispositivo que solo permite la comunicación mediante llamadas y mensajes de texto, puede contar con aplicaciones no tan complejas como agenda telefónica, calculadora, reloj, radio FM, pantalla a color o monocromática, etc.
 - Celular sofisticado: Posee las características de un dispositivo sencillo además de tener mayor capacidad de memoria interna y extraíble, conectividad mediante tecnología Bluetooth, Wi-Fi, reproductor de música y videos, altavoz, pantallas a color y touchscreen, cámara fotográfica, etc.
 - Smartphone o teléfono inteligente (ver Figura 8): Además de contar con las características que posee un celular sofisticado, un Smartphone posee un sistema operativo que permite instalar y ejecutar diversas aplicaciones y juegos, algunos de estos son gratuitos. El sistema operativo depende del equipo y de la marca del dispositivo móvil, los sistemas más utilizados en la actualidad son: Android, BlackBerry OS, Windows Phone, iOS, Symbian, entre otros. También cuenta con sensores tales como giroscopio, acelerómetro, sensor de luz, sensor de proximidad, de ubicación GPS, etc.



Figura 8 – Smartphone o teléfono inteligente (Disponible en http://www.clipartpanda.com/clipart_images/you-can-download-the-clip-art-1104135)

- Tabletas electrónicas: La tableta electrónica es otro dispositivo móvil que realiza principalmente las siguientes funciones: navegar por internet, envío de mensajes de texto, consulta de libros electrónicos, crear y consultar documentos de texto, recibir llamadas electrónicas, reproducción de videos y música cámara de video y fotografías juegos videoconferencia, etc.
- Prospectiva de dispositivos móviles para el año 2020. En el campo de la Ingeniería del Software es importante saber qué depara el futuro, ya que se tiene que estar preparados para las nuevas demandas del mercado productivo, laboral y escolar. Por ello en esta sección se analizarán los factores que intervienen en el uso de nuevas tecnologías y el impacto que ha tenido de una forma global tanto en países desarrollados como en los que están en vías de desarrollo.

Según (REYES & MORALES VELÁZQUEZ, s/a, págs. 2-4) señalan que las tecnologías de la información y comunicación han crecido exponencialmente y los sectores que mayormente se han impulsado son los del rubro móvil y los servicios de datos móviles.

En los países en desarrollo son cada vez más las personas que requieren de servicios móviles así como también en países subdesarrollados, para que se haya logrado este nuevo enfoque tuvieron que intervenir los siguientes factores:

- Incremento de usuarios de Internet.
- Incremento en el uso de dispositivos móviles debido a su abaratamiento y gran diversidad.
- Nueva generación de usuarios para el 2020, llamada la generación Z.
- El incremento del acceso a Internet por parte de la generación digital, prevé una saturación de los servicios, que incluye así una mayor demanda en cuanto a calidad.

2.5 Edición e implementación de objetos de aprendizaje móviles

2.5.1 Software con licencia

Al referirse al término de Software con licencia, se entiende que se necesita tener un permiso para su uso.

El uso del Software tiene ciertas restricciones ya que no puede ser copiado, compartido, modificado, redistribuido o haberle aplicado Ingeniería inversa por ninguna persona que sea desarrolladora de Software o usuario, a no ser que en la propia licencia se especifique.

De la misma forma, la licencia aplica a un solo equipo de cómputo ya que se debe adquirir una nueva licencia para cada equipo diferente. (STORY, 2004, pág. 6)

A continuación se mostraran algunos términos que pueden ser empleados al momento de definir un Software como Software con licencia.

- Propiedad intelectual

Durante los años ochenta se preguntaron cómo debía protegerse el Software por lo que la solución es basarse en normas de los derechos de autor que rigen las obras literarias y musicales. (SGUERRA, 2009, pág. 21)

- Los derechos de autor

Los autores tienen una ley de propiedad intelectual, se desarrolló sobre la base del convenio de Berna en 1886 que en un principio era para proteger los trabajos literarios, separa los derechos en dos; por una parte están los derechos morales los cuales garantizan al autor controlar la divulgación de la obra bajo un nombre o un pseudónimo contando con el derecho de modificación y que sea de por vida; y por otra los derechos patrimoniales que hace referencia al derecho de explotar económicamente el producto creado, con una vigencia de 70 años aún después de muerto el autor.(SGUERRA, 2009, pág. 21)

En México los derechos de autor son protegidos por la Ley Federal del Derecho de Autor, en la cual se establece que autor es la persona física que ha creado una obra, específicamente en el ámbito computacional hace referencia a programas de cómputo. De acuerdo con (UNIÓN, CAMARA DE DIPUTADOS DEL H. CONGRESO DE LA, 2012, págs. 17-19) En la Ley Federal del Derecho de Autor, Capítulo IV, De los Programas de Computación y las Bases de Datos, Artículo 101:

“Se entiende por programa de computación la expresión original en cualquier forma, lenguaje o código, de un conjunto de instrucciones que, con una secuencia, estructura y organización determinada, tiene como propósito que una computadora o dispositivo realice una tarea o función específica.”

- Licencia

Es una manera de cesión de derechos, sin exclusividad y se supone aceptado por el receptor desde el momento en el que adquiere el bien, para el caso del Software no se permite la copia, excepto las copias de seguridad del mismo usuario y no para uso de terceros.(SGUERRA, 2009, pág. 21)

De acuerdo con (LABRADOR, 2005, pág. s/p), se llama licencia a el contrato efectuado entre el desarrollador y el usuario, y se establecen los derechos y las obligaciones.

- El secreto comercial

Hace intento de ocultar la información que no se quiere que salga a la luz, esta es muy utilizada por los laboratorios en la fabricación de productos químicos o en la industria farmacéutica; en el Software se proponen a no entregar los programas fuentes sino los programas ya compilados, es decir, en binario.(SGUERRA, 2009, pág. 22)

- Las patentes

Es referida principalmente a los inventos, los cuales salen a uso del público lo que facilita su reproducción pero también limitándola, puede ser que el dueño permita o no su reproducción; si es permitida se pagara un precio estipulado por el propietario de la patente.

En la industria del Software las empresas multinacionales de Software buscan la aprobación de una Ley de Patentes para el Software, es un tema que está en discusión en la actualidad. Por su parte Richard Stallman trata de evitar que una ley de este tipo sea aprobada, su peligrosidad radica en que el programador puede crear algoritmos nuevos o creer que los ha inventado sin embargo ya pueden existir patentes que prohíben su uso y esto lo convertiría en casi imposible de saber, es decir, el problema radica en que si un programa tiene algoritmos ya patentados o que alguien patente un algoritmo simple y por lo tanto perjudique el desempeño y la creatividad de otros programadores. Stallman lo compara con la posibilidad de patentar un acorde musical, que frenaría la composición musical.(SGUERRA, 2009, pág. 2)

2.5.2 Software libre

El término Software libre según Richard Stallman se refiere a que el usuario posee libertad para:

- La ejecución del programa en cualquier sitio, con el propósito que sea y por el tiempo que sea.
- Estudio y adaptación de acuerdo a las necesidades propias, por lo tanto esto exige acceso al código fuente
- Redistribuirlo a modo que se colabore con otras personas.
- Mejora del código y su libre publicación

Para lograr estas características del Software libre, Stallman creó una licencia llamada *General Public Licence(GPL)*, con la cual el autor de un Software puede liberar su producto bajo las características anteriormente explicadas y así propiciar que su Software este en mejora continua.

En (FREE SOFTWARE FOUNDATION, 2012) menciona las siguientes libertades para los usuarios de Software libre:

- “La libertad de usar el programa, con cualquier propósito (libertad 0)”
- “La libertad de estudiar el funcionamiento del programa, y adaptarlo a sus necesidades (libertad 1). El acceso al código fuente es un prerequisite para esto”
- “La libertad de distribuir copias para ayudar a los demás (libertad 2)”

- *“La libertad de mejorar el programa y de publicar las mejoras, de modo que toda la comunidad se beneficie (libertad 3). El acceso al código fuente es un prerequisite para esto”*

El Software libre no implica que sea gratis, el Software libre puede ser vendido si así se desea pero este debe estar acompañado de servicios al usuario, ser empaquetado como producto final y brindar servicios de soporte a usuarios y también sus respectivas actualizaciones.

Sin embargo, la ventaja del Software libre es que el usuario tiene el privilegio de regalar, vender, alquilar y copiar el Software ya que la licencia GPL permite la copia legal.

A continuación se definen algunas terminologías utilizadas para el tipo de Software libre:

- Open Source Software: El principal objetivo de este tipo de Software es crear estándares abiertos, se fundó en un principio para el sistema operacional Unix.
- Open Software Initiative: Su objetivo es promover el código abierto, enfatiza en la producción de mejoras al Software y así crear productos nuevos en beneficio de los desarrolladores primordialmente, este concepto contrasta con el propuesto anteriormente ya que su principal beneficiario son los usuarios. Es liderado por Erik Raymond.
- Freeware: Es un tipo de licencia que no tiene costo y ayuda a promover otros desarrollos sin ser de código abierto, como por ejemplo java que no entrega los programas fuentes.
- Shareware: Este término se refiere más bien a una técnica de mercadotecnia la cual consiste en dejar que el usuario utilice el Software por un tiempo y con limitaciones y hasta que decide comprarlo
- Charityware o Careware: En este tipo de Software se liberan los productos pero solicita cooperaciones voluntarias a los usuarios que los adquirieron, como por ejemplo el Software *vi*, y algún otro Software de dominio público en donde el autor renuncia a todos los derechos de manera explícita para beneficio de los demás.
- *Copyleft* (“*Izquierda de copia*”): Su principal característica de este tipo de Software que también es libre es que obliga a que sus mejoras también sean libres. Está en contraposición con los Copyright (“Derecho de copia”)(SGUERRA, 2009, págs. 2-4)

Cuando uno se tienen derechos de copyleft se garantiza que cualquier persona pueda redistribuir el Software, lo cual da al usuario la libertad de modificarlo o copiarlo, y estos derechos permanecen vigentes para todos los próximos usuarios.

El derecho de copyleft exige que un programa de Software copyleft al modificarlo o extenderlo también respete los mismos derechos.(FREE SOFTWARE FOUNDATION, 2012)

De los principales problemas que debe afrontar el Software libre se basa en la desconfianza de los usuarios hacia lo nuevo, por esta razón los competidores sacan ventaja y tratan de desacreditarlo.(SGUERRA, 2009, pág. 15)

En el Software libre también existen algunos tipos de licencias, las cuales se explicaran a continuación:

- *Berkeley Software Distribution (BSD)*

Permite la distribución del código fuente y del código binario sin obligar al autor a darlo y da el crédito a los autores. Fue desarrollada por la *Universidad de Berkeley* en EE.UU. Algunos Software que usan este tipo de licencia son: Servidor web apache, el protocolo de ventanas *Xwindow* o la plataforma de desarrollo *Tcl/Tk*.

- *General Public Licence (GPL)*

Fundada la promotora del proyecto GNU, la *Free Software Foundation*, es llamada también “copyleft” que hace referencia a “todos los derechos revocados”. En este tipo de licencia se obliga a incluir los programas fuentes y los binarios, se permiten modificarse y las nuevas versiones también pertenecerán a la licencia GPL.

- *Licencia Pública General Menor (LGPL)*

Se aplicó en un principio para proteger las librerías de los programas, L hacía referencia a *Library* y con el tiempo se cambió su significado a *Lesser*. Esta licencia permite el uso de programas libres con programas propietarios como se realiza con *Mozilla* u *OpenOffice.org*.(SGUERRA, 2009, págs. 23-24)

2.5.3 Características de hardware para las aplicaciones móviles

Es muy importante tomar en cuenta que características hardware tiene el dispositivo móvil para el cual se realizará una aplicación, ya que varía de un dispositivo móvil a otro.

En el momento de decidir para qué dispositivo móvil estará enfocada la aplicación, se debe ser consciente de los recursos hardware que este proporciona.

Algunas de las características que se puedan encontrar son como por ejemplo el tamaño de pantalla y si es que cuenta con sensores, cámara integrada, micrófono, pantalla táctil, entre otros. En esta sección se describirán las características más generalizadas que se tienen en los dispositivos móviles utilizados actualmente:

- *Pantallas táctiles:* Son también conocidas como pantallas tipo touch, son dispositivos electrónicos que pueden interactuar con los usuarios de forma directa sin necesidad de un teclado o un ratón, basta simplemente con que la persona deslice su dedo sobre ella para poder usarla. Se le clasifica como un hardware de entrada/salida.
- *Sensores:* Un sensor es un dispositivo que detecta estímulos externos y responde a estos, puede también transformar magnitudes físicas o químicas en magnitudes eléctricas. Un sensor permite tener información del entorno.(DEFINICION DE, 2008, pág. s/p)

¿Tipos de sensores en los dispositivos móviles?

Son diversos los sensores utilizados actualmente y están integrados principalmente en los teléfonos inteligentes, a continuación se mostraran los sensores que existen y sus funcionalidades.

- Acelerómetro
- Giroscopio
- Sensor de orientación
- Otros

Estos sensores no son muy comunes entre los dispositivos móviles, sin embargo vale la pena hacer mención de ellos, y son los siguientes: Sensor de luz, de proximidad, de temperatura, de presión, de magnetismo y sensor de humedad.

- *Cámara:* Permiten tomar fotografías video, su resolución es medida en pixeles, las imágenes captadas pueden ser reproducidas al instante, ya que pasan directamente la información a la memoria del dispositivo.
- *Tarjeta de red:* Dispositivo electrónico que brinda la capacidad de conexión a una red.
- *Memoria:* Sirve para almacenar la información de un dispositivo, puede ser interna o externa el rango de capacidad actualmente se encuentra en los Terabytes en el caso de las computadoras personales en cuanto a los dispositivos móviles los más recientes tienen capacidad de hasta 64 Gigabytes de memoria interna.
- *Teclado:* Es un dispositivo hardware, considerado de entrada, ya que por medio de él se puede introducir la información.

2.5.4 Seguridad para aplicaciones móviles

Las aplicaciones para dispositivos móviles no están exentas de tener vulnerabilidades que puedan ser explotadas por intrusos que deseen apoderarse de datos e información personales y causar severos daños.

Los hackers cibernéticos ahora enfocan sus objetivos en las aplicaciones para los dispositivos móviles, debido al auge que tiene y la gran aceptación entre los usuarios de la tecnología.

Estudios recientes indican que ha aumentado el uso de dispositivos móviles; principalmente los teléfonos inteligentes, como principal medio de acceso a Internet en comparación con las computadoras personales, debido a su característica de movilidad y también al reciente incremento de los puntos de acceso gratuito a la red de Internet.

Por ello es indispensable tener en cuenta que no debemos bajar la guardia en cuanto al tema de seguridad para aplicaciones móviles.

De acuerdo con (PIXÉLES, 2012, pág. T6) el virus troyano llamado Boxer afecta a teléfonos inteligentes que funcionan con el sistema operativo Android y la forma de infectarse es al momento de descargar una aplicación que porte esta amenaza, este malware tiene un gran nivel de propagación y por lo tanto representa un peligro.

Según (UNAMCERT, 2007, págs. 1-2) algunos ejemplos de malware que atacan a los dispositivos móviles son *Cabir*, *Comwar* y *Skuller.gen*. Se señala también que el auge en el incremento de la tecnología ha impulsado el desarrollo del malware para dispositivos móviles debido a la amplia capacidad de procesamiento, la sincronización de los equipos móviles con equipos de escritorio, el uso de aplicaciones empresariales y personales que permiten almacenar datos e información valiosa.

El malware puede ocasionar los siguientes problemas una vez instalado en un dispositivo móvil:

- Puede aprovecharse de las vulnerabilidades y robarse la información personal.
- Replicarse hacia los contactos guardados en la memoria del dispositivo y continuar la infección.
- Establecer contacto remoto y tomar el control del dispositivo el cual es capaz de realizar llamadas, enviar mensajes y también de ingresar a la red de Internet.
- *Smishing*, es el encargado de buscar la información confidencial dentro del contenido de algún mensaje de texto.

2.5.5 Software de desarrollo más populares

- iOS

Es un sistema operativo móvil desarrollado por la empresa Apple, se utiliza en dispositivos móviles desarrollados por la misma empresa tales como *iPhone*, *iPad* e *iPod touch*. Las aplicaciones son distribuidas a través de Apple Store. *iOS 9 SDK* es la versión más reciente en el año 2015 para el desarrollo de aplicaciones para dispositivos móviles de la marca Apple. Y para poder incursionar en desarrollo de iOS Apps se requiere tener una computadora Mac que tenga el sistema operativo *OS X v10.10.3 (Yosemite)* o mayor y contar con el programa Xcode Tools 7.0. También se debe contar con el registro en el

Programa de Desarrolladores Apple con el fin de que el dispositivo quede autorizado para desarrollo de aplicaciones.(Apple Inc, 2015)

- Android

De acuerdo con (DEITEL & DEITEL, pág. 15) el sistema operativo Android fue desarrollado en el año 2005 por Android Inc. la cual pertenece a Google y se basa en el kernel de Linux y en Java.

En el año 2005 se formó un grupo compuesto por 34 compañías, reunidas para continuar con el desarrollo de Android y para el año 2010 el número de compañías llegó a ser 79, el éxito fue muy abrumador de tal forma que para finales del año 2014 se activaban 350 millones de teléfonos inteligentes con el sistema operativo Android (Figura 9). Este sistema operativo es utilizado en distintas marcas de teléfonos inteligentes, en dispositivos lectores electrónicos, en computadoras tipo tableta, en pantallas tipo touch, autos, robots y reproductores multimedia. Para Julio del 2015 se estima que existen cerca de 1,600,000 millones de aplicaciones Android en el Google Play (NUMBER OF APPS AVAILABLE IN LEADING APP STORES AS 2015,2015)

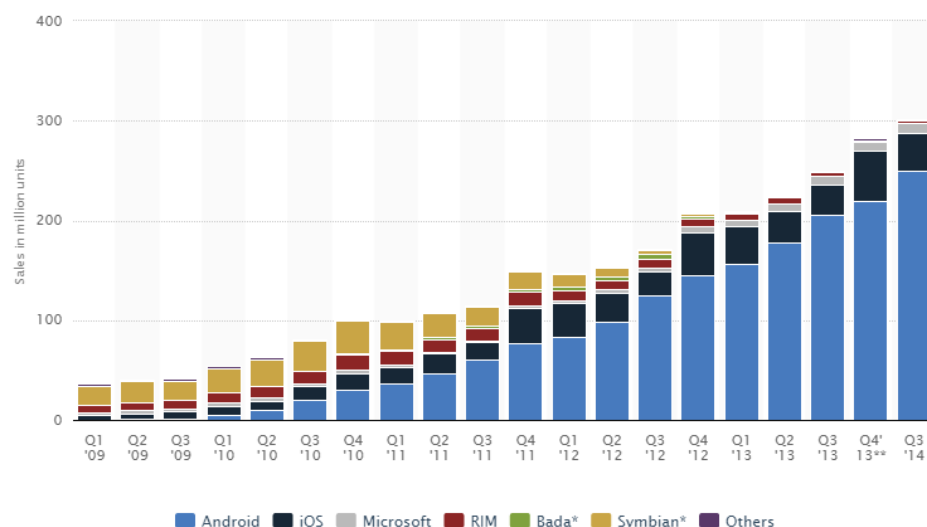


Figura 9 - Venta de teléfonos inteligentes a usuarios finales desde el primer cuarto del 2009 al tercer cuarto del 2014 (Global smartphone sales to end users from 1st quarter 2009 to 3rd quarter 2014, by operating system (in million units),2015)

En el mes de Octubre del 2015 se lanzó al mercado la versión Marshmallow 6.0 (ANDROID DEVELOPERS BLOG, 2015).

De acuerdo con (MEIER, 2010, págs. 29-30), la mayor parte de las aplicaciones creadas estarán delimitadas principalmente dentro de las siguientes categorías:

- *Aplicaciones de primer plano:* Es un tipo de aplicación que solo se puede utilizar cuando se encuentra en primer plano, y cuando no es visible está suspendida, un ejemplo de este tipo de aplicaciones son comúnmente los juegos. Al crear una aplicación de primer plano, se recomienda salvar su estado, ya que cuando el usuario recibe una llamada, accede a otra aplicación, recibe un mensaje o alguna otra actividad, la aplicación de primer plano quedara en espera.
- *Aplicaciones en segundo plano:* Corresponde a un tipo de aplicación con interacción limitada, con excepción al momento en que es configurada por el usuario, pasan la mayor parte del tiempo oculto. Ejemplos de estos servicios son los de auto respuesta a mensajes, las alarmas, recordatorios, etc. Se encuentran constantemente escuchando los mensajes o acciones enviadas por el hardware, el sistema, o alguna otra aplicación, más que depender propiamente del usuario.
- *Aplicaciones intermitentes:* Estas aplicaciones esperan algún tipo de actividad pero la mayor parte del tiempo trabajan en segundo plano. A menudo dichas aplicaciones son ejecutadas silenciosamente y envían alguna notificación a los usuarios cuando es pertinente. Ejemplo: el reproductor de música.
- *Aplicaciones reproductoras:* Algunas aplicaciones solo son reproducidas y mostradas en la pantalla principal, como un calendario o un reloj.

- Windows Mobile

Diseñado por Microsoft anteriormente solo estaba enfocado a dispositivos móviles, principalmente a teléfonos inteligentes, existen diferentes versiones de sistemas operativos debido a la diversidad de dispositivos y sus diferentes características propias del hardware como lo son la pantalla tipo touch, entre otras características, sin embargo

desde que se lanzó Windows 8 se introdujo Windows Runtime (WinRT), una evolución del modelo de desarrollo de Windows app. WinRT permitió que las aplicaciones creadas para móviles puedan también ser usadas en Windows. Con el lanzamiento de Windows 10 se lanza *Universal Windows Platform* (UWP), que es un modelo más avanzado del WinRT. (MICROSOFT, 2015)

Capítulo 3.– Metodología

3 Metodología para el desarrollo de Objetos de Aprendizaje y su adaptación en entornos móviles

Actualmente no se cuenta con una metodología base para el desarrollo de aplicaciones móviles, ya que cada desarrollador se adapta a las metodologías existentes para la creación de Objetos de Aprendizaje y los adaptan a entornos móviles.

En este trabajo se utiliza la metodología ADDIE (por sus siglas en ingles Analysis, Design, Development, Implementation and Evaluation) para la creación de Objetos de Aprendizaje.

La metodología ADDIE, es un modelo genérico de diseño instruccional, bastante flexible que permite la modificación y elaboración basada en las necesidades de la situación Instruccional, consta de 5 pasos (Figura 10): Análisis, Diseño, Desarrollo, Implementación, y Evaluación de los materiales de aprendizaje y las actividades.

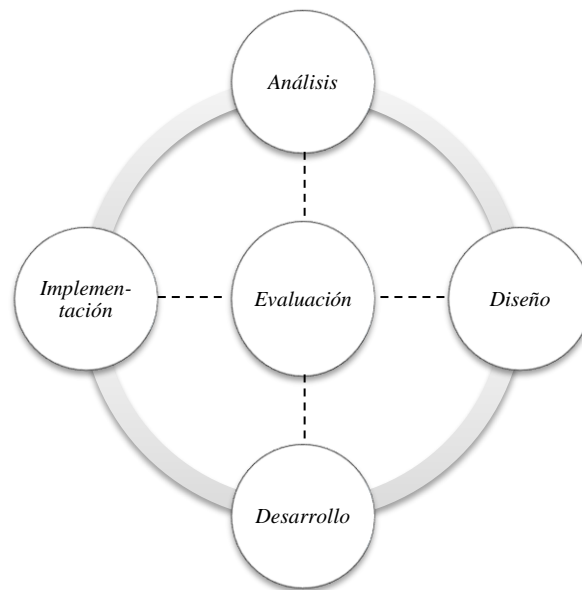


Figura 10 - Concepto del Modelo ADDIE (MARIBE, 2009)

3.1 Análisis

Debido al alto índice de reprobación que tiene la materia de Programación, se busca implementar objetos de aprendizaje, que ayuden a reforzar los conocimientos adquiridos en clase por parte de los alumnos, se aprovechan las ventajas de la tecnología móvil ya que pueden ser consultadas por el alumnado en cualquier momento y espacio con mayor facilidad. En esta etapa se plantearán los objetos de aprendizaje y la estrategia para atacar un determinado problema de aprendizaje.

3.2 Diseño

Al diseñar un recurso didáctico que será utilizado por profesores y alumnos, se debe tener en cuenta la manera en que se desarrolla el proceso de enseñanza-aprendizaje dentro del aula para cumplir ampliamente el objetivo por el cual fue creado dicho recurso. En esta etapa se plantean los contenidos que se manejarán dentro del juego y la estrategia para atacar los problemas de aprendizaje. Se busca que los objetos de aprendizaje tengan elementos de todos los estilos de representación de información que faciliten la adquisición del conocimiento para todos los estudiantes sin importar su estilo de representación predominante.

3.3 Desarrollo

Los objetos de aprendizaje móviles que se desarrollarán será una aplicación móvil creada para los dispositivos móviles con sistema operativo Android, enfocado principalmente para teléfonos inteligentes (smartphone) y tabletas electrónicas. La aplicación será para la materia de Programación, la cual contemplará los contenidos de mayor dificultad de aprendizaje para el alumno. Esta fase es destinada a producir el OA en formato digital y almacenarlo en un repositorio.

La metodología de desarrollo utilizada es Ágil Scrum para su adaptación a entornos móviles, como ya se mencionó anteriormente esta cuenta con características de desarrollo que satisfacen algunas necesidades en las aplicaciones móviles, como lo son: su usabilidad en equipos

de trabajo no muy grandes, la adaptabilidad a los cambios en los requerimientos de la aplicación y el tiempo de desarrollo es muy limitado.

3.4 Implementación y Evaluación

La aplicación móvil creada está enfocada principalmente para teléfonos inteligentes (smarphones) y tabletas que contengan sistema operativo Android. Los alumnos podrán descargar esta aplicación desde una tienda virtual o instalar el archivo con extensión *.apk directamente en sus teléfonos o tabletas electrónicas.

Para evaluar la aplicación se pretende que los maestros de Programación de la Facultad de Ciencias Químicas e Ingeniería de la UABC, la incorporen como herramienta de apoyo en su clase, una vez que los alumnos la tengan instalada y hagan uso de ella se les aplicara una encuesta para conocer su opinión acerca de los niveles y el diseño de la aplicación.

Es importante evaluar constantemente cada una de las etapas y revisar la evaluación del proceso tanto como sea necesario.

Capítulo 4.– Desarrollo

4 Caso de estudio: Objetos de Aprendizaje en entornos móviles para el apoyo a la enseñanza de Programación Estructurada

4.1 Análisis

En esta etapa se determinaron los temas del curso de programación que requieren mayor atención. Esta selección de temas se llevó a cabo mediante una revisión de la carta descriptiva de Programación en conjunto con el análisis de los resultados de exámenes colegiados. En base a este análisis se determinaron cuáles son los temas en los que los alumnos tienen mayor dificultad para aprender, éstos temas las estructuras iterativas y el desarrollo de la lógica.

El examen Colegiado fue aplicado a 8 grupos de Segundo semestre que cursaban la material de Programación Estructurada en el periodo 2013-2 en la FCQI. El examen consta de 40 reactivos. Se analizaron 12 reactivos con mayor índice reprobatorio por grupo, es decir, se tomaron aquellos reactivos en la que la mayoría de los grupos obtuvieron una calificación menor que la media en la FCQI.

- Se determinó que los temas con mayor deficiencia son:
 - Algoritmos
 - Zonas de memoria
 - Expresiones Básicas
 - Función
 - Ciclos
 - Arreglos Unidimensionales
 - Arreglos Bidimensionales

4.2 Diseño

Como parte del análisis realizado, se detectó la necesidad de un sistema que sea portátil y que este enfocado especialmente a las necesidades de los estudiantes de Programación, por lo que

se contempló una aplicación móvil para dispositivos con sistema Operativo Android, nombrado El Fox & Robot.

De acuerdo al diseño del objeto de aprendizaje presentado en este trabajo no se incluyen los temas de arreglos ni el de zonas de memoria.

4.3 Desarrollo

Para el desarrollo de la aplicación móvil se utilizó la metodología SCRUM por el tipo de aplicación que se desarrolla se necesita tener bloques temporales cortos y fijos (sprints, iteraciones). En el que cada sprint (iteración) tiene que proporcionar un resultado completo para poder a completar la aplicación (ver Figura 11).

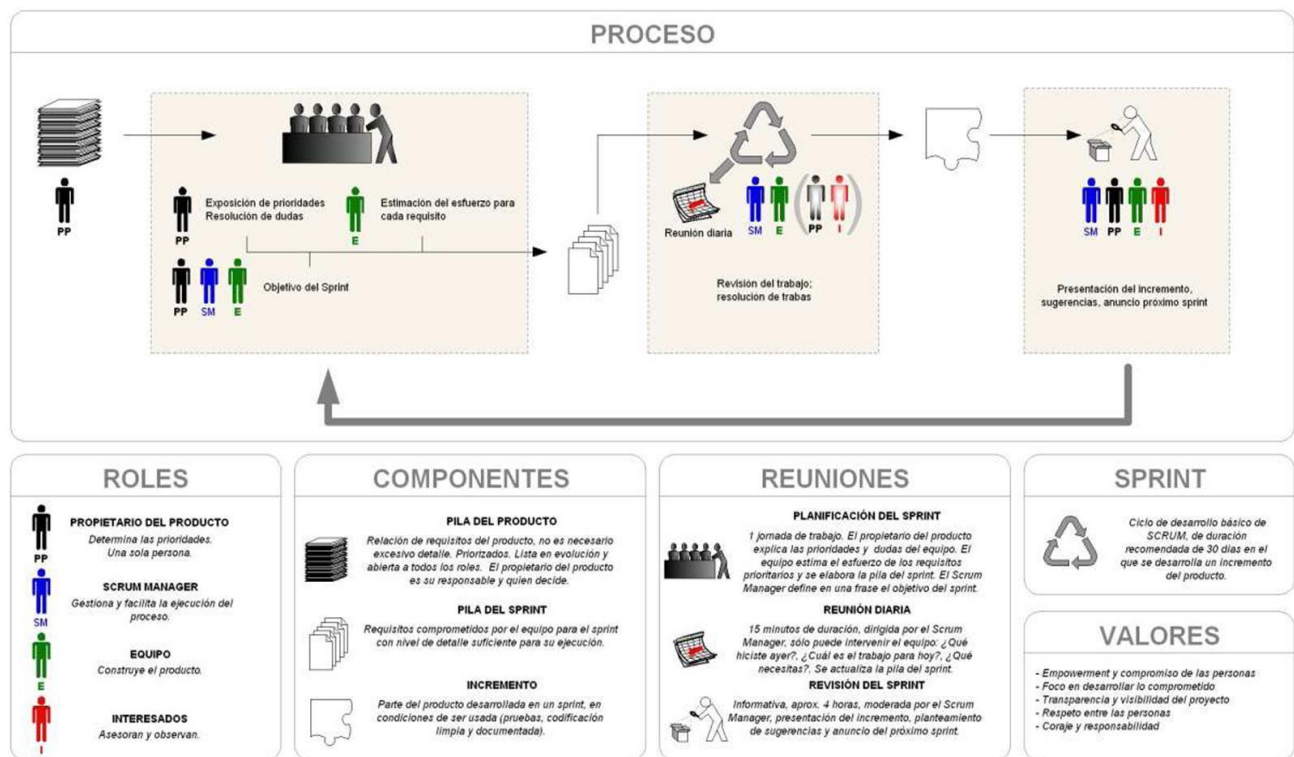


Figura 11 - Proceso de Metodología SCRUM (Disponible en https://commons.wikimedia.org/wiki/File:Ficha_scrum.jpg)

- Sprints utilizados para el desarrollo de la aplicación móvil El Fox & Robot:
 - La Funcionalidad de los objetos principales (el Fox, El Robot, El diskette)
 - La alimentación de las instrucciones a los objetos
 - Desarrollo de la interfaz gráfica
 - Creación de los niveles
 - Creación de los menús
 - Crear aplicación en Android

Para el desarrollo de la aplicación móvil se utilizó la herramienta GameMaker: Studio, que es una completa herramienta de creación de videojuegos multiplataforma capaz de exportar cualquier desarrollo a Windows, Mac, iOS, Android o HTML 5. Estos tres últimos, solo están disponibles para las versiones de pago de la aplicación. En este caso se compró la versión de Android.

Se eligió realizar la aplicación en la herramienta GameMaker ya que cuenta con una interfaz muy intuitiva desde la que se pueden gestionar todas las fases del desarrollo del videojuego mediante una filosofía 'drag and drop' que evita la necesidad de conocer un lenguaje de programación en profundidad para crear un juego profesional. Aún así, dispone de su propio lenguaje de programación integrado (el llamado Game Maker Language) con el que se crearon rutinas propias.

Game Maker Language (GML) es un lenguaje de programación interpretado desarrollado para usarse como complemento de la aplicación de desarrollo de juegos Game Maker. El lenguaje es utilizado en el programa para dar mayor potenciación al desarrollo de juegos u otros sistemas dentro de tal. El lenguaje fue codificado en el lenguaje Delphi. Es un lenguaje de programación muy flexible, su sintaxis es comparable con la de C++, aunque también se ve influenciado por la sintaxis de otros lenguajes como Pascal. La estructura general del código GML puede tomar diferentes formas de sintaxis, ya que no se ve sujeta a una sintaxis estricta como la de C. Es capaz de trabajar con números, cadenas de texto y demás de manera nativa.

A continuación se presentan los sprints creados en la herramienta GameMaker: Studio para la creación de la aplicación móvil el Fox y el Robot:

- Sprint 1 - Programación de las funcionalidades básicas de los objetos el Fox, El Robot, El diskette (Figura 12). En la Figura 13 se muestra la programación del objeto Robot.

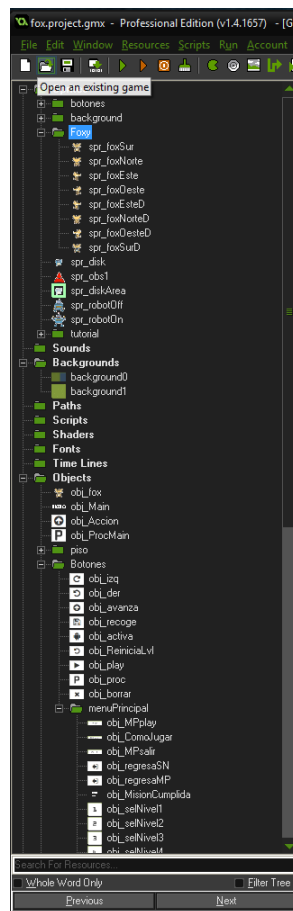


Figura 12 – GameMaker: Objetos principales: el Fox, el Robot, el Diskette (Fuente Propia)

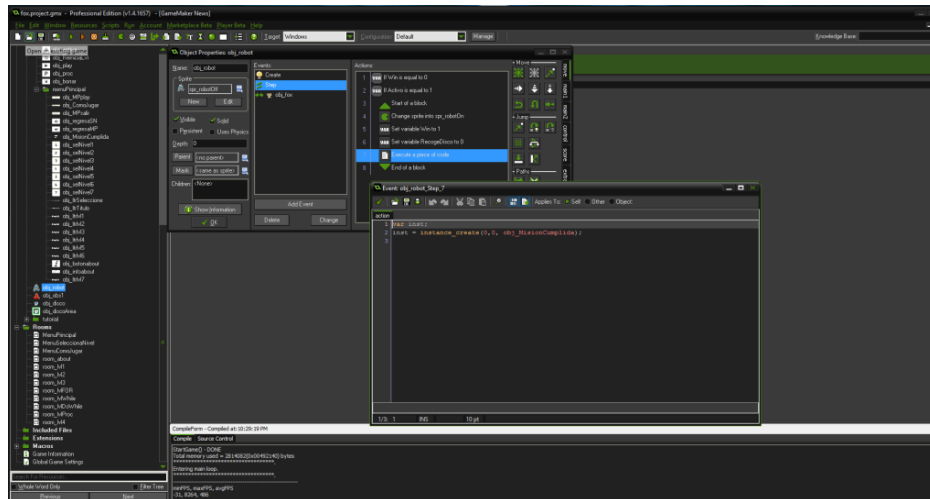


Figura 13 – GameMaker: Programación del objeto Robot (Fuente Propia)

- Sprint 2 – Programación de las entradas de información que tendrá el juego para que den movimiento a alguno de los objetos (Figura 14 y Figura 15).

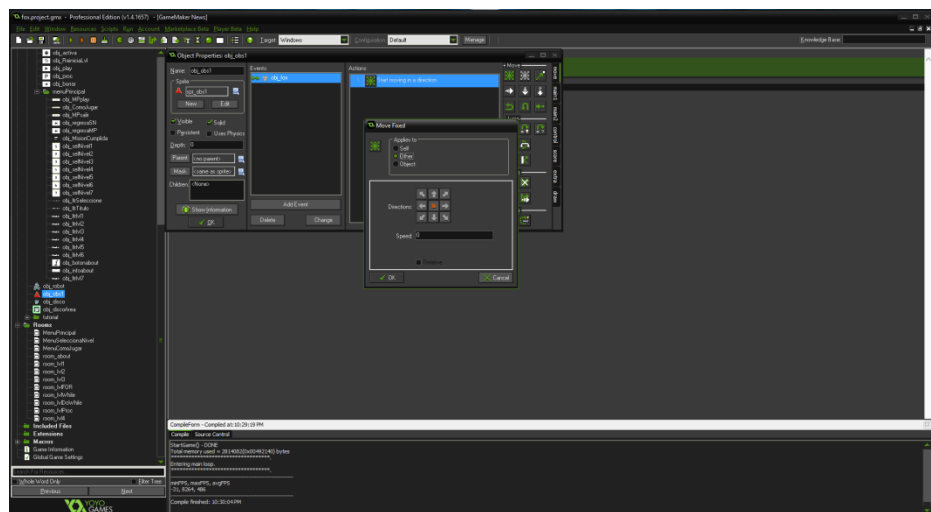


Figura 14 – GameMaker: Programación de movimiento (Fuente Propia)

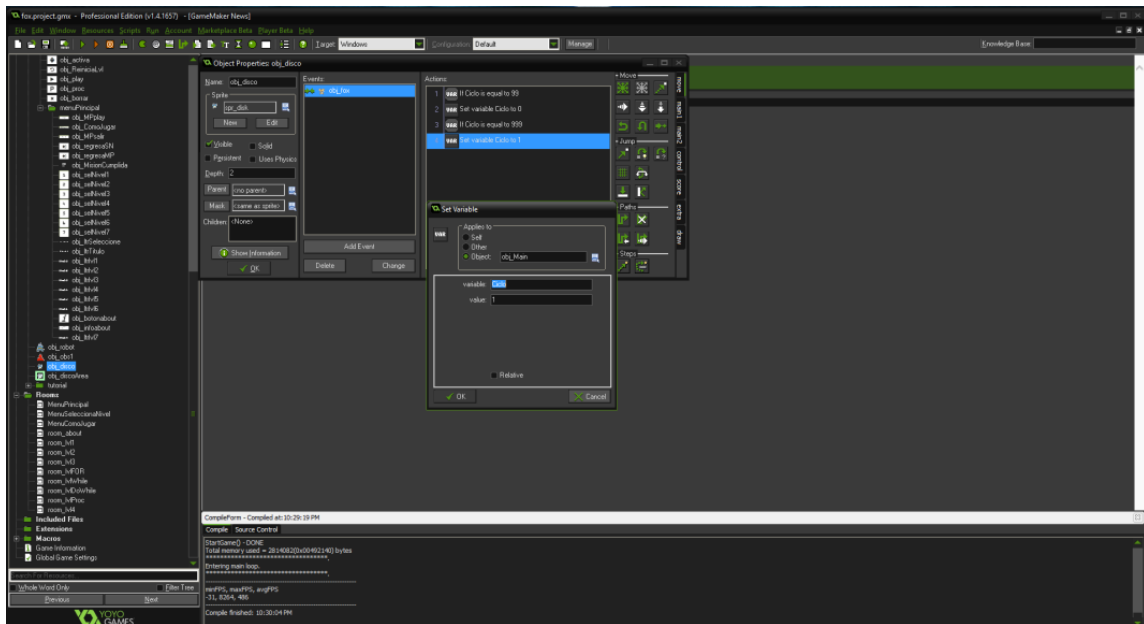


Figura 15 – GameMaker: Programación de objetos (Fuente Propia)

- Sprint 3 - Desarrollo de la interfaz gráfica; menú principal y 7 niveles (Figura 16 y Figura 17).

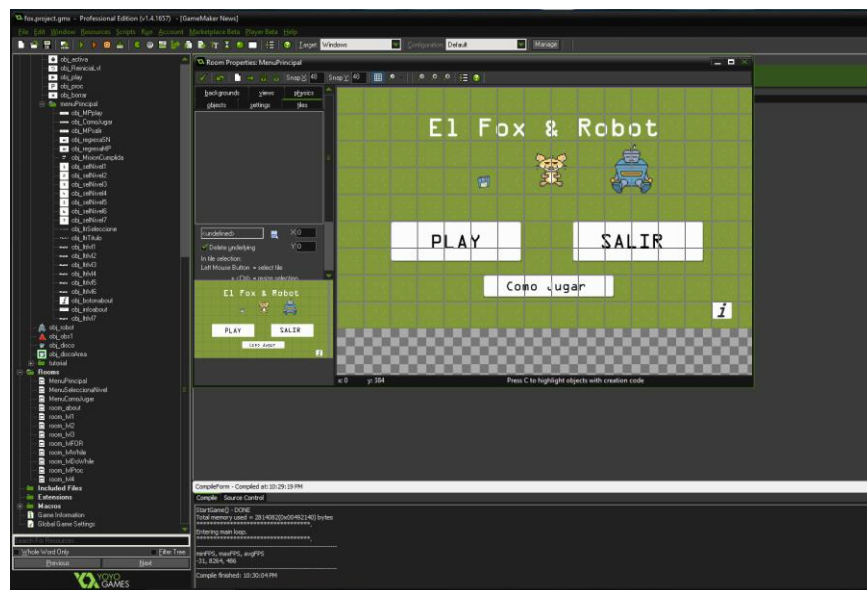


Figura 16 – GameMaker: Interfaz gráfica de menú principal (Fuente Propia)

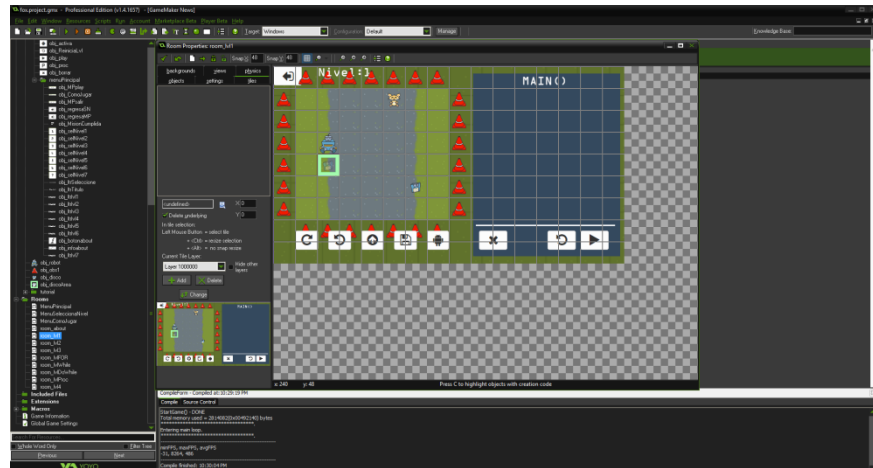


Figura 17 – GameMaker: Diseño de nivel 1 (Fuente Propia)

- Sprint 4 – Eliminación de bugs en los 7 niveles (Figura 18) y desarrollo de menús (Figura 19 y Figura 20).

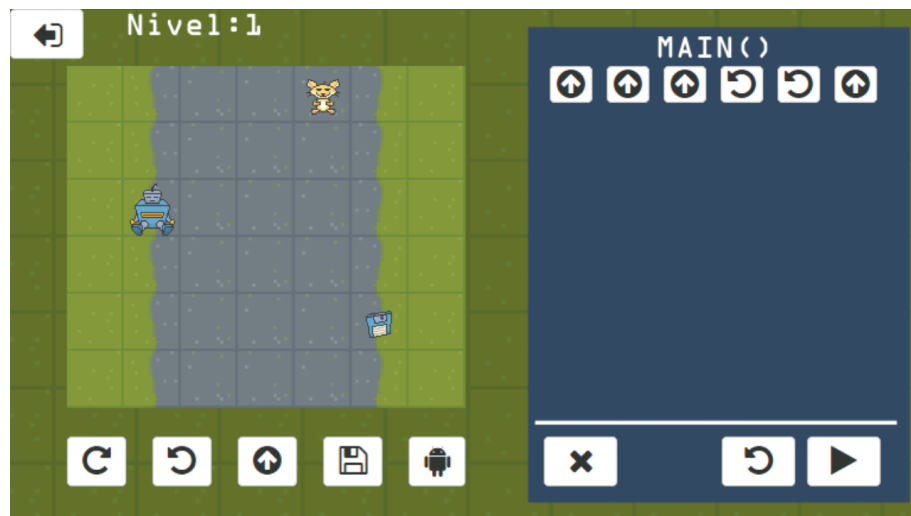


Figura 18 – GameMaker: Nivel 1 Secuencia de Instrucciones (Fuente Propia)

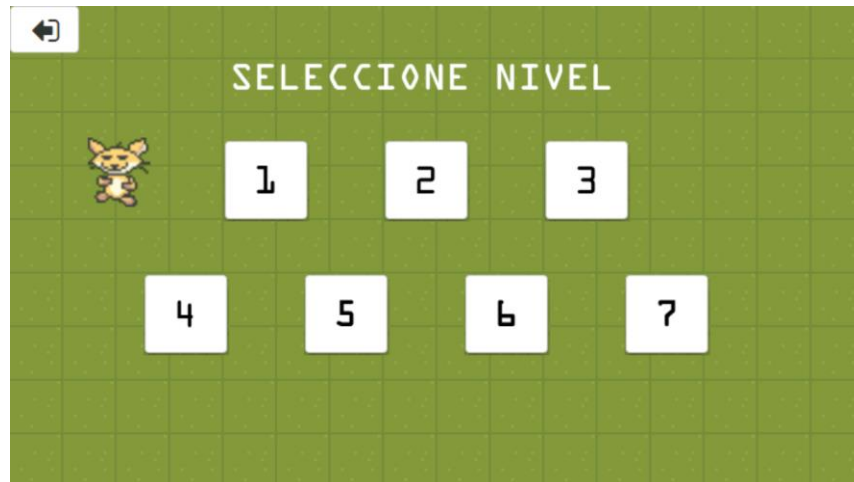


Figura 19 – GameMaker: Menú para elegir el nivel a jugar (Fuente Propia)



Figura 20 – GameMaker: Menú principal (Fuente Propia)

- Sprint 5 – Generación de archivo .apk en GameMaker, para compatibilidad de la aplicación con dispositivos móviles que tengan Sistema Operativo Android (Figura 21, Figura 22 y Figura 23).

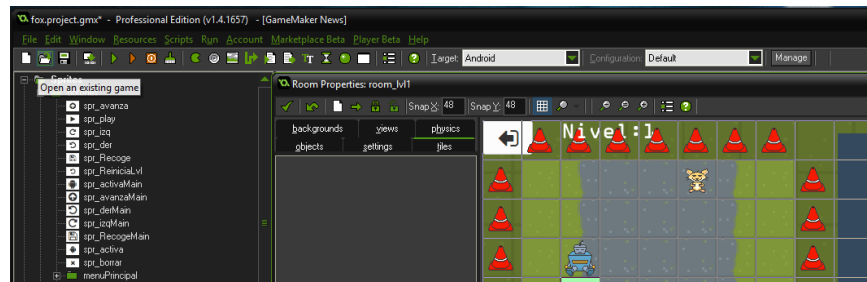


Figura 21 – GameMaker: Seleccionar el tipo de sistema operativo Android (Fuente Propia)

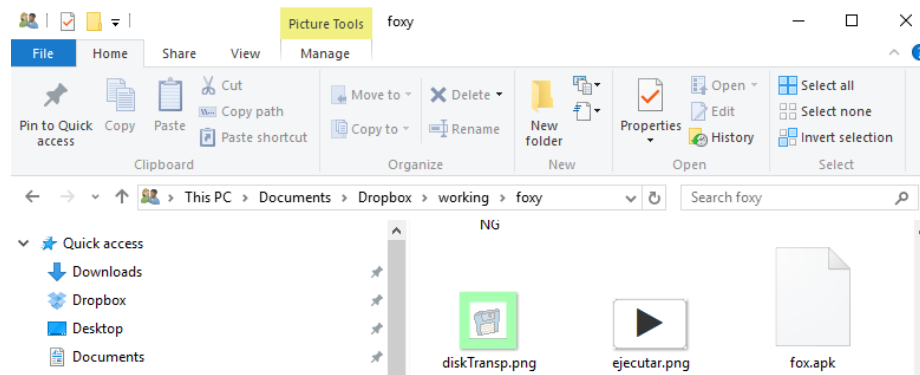


Figura 22 – GameMaker: Guardar archivo en formato .apk (Fuente Propia)

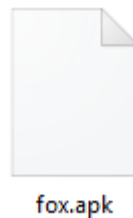


Figura 23 – GameMaker: Archivo ejecutable (.apk) (Fuente Propia)

4.4 Implementación y Evaluación

Se pretende que los objetos de aprendizaje realizados en este trabajo se implementen en todos los grupos de Programación de la Facultad de Ciencias Químicas e Ingeniería y así mismo poder evaluar su funcionalidad, sin embargo estas dos etapas quedan contempladas para las siguientes etapas del proyecto así como agregar niveles referentes a arreglos.

Capítulo 5.– Pruebas y Resultados

5 Funcionamiento de los prototipos de objetos de aprendizaje

A continuación se mostrará el funcionamiento de los prototipos de objetos de aprendizaje, su funcionamiento fue probado en un teléfono inteligente modelo Moto G (2da generación) Fabricado por Motorola con sistema operativo Android versión Lollipop (5.0.2) (ver Figura 24) y en una tableta electrónica marca Samsung, modelo GT-N8013 con sistema operativo Android versión Jelly Bean (4.1.2) (ver Figura 25).

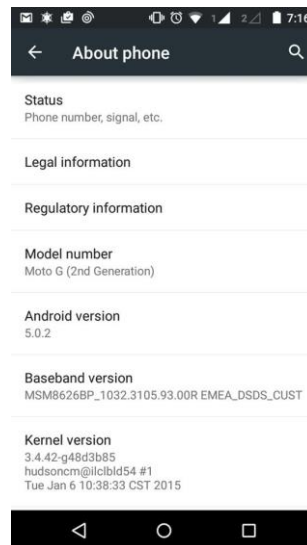


Figura 24 - Información del teléfono inteligente utilizado para pruebas (Fuente Propia)

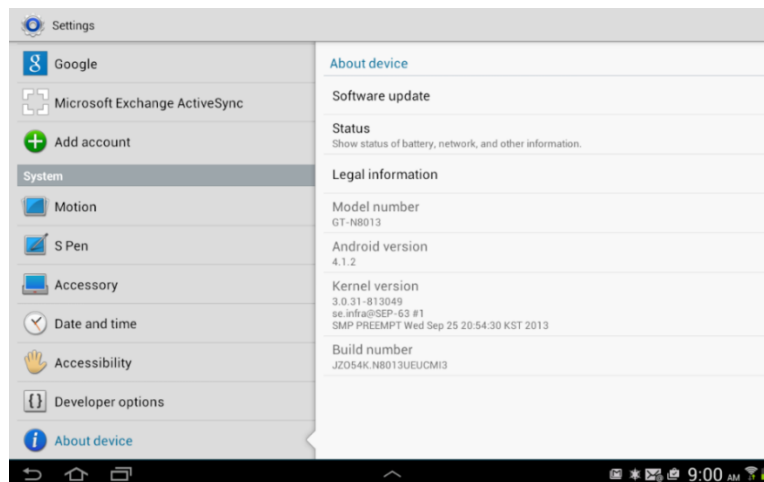


Figura 25 - Información de la tableta electrónica utilizado para pruebas (Fuente Propia)

A continuación se mencionan los pasos que se siguieron para probar las funcionalidades de la aplicación móvil (en este trabajo solo mostramos el resultado obtenido en la instalación del teléfono inteligente):

1. Descarga de la aplicación – Se descargó la aplicación (Figura 26) y se instaló en el teléfono inteligente Motorola modelo Moto G (Figura 27).

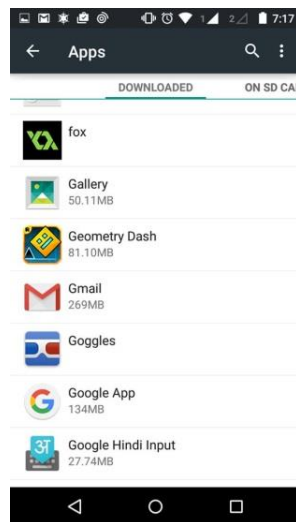


Figura 26 - Descarga de aplicación en el teléfono inteligente Motorola modelo Moto G (Fuente Propia)

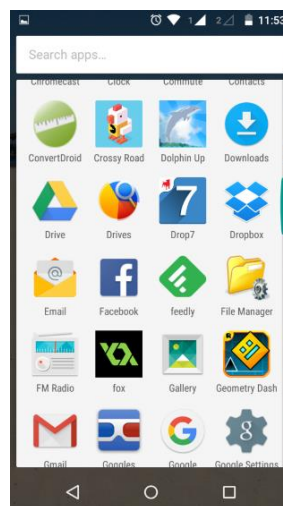



Figura 27 - Instalación de aplicación en el teléfono inteligente Motorola modelo Moto G (Fuente Propia)

2. Ejecución de la aplicación - Al ejecutar el Icono se observa la siguiente pantalla (Figura 28) donde el alumno tiene 4 opciones: Jugar, salir, ver el tutorial de como jugar o ver la información de cómo jugar.



Figura 28 - Pantalla principal (Fuente Propia)

3. Revisión del botón de “Como jugar”- Este botón le mostrara al alumno una serie de pantallas en la cual cada una de ellas tendrá una breve descripción de las instrucciones que le puede dar a El Fox (ver Figura 29 - Figura 33). Para regresar al menú anterior en cualquier momento el alumno debe de presionar el botón .

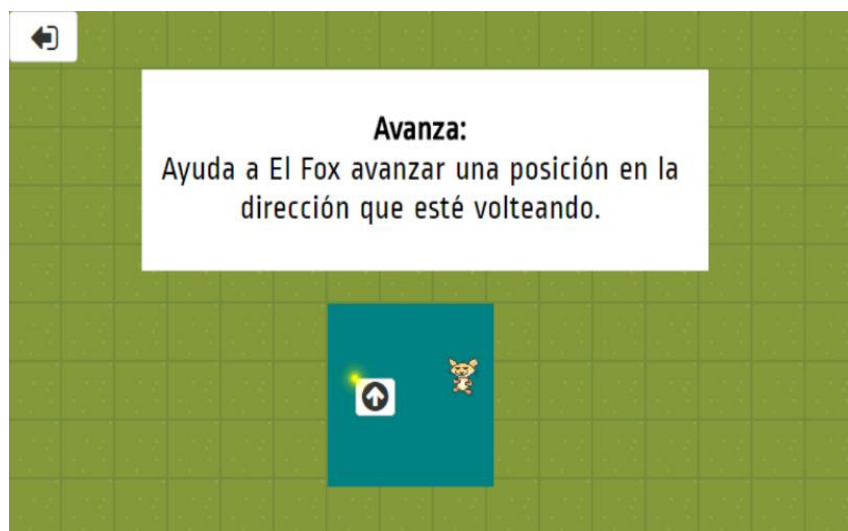


Figura 29 – Pantalla de Ayuda: Instrucción Avanza (Fuente Propia)



Figura 30 - Pantalla de Ayuda: Instrucción Rota Derecha (Fuente Propia)



Figura 31 - Pantalla de Ayuda: Instrucción Rota Izquierda (Fuente Propia)



Figura 32 - Pantalla de Ayuda: Instrucción Recoge Disquette (Fuente Propia)



Figura 33 - Pantalla de Ayuda: Instrucción Activa Robot (Fuente Propia)


4. Revisión del botón de “Información”.- Al presionar este botón se le mostrara al alumno la información básica de la aplicación como el nombre, versión, entre otras cosas (ver Figura 34) . Para regresar al menú anterior el alumno debe de presionar el botón .



Figura 34 – Pantalla de información de la aplicación (Fuente Propia)

5. Revisión del botón de “Play”.- Al presionar este botón el alumno podrá seleccionar 7 niveles en los que desee jugar (Figura 35).

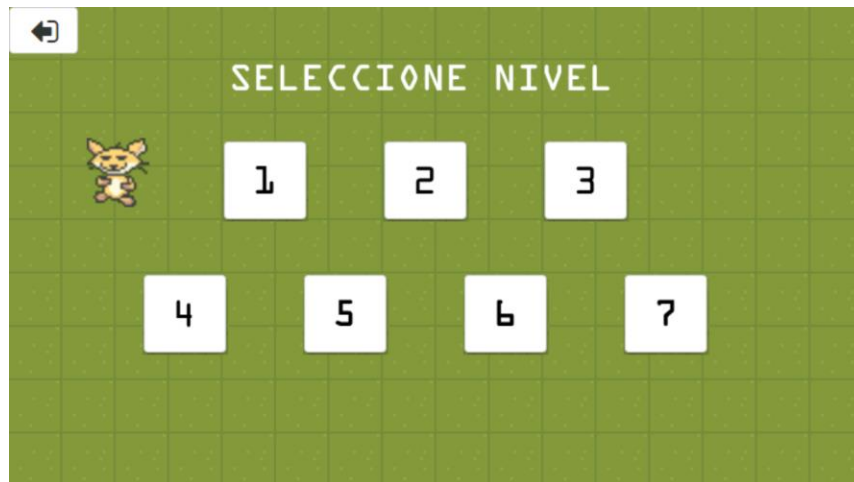


Figura 35 – Menú para seleccionar el nivel a jugar (Fuente Propia)

- a. **Nivel 1.-** El Fox siempre comienza direccionado hacia enfrente y en la misma posición (coordenada) (ver Figura 36), el alumno deberá de encontrar la secuencia de instrucciones que deberá de darle al Fox, para que recoja el Diskette y lo ingrese al Robot (Figura 37). El Robot solo se activa si el Fox está frente a él.



Figura 36 - Nivel 1 : Secuencia de Instrucciones (Fuente Propia)



Figura 37 – Nivel 2 : Nivel 3 : Secuencia de Instrucciones (Fuente Propia)

- b. **Nivel 2 y 3.-** Son muy similares al nivel 1 la diferencia es que hay más obstáculos que evadir y la dificultad para el alumno es mayor (ver Figura 38 y Figura 39).



Figura 38 - Nivel 3 : Secuencia de Instrucciones (Fuente Propia)



Figura 39 – Nivel 3 : Lógica de Instrucciones (Fuente Propia)

- c. **Nivel 4, 5 y 6.-** En estos niveles se introducen ciclos For (Figura 40 y Figura 41), Do (Figura 43) y While (Figura 42), en estos niveles el alumno deberá encontrar la secuencia de instrucciones que van dentro del ciclo para que el Fox pueda recoger el Diskette y llevarlo hacia el Robot.



Figura 40 – Nivel 6 – Ciclo For (Fuente Propia)

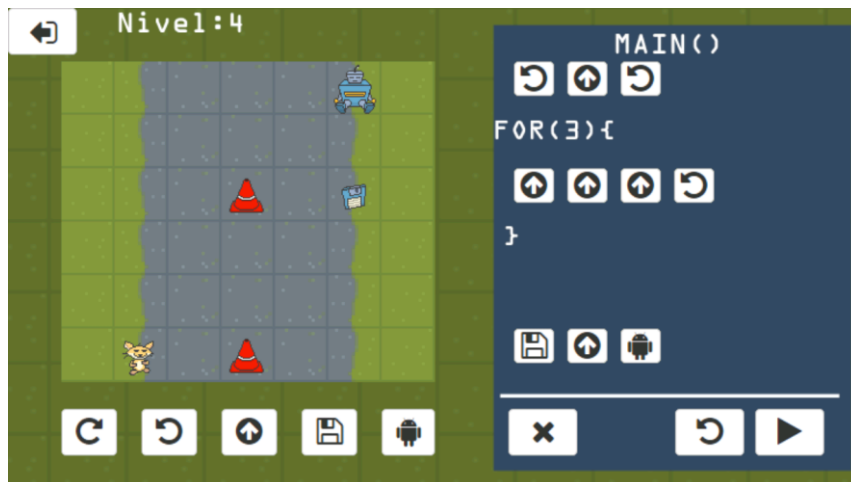


Figura 41 – Nivel 6 : Ciclo For con instrucciones (Fuente Propia)

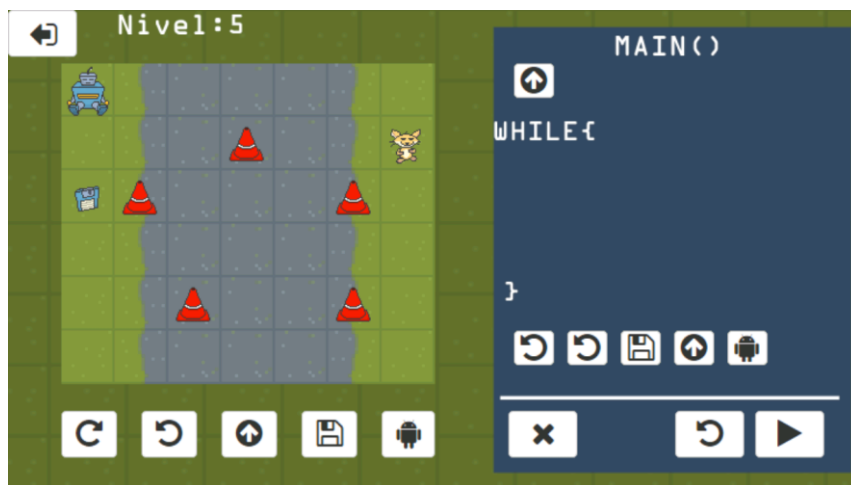


Figura 42 – Nivel 6 : Ciclo While (Fuente Propia)



Figura 43 – Nivel 6 : Ciclo Do (Fuente Propia)

- The screenshot shows a game interface with a 10x10 grid. The grid has a green border and a grey center. A cat is at (1,1), a car is at (2,3), and there are several obstacles (red cones) at (2,2), (2,4), (3,2), (3,4), (4,2), (4,4), (5,2), (5,4), (6,2), (6,4), (7,2), (7,4), (8,2), (8,4), (9,2), (9,4), (10,2), (10,4). The right panel shows the code editor with a 'MAIN()' function and a 'PROC' block.

The screenshot shows the Scratch environment. The stage is a 10x10 grid with a green background. A cat sprite is at (1,1), a car sprite is at (2,3), and there are several cone obstacles at (3,1), (3,2), (4,1), (4,3), (5,1), (5,2), (5,3), (6,1), (6,2), (6,3), (7,1), (7,2), (7,3), (8,1), (8,2), (8,3), (9,1), (9,2), (9,3), (10,1), (10,2), (10,3), (10,4), (10,5), (10,6), (10,7), (10,8), (10,9), (10,10). The 'MAIN()' script area is visible on the right, and the 'PROC:' area is partially visible below it.

76

Capítulo 6.– Conclusiones

6 Conclusiones

Se realizó el análisis de la problemática que surge del alto índice de reprobación de las materias de segundo semestre por parte de los alumnos y sus probables causas, y también algunos problemas detectados por parte de los profesores. Con base en dicha investigación, se confirmó que la incorporación de objetos de aprendizaje, al sector educativo ayuda tanto a profesores como a alumnos, ya que este tipo de materiales educativos son empleados para facilitar el proceso de enseñanza-aprendizaje.

Se diseñaron prototipos de objetos de aprendizaje específicos para la materia de Programación, con el propósito de apoyar en el estudio y refuerzo de los temas: estructuras de control básicas de la programación.

Se espera que las nuevas generaciones de alumnos y maestros de Programación incorporen con mayor auge el uso de dichos dispositivos móviles para reforzar el aprendizaje en el aula y como una herramienta más en el proceso de enseñanza- aprendizaje y con su uso disminuir el alto índice de reprobación.

Además, la aplicación se diseñó como objetos de aprendizaje, que presentan la gran ventaja de ser reutilizables, accesibles y transportables.

Referencias

- ALVARES, L. G. (2004). GRUPO DE INVESTIGACIÓN DE TECNOLOGÍAS DE APRENDIZAJE. Consultado el 1 de Marzo de 2014. Disponible en GITA: www.gita.cl/files/2_Objeto%20de%20Aprendizaje.pdf
- ANDROID DEVELOPERS BLOG. (SEPTIEMBRE de 2015). ANDROID 6.0 MARSHMALLOW COMING TO DEVICES SOON. Consultado el 17 de Noviembre de 2013. Disponible en <http://android-developers.blogspot.com/2015/09/android-marshmallow-ready-for-devices.html>
- APPLE INC. (SEPTIEMBRE de 2015). IOS DEVELOPER LIBRARY – iOS 9 RELEASE NOTES, Consultado el 20 de Septiembre de 2015. Disponible en https://developer.apple.com/library/ios/releasenotes/General/RN-iOSSDK-9.0/index.html#//apple_ref/doc/uid/TP40016202
- ÁVILA, P., & BOSCO, M. D. (ABRIL de 2001). INVESTIGACIÓN ILCE. Consultado el 24 de Noviembre de 2013. Disponible en ambientes virtuales de aprendizaje una nueva experiencia: http://investigacion.ilce.edu.mx/panel_control/doc/c37ambientes.pdf
- BARTOLOMÉ, A. (2004). PIXEL-BIT REVISTA MEDIOS Y EDUCACIÓN. Consultado el 29 de Noviembre de 2013. Disponible en Blended Learning: http://www.lmi.ub.es/personal/bartolome/articuloshtml/04_blended_learning/documentacion/1_bartolome.pdf
- BRAUDE, E. J. (2003). INGENIERÍA DEL SOFTWARE UNA PERSPECTIVA ORIENTADA A OBJETOS. México: ALFAOMEGA.
- CASTRO, E. P., LANDA DURAN, P., & VEGA VALERO, C. Z. (s.f.). APRENDIZAJE AUTORREGULADO: UNA VISIÓN CONCEPTUAL. Consultado el 25 de Febrero de 2014. Disponible en <http://www.iztacala.unam.mx/carreras/psicologia/psiclin/vol9num2/vol9n2art1.pdf>

- D. BRANSFORD, J., L. BROWN, A., & R. COCKING, R. (2007). AMBIENTES DE APRENDIZAJE EN LA ESCUELA. México: SEP.
- DEFINICION DE. (2008). Consultado el 9 de Enero de 2014. Disponible en <http://definicion.de/sensor/>
- DEITEL, P., & DEITEL, H. (2012). JAVA CÓMO PROGRAMAR. México: Pearson.
- DIGERATI ESPAÑA. (2008). LA PARTE FÍSICA DE LA RED. REDES ONE CLICK, 13. EL NACIONAL. Consultado el 26 de Junio de 2014. Disponible en [www.el-nacional.com/ tecnologia/Colombia-subastalicensias-operar-tecnologia_0_215378728.html](http://www.el-nacional.com/tecnologia/Colombia-subastalicensias-operar-tecnologia_0_215378728.html)
- ESPAÑOLA, D. D. (1984). DICCIONARIO DE LA LENGUA ESPAÑOLA (Vol. TOMO II). MADRID.
- ESPAÑOLA, R. A. (2012). DICCIONARIO DE LA REAL ACADEMIA ESPAÑOLA. Consultado el 6 de enero de 2014. Disponible en Real Academia Española: http://buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=Software
- FACULTAD DE INGENIERÍA. LABORATORIO DE REDES Y SEGURIDAD. (s/a). LABORATORIO DE REDES Y SEGURIDAD. Consultado el 6 de Noviembre de 2013. Disponible en <http://redyseguridad.fip.unam.mx/proyectos/Wi/wimax/wimax.html>
- FOROUZAN, B. A. (2002). TRANSMISIÓN DE DATOS Y REDES DE COMUNICACIONES. McGraw Hill.
- FREE SOFTWARE FOUNDATION. (NOVIEMBRE de 2013). EL SISTEMA OPERATIVO GNU. Consultado el 24 de Noviembre de 2013. Disponible en <http://www.gnu.org/home.es.html>

- FREE SOFTWARE FOUNDATION. (9 de Noviembre de 2013). EL SISTEMA OPERATIVO GNU. Consultado el 24 de Noviembre de 2013. Disponible en <http://www.gnu.org/copyleft/copyleft.html>
- FREEDMAN, A. (1993). DICCIONARIO DE COMPUTACIÓN. McGraw-Hill.IEEE. (1993). StandarsCollections:Software Engineering.
- GALEANA, LOURDES DE LA O. (2012). OBJETOS DE APRENDIZAJE. Consultado el 25 de Octubre del 2013. Disponible en http://www.cudi.edu.mx/primavera_2004/presentaciones/Lourdes_Galeana.pdf.
- GALLARDO, F. A. (JUNIO de 2012). BLOG DE LA UNIVERSIDAD VERACRUZANA. Consultado el 19 de Diciembre de 2013. Disponible en medios en los que se utilizan diferentes ambientes de aprendizaje: <http://www.uv.mx/blogs/sea/2012/06/26/medios-que-se-utilizan-en-losdiferentes-ambientes-de-aprendizaje/>
- GLOBAL SMARTPHONE SALES TO END USERS FROM 1ST QUARTER 2009 TO 3RD QUARTER 2014, BY OPERATING SYSTEM (IN MILLION UNITS). Consultado el 1 de Marzo de 2015. Disponible en <http://www.statista.com/statistics/266219/global-smartphone-sales-since-1st-quarter-2009-by-operating-system/>
- GONZÁLEZ, M. B. (AGOSTO de 2009). TESIS DIGITALES. Consultado el 1 de Marzo de 2014. Disponible en http://oreon.dgbiblio.unam.mx:8991/F/IGMGPXSJ17Q6S19Y1UDQ1AIVHFY21UFYAQ1ACSRUK9QUFULMIJ-20785?func=findb&request=BEATRIZ+GARZA+GONZALEZ&find_code=WRD&adjacent=N&local_base=TESISDIG&x=74&y=17&filter_code_2=WYR&filter_request_2=&filter_code_3=WYR&filter_re

- HODELL, C. (2011). ISD FROM THE GROUND UP: A NO-NONSENSE APPROACH TO INSTRUCTIONAL DESIGN. Alexandria, Virginia: ASTD Press Editorial (Pag. 24-28)
- IEEE. (1993). Standars Collections :Software Engineering.
- JIMÉNEZ ROJAS, J. R., & SOTO ASTORGA, R. D. (Septiembre de 2009). SEGURIDAD UNAM. Consultado el 4 de Noviembre de 2013. Disponible en <http://www.seguridad.unam.mx/descarga.dsc?arch=1736>
- LABRADOR, R. M. (SEPTIEMBRE de 2005). Consultado el 24 de Noviembre de 2013. Disponible en <http://www.informatica.us.es/~ramon/articulos/LicenciasSoftware.pdf>
- LÓPEZ GUZMÁN, C., & GARCÍA PEÑALVO, J. (10 de Noviembre de 2004). REVISTA DIGITAL UNIVERSITARIA. Consultado el 19 de Diciembre de 2013. Disponible en la reutilización de recursos educativos en la Educación Superior apoyada por E-learning: http://www.revista.unam.mx/vol.5/num10/art64/nov_art64.pdf
- MARIBE, R. (2009). INSTRUCTIONAL DESIGN: THE ADDIE APPROACH. New York: Springer (Pag.1-5).
- MARTÍNEZ, D. A. (2002). BLENDED LEARNING: MODELO VIRTUAL PRESENCIAL DE APRENDIZAJE Y SU APLICACIÓN EN ENTORNOS EDUCATIVOS. Consultado el 19 de Diciembre de 2013. Disponible en DGDE: http://www.dgde.ua.es/congresotic/public_doc/pdf/31972.pdf
- MEIER, R. (2009). Professional ANDROID 2 APPLICATION DEVELOPMENT. Canada: Wiley Publish.

- MICROSOFT. (2015). WHAT'S A UNIVESAL WINDOWS APP? Consultado el 7 de Mayo de 2014. Disponible en:
<https://msdn.microsoft.com/library/windows/apps/dn726767.aspx>
- MONTERREY, I. T. (2000). EL APRENDIZAJE BASADO EN PROBLEMAS COMO TÉCNICA DIDACTICA. Consultado el 18 de Febrero de 2014. Disponible en
<http://www.ub.edu/mercanti/ap.pdf>
- NUMBER OF APPS AVAILABLE IN LEADING APP STORES AS 2015. Consultado el 18 de Febrero de 2014. Disponible en:
<http://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/>
- PALACIOS, ALMA L. (2008). TESIS: UN CONJUNTO DE OBJETOS DE APRENDIZAJE PARA APOYAR LA ENSEÑANZA DE LA PROGRAMACIÓN. Consultado el 12 Mayo de 2013.
- PINEDA, D. P. (2012). APRENDE EN LINEA. Consultado el 24 de Noviembre de 2013. Disponible en:
http://aprendeonline.udea.edu.co/banco/html/ambiente_virtual_de_aprendizaje/
- PIXÉLES. (NOVIEMBRE de 2012). ANDROID SE INFECTA CON TROYANO. EL UNIVERSAL, pág. T6.
- PRESSMAN, R. S. (2006). INGENIERÍA DEL SOFTWARE UN ENFOQUE PRÁCTICO. McGraw-Hill.
- PULIDO, P. O. (2002). TESIS: LA ENSEÑANZA DEL ÁLGEBRA LINEAL MEDIANTE SISTEMAS DE INFORMÁTICOS DE CÁLCULO ALGEBRAICO. Consultado el 25 de Febrero de 2014.

- QUILLET, D. E. (1985). DICCIONARIO ENCICLOPÉDICO QUILLET (Vol.7). Editorial Cumbre S.A.
- QUILLET. (1988). DICCIONARIO ENCICLOPÉDICO QUILLET (Vol. I). México: Editorial Cumbre, S.A.
- QUILLET. (1988). DICCIONARIO ENCICLOPÉDICO QUILLET (Vol. TOMO V). México: Editorial Cumbre S.A.
- RAMOS, J. L. (ENERO de 2012). INNOVAN EN APPS PARA LA EDUCACIÓN. EL UNIVERSAL (34,418), pág. T2.
- REYES, O. C., & MORALES VELÁZQUEZ, M. P. (s.f.). Consultado el 17 de Octubre de 2013. Disponible en <http://www.uv.mx/veracruz/usbi/alci12/media/docs/ponencias/pon58.pdf>
- RODRÍGUEZ, R. C. (2009). TESIS: DESARROLLO DE UN TALLER PARA CAPACITAR A LOS PROFESORES DE PRIMER SEMESTRE QUE IMPARTEN LA ASIGNATURA DE ÁLGEBRA DE LA FACULTAD DE INGENIERÍA DE LA UNAM, PARA DISEÑAR OBJETOS DE APRENDIZAJE EN PARTICULAR DEL TEMA SISTEMAS DE ECUACIONES. Consultado el 2 de Marzo de 2014.
- ROSARIO, J. (2005). CIBERSOCIEDAD. Consultado el 17 de Noviembre de 2013. Disponible en: <http://www.cibersociedad.net/archivo/articulo.php?art=218>
- SALINAS, A. M. (2009). TESIS: REDES INALÁMBRICAS COMO ELEMENTO INDISPENSABLE EN LA COMUNICACIÓN TELEFÓNICA CELULAR. Consultado el 2 de Marzo de 2014. Disponible en <http://132.248.9.195/ptb2011/febrero/0666536/Index.html>

- SCHACH, S. R. (2006). INGENIERÍA DE SOFTWARE CLÁSICA Y ORIENTADA A OBJETOS. McGraw HILL.
- SGUERRA, M. I. (2009). GNU/LINUX Y EL SOFTWARE LIBRE Y SUS MÚLTIPLES APLICACIONES. México: ALFAOMEGA. Consultado el 31 de Octubre de 2013. Disponible en http://www.iprsonline.org/unctadictsd/docs/CS_Story.pdf
- SOTELO, GONZÁLEZ JOAQUÍN (2008). DEL E-LEARNING AL M-LEARNING. UNA ACADEMIA EN CADA IPHONE. Consultado el 4 Marzo 2014. Disponible en http://sociedadinformacion.fundacion.telefonica.com/seccion=1268&idioma=es_ES&id=2009102909230001&activo=6.do.
- TIMAURE, E. A. (1998). NEUTRON. Consultado el 7 de Diciembre de 2013. Disponible en <http://neutron.ing.ucv.ve/revista-e/No8/EATimaure%5CWap.htm>
- TISAL, J. (2000). LA RED GSM. PARANINFO THOMSON LEARNING. UNAMCERT, S. (1 de Enero de 2007). Consultado el 6 de Noviembre de 2013. Disponible en Revista seguridad UNAM: <http://revista.seguridad.unam.mx/numero-01/códigos-maliciosos-en-dispositivos-móviles>
- UNESCO. (2004). LAS TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN EN LA FORMACIÓN DOCENTE. Consultado el 25 de Febrero de 2014. Disponible en UNESCO: <http://unesdoc.unesco.org/images/0012/001295/129533s.pdf>
- UNIÓN, CÁMARA DE DIPUTADOS DEL H. CONGRESO DE LA. (ENERO de 2012). LEY FEDERAL DEL DERECHO DE AUTOR. Consultado el 24 de Noviembre de 2014. Disponible en <http://www.diputados.gob.mx/LeyesBiblio/pdf/122.pdf>
- WEITZENFELD, A. (2005). INGENIERÍA DE SOFTWARE ORIENTADA A OBJETOS CON UML, JAVA E INTERNET. Editorial: THOMSON.

- WHITEHEAD, P. (2004). APRENDA VISUALMENTE REDES. ST Editorial.
- WILEY, D. A. (2000). CONNECTING LEARNING OBJECTS TO INSTRUCTIONAL DESIGN THEORY: A DEFINITION, A METAPHOR, AND A TAXONOMY.
Consultado el 1 de Marzo de 2014. Disponible en The Instructional Use of Learning Objects: <http://reusability.org/read/chapters/wiley.doc>

Anexo A – Reactivos de Examen Colegiado

Un examen colegiado es un examen que es construido e instrumentado por un grupo de profesores de la misma área de conocimiento. Esto propicia el establecimiento de estándares mínimos de calidad en el aprendizaje y de alguna manera puede reorientar la actividad de los profesores hacia las competencias de aprendizaje, resolución de problemas, desarrollo de habilidades del pensamiento, de cálculo y de comprensión de conceptos.

En la Tabla 4 se muestran los reactivos con los que cuenta el examen colegiado de Programación del ciclo 2010-2; el eje curricular con las unidades a evaluar, el promedio de cada grupo de Programación por reactivo, el promedio de todos los grupos de la FCQI por reactivo y el promedio de todos los grupos de la UABC a nivel estado que llevan Programación por reactivo.

Tabla 4 - Tabla de resultados y reactivos de examen colegiado de Programación del ciclo 2012-2

PROGRAMACION												
Reactivo	No. Unidad	Unidad	21	22	23	24	25	26	27	28	FCQI	UABC
1	1	Metodología para la solución de problemas	0.76	0.80	1.00	0.84	0.77	0.81	0.82	0.86	0.82	0.89
2	1	Metodología para la solución de problemas	0.55	0.37	0.44	0.42	0.30	0.43	0.36	0.50	0.41	0.45
3	1	Metodología para la solución de problemas	0.85	0.91	0.84	0.63	0.68	0.81	0.64	0.79	0.78	0.83
4	2	Introducción al lenguaje de programación C	0.64	0.71	0.56	0.58	0.48	0.62	0.64	0.43	0.59	0.59
5	2	Introducción al lenguaje de programación C	0.52	0.43	0.64	0.84	0.45	0.48	0.91	0.79	0.56	0.58
6	2	Introducción al lenguaje de programación C	0.39	0.49	0.48	0.58	0.39	0.71	0.27	0.71	0.51	0.48
7	2	Introducción al lenguaje de programación C	0.48	0.83	0.84	0.37	0.45	0.55	0.45	0.79	0.59	0.68
8	2	Introducción al lenguaje de programación C	0.52	0.49	0.64	0.37	0.34	0.83	0.36	0.86	0.55	0.66
9	2	Introducción al lenguaje de programación C	0.24	0.06	0.24	0.42	0.14	0.12	0.18	0.14	0.17	0.16
10	2	Introducción al lenguaje de programación C	0.73	0.83	0.84	0.68	0.89	0.88	0.64	0.71	0.81	0.86
11	2	Introducción al lenguaje de programación C	0.42	0.49	0.60	0.42	0.32	0.43	0.36	0.64	0.44	0.52
12	2	Introducción al lenguaje de programación C	0.79	0.74	0.60	0.63	0.52	0.74	0.55	0.50	0.65	0.67
13	2	Introducción al lenguaje de programación C	0.52	0.54	0.44	0.74	0.41	0.62	0.36	0.71	0.53	0.55
14	2	Introducción al lenguaje de programación C	0.36	0.46	0.56	0.58	0.25	0.50	0.73	0.64	0.46	0.55
15	2	Introducción al lenguaje de programación C	0.45	0.51	0.76	0.68	0.50	0.88	0.55	0.71	0.63	0.61
16	2	Introducción al lenguaje de programación C	0.45	0.69	0.72	0.53	0.48	0.60	0.64	0.71	0.58	0.51
17	3	Funciones	0.42	0.46	0.48	0.16	0.30	0.24	0.55	0.29	0.35	0.46
18	3	Funciones	0.76	0.74	0.64	0.63	0.45	0.57	0.55	0.43	0.61	0.66
19	3	Funciones	0.67	0.83	0.56	0.26	0.34	0.40	0.73	0.57	0.53	0.57
20	3	Funciones	0.36	0.31	0.28	0.53	0.36	0.38	0.36	0.50	0.37	0.46
21	3	Funciones	0.61	0.57	0.72	0.89	0.50	0.83	0.64	0.64	0.66	0.71
22	4	Estructuras de control de selección	0.64	0.69	0.60	0.63	0.41	0.62	0.45	0.43	0.57	0.63
23	4	Estructuras de control de selección	0.70	0.83	0.72	0.74	0.57	0.67	0.91	0.71	0.70	0.75
24	4	Estructuras de control de selección	0.55	0.54	0.68	0.47	0.32	0.69	0.55	0.57	0.54	0.60
25	4	Estructuras de control de selección	0.36	0.60	0.64	0.63	0.27	0.64	0.45	0.43	0.50	0.58
26	4	Estructura de control de iteración	0.12	0.17	0.32	0.05	0.27	0.14	0.18	0.36	0.20	0.23
27	4	Estructura de control de iteración	0.30	0.31	0.20	0.32	0.30	0.17	0.27	0.36	0.27	0.29
28	4	Estructura de control de iteración	0.21	0.17	0.32	0.32	0.14	0.14	0.18	0.21	0.20	0.26
29	4	Estructura de control de iteración	0.24	0.49	0.24	0.26	0.25	0.26	0.27	0.36	0.30	0.36
30	4	Estructura de control de iteración	0.09	0.40	0.32	0.53	0.14	0.43	0.09	0.29	0.29	0.31
31	5	Arreglos	0.45	0.34	0.28	0.53	0.20	0.40	0.82	0.64	0.39	0.44
32	5	Arreglos	0.36	0.34	0.44	0.32	0.16	0.17	0.09	0.43	0.28	0.36
33	5	Arreglos	0.09	0.06	0.12	0.16	0.16	0.24	0.09	0.21	0.14	0.19
34	5	Arreglos	0.03	0.37	0.16	0.00	0.11	0.29	0.27	0.29	0.19	0.39
35	5	Arreglos	0.33	0.29	0.28	0.26	0.27	0.24	0.45	0.43	0.30	0.37
36	5	Arreglos	0.18	0.29	0.16	0.21	0.25	0.17	0.45	0.29	0.23	0.28
37	5	Arreglos	0.48	0.63	0.28	0.53	0.34	0.10	0.55	0.29	0.38	0.35
38	5	Arreglos	0.36	0.31	0.36	0.16	0.16	0.71	0.27	0.50	0.37	0.48
39	5	Arreglos	0.18	0.20	0.36	0.32	0.36	0.43	0.27	0.43	0.32	0.29
40	5	Arreglos	0.15	0.20	0.28	0.42	0.18	0.50	0.45	0.14	0.28	0.30

La mayoría de los grupos salieron muy bien en la evaluación de los 3 primeros reactivos que pertenecen a la Unidad 1.”Metodología para la solución de problemas”.

En la Unidad 2. “Introducción al Lenguaje de Programación C”, la mayoría de los grupos a los que se les aplico el examen salieron bajos en 3 de 13 reactivos:

- Reactivo #7 - Estructura Básica de un programa (ver Figura 46 para más detalles).

1. DATOS DE IDENTIFICACIÓN DEL CONTENIDO A EVALUAR			
1.1 REACTIVO (S):		7	
1.2 CURSO: Programación		1.3 UNIDAD: 2. Introducción al lenguaje de programación C	
1.4 TEMA: 2.2 Estructura básica de un programa		1.5 SUBTEMA: 2.2.5 La función principal main()	
2. COMENTARIO ACLARATORIO ACERCA DEL SENTIDO DEL CONTENIDO			
La función main() es obligatoria en todo programa de C, por más sencillo o complejo que sea un programa esta función nunca debe omitirse, ya que indica el comienzo del programa y es la primer función que será llamada al ejecutarse el programa. La función main requiere los paréntesis () después de la palabra main. El alumno deberá ser capaz de identificar el uso correcto de la función main().			
2.1 COMPETENCIA		Aplicar la metodología de programación utilizando los fundamentos del lenguaje C, para elaborar programas secuenciales básicos que solucionen problemas reales en el área de ingeniería, con organización y creatividad.	
2.2 INDICADOR		Aplicar la función main() en una estructura básica de programa.	
2.3 TIPO DE CONTENIDO		CONCEPTO (X)	
2.4 DIFICULTAD		PROCEDIMIENTO ()	
REPRODUCCIÓN (X)		CONEXIÓN ()	
REFLEXIÓN ()			
3.ATRIBUTOS RELEVANTES DE LOS ESTÍMULOS QUE SE PRESENTARÁN A LOS ESTUDIANTES			
3.1 INSTRUCCIONES PARA RESPONDER EL REACTIVO			
Seleccionar el programa que utiliza la función main de manera correcta.			
3.2 BASE DEL REACTIVO			
Se proporcionará al examinado un programa con la estructura básica. Se le pedirá seleccionar entre cuatro opciones aquel programa que al ejecutarse llame a la función main.			
3.3 VOCABULARIO E INFORMACIÓN TEXTUAL, GRÁFICA O TABULAR A EMPLEAR			
La información que se proporcionará al examinado serán cuatro ejemplos de un programa con la estructura básica haciendo énfasis en la función main().			
3.4 DISTRACTORES			
Los distractores serán diferentes formas de llamar la función main() dentro de la estructura del programa.			
3.5 RESPUESTA CORRECTA			
Será aquella cuya selección corresponda a un programa con la estructura correcta, de manera que al ejecutarse el programa la función main() sea llamada.			
4 REACTIVO MUESTRA			
Seleccione el programa en el cual la función principal es llamada correctamente.			
a) #include<stdio.h> void main(){ printf("Buen día"); }			
b) #include<stdio.h> void Main(){ printf("Buen día"); }			
c) #include<stdio.h> void main{ printf("Buen día");			

<pre> } d) #include<stdio.h> void Main{ printf("Buen día"); } </pre>
4.1 TIEMPO ESTIMADO DE EJECUCIÓN 1 minuto
4.2 CONGRUENCIA COMPETENCIA DEL ÍTEM – COMPETENCIA DE LA UNIDAD O DEL CURSO El alumno debe conocer y entender la importancia del llamado de la función principal y de que manera influye para que cualquier programa que haga pueda ser ejecutado.

Figura 46 - Formato para elaborar especificaciones de reactivos – Reactivo #7

- Reactivos #14 y 15 - Expresiones básicas (ver Figura 47 para más detalles)

1. DATOS DE IDENTIFICACIÓN DEL CONTENIDO A EVALUAR			
1.1 REACTIVO (S):		Items 14 y 15	
1.2 CURSO: Programación		1.3 UNIDAD: II Introducción al lenguaje de programación c	
1.4 TEMA: 2.5 Expresiones Básicas		1.5 SUBTEMA: 2.5.3 Funciones de Entrada/Salida (scanf(), printf())	
2. COMENTARIO ACLARATORIO ACERCA DEL SENTIDO DEL CONTENIDO Las funciones printf(); y Scanf(); son importantes ya que permiten la entrada de datos y salida de información. La especificación atenderá el uso de las funciones de entrada y salida, considerando especificadores de formato y secuencias de escape.			
2.1 COMPETENCIA		Aplicar la metodología de programación utilizando los fundamentos del lenguaje C, para elaborar programas secuenciales básicos que solucionen problemas reales en el área de ingeniería, con organización y creatividad.	
2.2 INDICADOR		Utilizar la sintaxis de código de acuerdo a los fundamentos de la programación para la entrada y salida de información	
2.3 TIPO DE CONTENIDO		CONCEPTO () PROCEDIMIENTO (x)	
2.4 DIFICULTAD		REPRODUCCIÓN () CONEXIÓN () REFLEXIÓN (x)	
3. ATRIBUTOS RELEVANTES DE LOS ESTÍMULOS QUE SE PRESENTARÁN A LOS ESTUDIANTES			
3.1 INSTRUCCIONES PARA RESPONDER EL REACTIVO Escoger la sintaxis correcta del código			
3.2 BASE DEL REACTIVO Se proporcionará a los examinados instrucciones de captura de datos y la salida de información, el examinado lo analizará y seleccionará la sintaxis correcta para la captura y salida de información.			
3.3 VOCABULARIO E INFORMACIÓN TEXTUAL, GRÁFICA O TABULAR A EMPLEAR: La información que se le proporcionará al examinado es un texto de lo que debería hacer un código en c que contenga funciones de entrada y salida de datos (printf(); y scanf();).			
3.4 DISTRACTORES Serán instrucciones que no satisfagan los requerimientos del reactivo tales como errores de sintaxis.			
3.5 RESPUESTA CORRECTA Serán instrucciones que satisfagan los requerimientos del reactivo			
4 REACTIVO MUESTRA Item 14 Identifica cual es la sintaxis correcta para mostrar en pantalla 3.1416 a){float n=3.14159265;printf("%1.4f\n",n);} b){float n=3.14159265;printf("%4.1f\n",&n);} c){float n=3.14159265;printf("%1.4f\n",&n);} d){float n=3.14159265;printf("%4.2f\n",n);} Item 15 Identifica cual es la sintaxis correcta para capturar el siguiente numero 3.1416 {float n; scanf("%f",&n);} {float n; scanf("%s",n);} {float n; scanf("%i",n);} {float n; scanf("%l",&n);}			
4.1 TIEMPO ESTIMADO DE EJECUCIÓN 1 minuto			
4.2 CONGRUENCIA COMPETENCIA DEL ÍTEM – COMPETENCIA DE LA UNIDAD O DEL CURSO: Aplicar la metodología de programación utilizando los fundamentos del lenguaje C, para elaborar programas secuenciales básicos que solucionen problemas reales en el área de ingeniería, con organización y creatividad.			

Figura 47 - Formato para elaborar especificaciones de reactivos – Reactivos #14 y 15

En la Unidad 3. “Funciones”, la mayoría de los grupos de Programación salieron bajos en 2 de 5 reactivos:

- Reactivo #20 – Funciones (ver Figura 48 para más detalles)

1. DATOS DE IDENTIFICACIÓN DEL CONTENIDO A EVALUAR			
1.1 REACTIVO (S):		20	
1.2 CURSO: Programación		1.3 UNIDAD: 3	
1.4 TEMA: 3.1 Función		1.5 SUBTEMA: 3.1.3 Llamada a Función	
2. COMENTARIO ACLARATORIO ACERCA DEL SENTIDO DEL CONTENIDO			
En este contenido el alumno aprenderá a llamar a la función dentro de otra función principal. Así como es importante a la hora de declarar la función, hacerlo de manera correcta con todos los parámetros completos, a la hora de llamarla se debe cuidar la estructura para su correcto funcionamiento.			
2.1 COMPETENCIA	Diseñar y codificar funciones para la optimización del código generado en la solución de problemas de procesamiento de información aplicando la metodología de programación, con actitud propositiva y organizada.		
2.2 INDICADOR	Identificar el correcto llamado de una función dentro de un programa.		
2.3 TIPO DE CONTENIDO	CONCEPTO ()		PROCEDIMIENTO (x)
2.4 DIFICULTAD	REPRODUCCIÓN ()	CONEXIÓN ()	REFLEXIÓN (x)
3. ATRIBUTOS RELEVANTES DE LOS ESTÍMULOS QUE SE PRESENTARÁN			
3.1 INSTRUCCIONES PARA RESPONDER EL REACTIVO			
Escoger la opción que identifique un llamado de función correctamente estructurada.			
3.2 BASE DEL REACTIVO: Se proporcionará al examinado cuatro opciones de llamado de una función, de los cuales solo un tendrá las características completas y correctas.			
3.3 VOCABULARIO E INFORMACIÓN TEXTUAL, GRÁFICA O TABULAR A EMPLEAR: Se le proporcionará al examinado cuatro ejemplos de maneras de llamar a una función			
3.4 DISTRACTORES: 3 Maneras incorrectas o faltantes de parámetros para llamar a una función.			
3.5 RESPUESTA CORRECTA			
Será la que tenga correctamente el llamado a la función cumpliendo con todos los requerimientos del mismo.			
4 REACTIVO MUESTRA			
Considerando la siguiente función			
<pre>int cubo (int base) { int potencia; potencia = base * base * base; return potencia; }</pre>			
¿Cuál de las siguientes opciones, indica la manera correcta de llamar a la función?			
a) int numero; printf("El cubo del número %d es %f\n", numero, potencia(numero,3));			
b) int numero; printf("El cubo del número %d es %d\n", numero, (numero*numero*numero));			
c) int numero; printf("El cubo del número %d es %f\n", numero, cubo(numero*numero*numero));			
d) int numero; printf("El cubo del número %d es %d\n", numero, cubo(numero));			
4.1 TIEMPO ESTIMADO DE EJECUCIÓN: 3 minutos			
4.2 CONGRUENCIA COMPETENCIA DEL ÍTEM – COMPETENCIA DE LA UNIDAD O DEL CURSO: Dentro de este bloque 3 ya entendimos lo que es una función ya la estructuramos y definimos así que ahora nos preparamos para hacer uso de ella hasta lograr la aplicación y uso de funciones.			

Figura 48 - Formato para elaborar especificaciones de reactivos – Reactivo #20

- Reactivo #21 – Funciones (ver Figura 49 para más detalles)

1. DATOS DE IDENTIFICACIÓN DEL CONTENIDO A EVALUAR			
1.1 REACTIVO (S):		21	
1.2 CURSO: Programación		1.3 UNIDAD: 3	
1.4 TEMA: 3.1 Función		1.5 SUBTEMA: 3.1.4 Valor de retorno	
2. COMENTARIO ACLARATORIO ACERCA DEL SENTIDO DEL CONTENIDO			
Después de declarar una función y llamarla, necesitamos aprender a obtener el valor de retorno de la operación de la función así como reconocer el tipo de respuesta que nos mandará para ver que esté haciendo bien el ciclo de entrada-proceso-salida.			
2.1 COMPETENCIA		Diseñar y codificar funciones para la optimización del código generado en la solución de problemas de procesamiento de información aplicando la metodología de programación, con actitud propositiva y organizada.	
2.2 INDICADOR		Evaluar el resultado del proceso de una función y manipule correctamente los valores de retorno.	
2.3 TIPO DE CONTENIDO		CONCEPTO ()	PROCEDIMIENTO (x)
2.4 DIFICULTAD	REPRODUCCIÓN ()	CONEXIÓN ()	REFLEXIÓN (x)
3. ATRIBUTOS RELEVANTES DE LOS ESTÍMULOS QUE SE PRESENTARÁN A LOS ESTUDIANTES			
3.1 INSTRUCCIONES PARA RESPONDER EL REACTIVO			
Seleccionar el o los valores de retorno correctos según la función presentada.			
3.2 BASE DEL REACTIVO			
Se proporcionará al examinado cuatro posibles valores de retorno de una función			
3.3 VOCABULARIO E INFORMACIÓN TEXTUAL, GRÁFICA O TABULAR A EMPLEAR:			
Se le proporcionará al examinado el código correcto de un programa y se le solicitará que seleccione de cuatro opciones posibles la que represente correctamente la salida después de ejecutarse la función.			
3.4 DISTRACTORES			
Los distractores podrán ser salidas parecidas pero no completamente correctas.			
3.5 RESPUESTA CORRECTA			
Será aquella que sea realmente el resultado después de correr la función.			
4 REACTIVO MUESTRA			
Dado el siguiente bloque de código, ¿cuál será la salida del programa?			
<pre>int cubo(int base), main() { int numero; for(numero=1; numero<=5; numero++) { printf("El cubo del número %d es %d\n", numero, cubo(numero)); } return 0; } int cubo(int base) { int potencia; potencia = base * base * base; return potencia; }</pre>			
a) 0			

<p>b) El cubo del número 1 es 1 El cubo del número 2 es 8 El cubo del número 3 es 27 El cubo del número 4 es 64 El cubo del número 5 es 125</p> <p>c) El cubo del número 5 es 125</p> <p>d) El cubo del número es potencia</p>
<p>4.1 TIEMPO ESTIMADO DE EJECUCIÓN 2 minutos</p>
<p>4.2 CONGRUENCIA COMPETENCIA DEL ÍTEM – COMPETENCIA DE LA UNIDAD O DEL CURSO</p> <p>Como se mencionó en otro de las especificaciones de reactivos de esta unidad 3, las competencias son “Diseñar y codificar funciones para la optimización del código generado en la solución de problemas”, así que en este apartado nos aseguramos que el alumno después que codificó, mando llamar a la función, manipule correctamente los valores de retorno para la correcta solución del problema planteado.</p>

Figura 49 - Formato para elaborar especificaciones de reactivos – Reactivo #21

En la Unidad 4. “Estructuras de Control de Iteración”, la mayoría de los grupos de Programación salieron bajos en 3 de 5 reactivos:

- Reactivo # 26 - Teoría de ciclos (ver Figura 50 para más detalles)

1. DATOS DE IDENTIFICACIÓN DEL CONTENIDO A EVALUAR				
1.1 REACTIVO (S):		Ítem 26		
1.2 CURSO: Programación		1.3 UNIDAD: V Estructuras de control e iteración		
1.4 TEMA: 5.1 Teoría de ciclos		1.5 SUBTEMA: 5.1.2 Contadores		
2. COMENTARIO ACLARATORIO ACERCA DEL SENTIDO DEL CONTENIDO				
Los contadores se utilizan con la finalidad de contar sucesos o acciones internas de un ciclo; deben realizar una operación de inicialización y posteriormente las sucesivas de incremento o decremento del mismo.				
2.1 COMPETENCIA		Elaborar programas aplicando la metodología de programación y teoría de arreglos para simplificación de manejo de datos, de manera eficiente y organizada		
2.2 INDICADOR		Utilizar contadores en un segmento de código		
2.3 TIPO DE CONTENIDO		CONCEPTO ()		PROCEDIMIENTO (x)
2.4 DIFICULTAD	REPRODUCCIÓN (x)	CONEXIÓN ()		REFLEXIÓN ()
3.ATRIBUTOS RELEVANTES DE LOS ESTÍMULOS QUE SE PRESENTARÁN A LOS ESTUDIANTES				
3.1 INSTRUCCIONES PARA RESPONDER EL REACTIVO				
Escoger la respuesta adecuada a la pregunta propuesta.				
3.2 BASE DEL REACTIVO				
Se le mostrara al examinado un código de programación donde deberá ser capaz de identificar la respuesta correcta a la pregunta propuesta.				
3.3 VOCABULARIO E INFORMACIÓN TEXTUAL, GRÁFICA O TABULAR A EMPLEAR:				
La información que se proporcione al examinado será un fragmento de código.				
3.4 DISTRACTORES				
Los distractores serán posibles respuestas erróneas, que el alumno deberá identificar				
3.5 RESPUESTA CORRECTA				
Sera aquella cuya información sea la correcta.				
4 REACTIVO MUESTRA				

<pre> int np=0, ni=0, tn=0,n; do{ printf("Teclee un numero"); scanf("%d",&n); s=s+n; if(n%2) ni=ni+1; else np=np+1; tn++; }while(n != 0); </pre> <p>¿En el código anterior se utilizan contadores, cuantos son y en que línea se encuentran?</p> <p>A) Son 3 y están en la línea 7, 9 y 10</p> <p>B) Son 3 y están en la línea 5, 7 y 9</p> <p>C) Es 1 y se encuentra en la línea 5</p> <p>D) Son 2 y están en la línea 7 y 9</p>
<p>4.1 TIEMPO ESTIMADO DE EJECUCIÓN 2 minutos</p>
<p>4.2 CONGRUENCIA COMPETENCIA DEL ÍTEM – COMPETENCIA DE LA UNIDAD O DEL CURSO</p> <p>Elaborar programas aplicando la metodología de programación y la teoría de ciclos para dar solución a problemas que requieran procesos repetitivos, con actitud positiva, de manera eficiente y organizada.</p>

Figura 50 - Formato para elaborar especificaciones de reactivos – Reactivo #26

- Reactivo # 28 - Tipos de Ciclos (ver Figura 51 para más detalles)

1. DATOS DE IDENTIFICACIÓN DEL CONTENIDO A EVALUAR			
1.1 REACTIVO (S):		Ítem 28	
1.2 CURSO: Programación		1.3 UNIDAD: V Estructuras de control de iteración	
1.4 TEMA: 5.2 Tipos de ciclos		1.5 SUBTEMA: 5.2.1 Ciclos controlados por contador	
2. COMENTARIO ACLARATORIO ACERCA DEL SENTIDO DEL CONTENIDO			
La aplicación de un ciclo controlado por contador resuelve el problema de repetir ciertos segmentos del programa, realizando un número determinado de iteraciones, lo cual nos permite optimizar el código. El alumno deberá interpretar cuando es conveniente la utilización de un ciclo, tomando en cuenta el número de iteraciones adecuadas.			
2.1 COMPETENCIA		Elaborar programas aplicando la metodología de programación y teoría de arreglos para simplificación de manejo de datos, de manera eficiente y organizada	
2.2 INDICADOR		Aplicar un ciclo controlado por contador en un segmento de código para resolver el enunciado de un problema	
2.3 TIPO DE CONTENIDO		CONCEPTO ()	PROCEDIMIENTO (x)
2.4 DIFICULTAD	REPRODUCCIÓN (x)	CONEXIÓN ()	REFLEXIÓN ()
3. ATRIBUTOS RELEVANTES DE LOS ESTÍMULOS QUE SE PRESENTARÁN A LOS ESTUDIANTES			
3.1 INSTRUCCIONES PARA RESPONDER EL REACTIVO			
Escoger el segmento de código que corresponda a la solución del problema propuesto.			
3.2 BASE DEL REACTIVO			
Se le planteará al examinado un problema donde deberá ser capaz de identificar el código correcto que represente la solución del problema.			
3.3 VOCABULARIO E INFORMACIÓN TEXTUAL, GRÁFICA O TABULAR A EMPLEAR:			
La información que se proporcione al examinado serán fragmentos de código.			
3.4 DISTRACTORES			
Los distractores serán segmentos de código erróneos, los cuales tendrán errores lógicos, que el alumno deberá identificar.			
3.5 RESPUESTA CORRECTA			
Sera aquella cuyo segmento de código represente la solución del problema.			
4 REACTIVO MUESTRA			

Se requiere determinar el promedio de 10 números consecutivos pares, iniciando en el número 12.

¿Cuál de los siguientes códigos representa la solución del problema?

A) `int i, sum=0, prom;
for(i=12; i<= 30; i=i+2)
sum=sum+i;
prom=sum/10;`

B) `int i, sum, prom;
for(i=12; i<= 30; i++)
sum=sum+i;
prom=sum/10;`

C) `int i, sum=0, prom;
for(i=12; i< 30; i=i+2)
sum=sum+n;
prom=sum/10;`

D) `int i, sum=0, prom;
for(i=0; i<= 10; i=i+2)
sum=sum+i;
prom=sum/30;`

4.1 TIEMPO ESTIMADO DE EJECUCIÓN 2 minutos

4.2 CONGRUENCIA COMPETENCIA DEL ÍTEM – COMPETENCIA DE LA UNIDAD O DEL CURSO

Elaborar programas aplicando la metodología de programación y la teoría de ciclos para dar solución a problemas que requieran procesos repetitivos, con actitud positiva, de manera eficiente y organizada.

Figura 51 - Formato para elaborar especificaciones de reactivos – Reactivo #28

- Reactivo # 29 - Tipos de Ciclos (ver Figura 52 para más detalles)

1. DATOS DE IDENTIFICACIÓN DEL CONTENIDO A EVALUAR			
1.1 REACTIVO (S):		Ítem 29	
1.2 CURSO: Programación		1.3 UNIDAD: V Estructuras de control de iteración	
1.4 TEMA: 5.2 Tipos de ciclos		1.5 SUBTEMA: 5.2.2 Ciclos controlados por centinela	
2. COMENTARIO ACLARATORIO ACERCA DEL SENTIDO DEL CONTENIDO			
La aplicación de un ciclo controlado por centinela resuelve el problema de repetir ciertos segmentos del programa, realizando un número indeterminado de iteraciones, lo cual nos permite optimizar el código. El alumno deberá interpretar cuando es conveniente la utilización de un ciclo, tomando en cuenta que no se sabe las veces que se va a repetir el ciclo. Para probar lo anterior, se elaborara un reactivo.			
2.1 COMPETENCIA		Elaborar programas aplicando la metodología de programación y teoría de arreglos para simplificación de manejo de datos, de manera eficiente y organizada	
2.2 INDICADOR		Aplicar un ciclo controlado por centinela en un segmento de código para resolver el enunciado de un problema	
2.3 TIPO DE CONTENIDO		CONCEPTO ()	PROCEDIMIENTO (x)
2.4 DIFICULTAD	REPRODUCCIÓN (x)	CONEXIÓN ()	REFLEXIÓN ()
3. ATRIBUTOS RELEVANTES DE LOS ESTÍMULOS QUE SE PRESENTARÁN A LOS ESTUDIANTES			
3.1 INSTRUCCIONES PARA RESPONDER EL REACTIVO			
Escoger el segmento de código que corresponda a la solución del problema propuesto.			
3.2 BASE DEL REACTIVO			
Se le planteara al examinado un problema donde deberá ser capaz de identificar el código correcto que represente la solución del problema.			
3.3 VOCABULARIO E INFORMACIÓN TEXTUAL, GRÁFICA O TABULAR A EMPLEAR:			
La información que se proporcione al examinado serán fragmentos de código.			
3.4 DISTRACTORES			
Los distractores serán segmentos de código erróneos, los cuales tendrán errores, que el alumno deberá identificar.			
3.5 RESPUESTA CORRECTA			
Sera aquella cuyo segmento de código represente la solución del problema.			
4 REACTIVO MUESTRA			

Se requiere determinar el total de dinero que se retira de un cajero de banco en un determinado día.
¿Cual de los siguientes códigos representa la solución del problema?

- A) `int i, sum=0, ret, op;`
 `do{`
 `printf("teclea cantidad de dinero a retirar");`
 `scanf("%d", &ret);`
 `sum= sum + ret;`
 `printf("¿otro retiro? /n 1)SI /n 0)NO");`
 `scanf("%d", &op);`
 `}while(op==1);`
 `Printf("total retirado del dia=%d",sum);`
- B) `int i, sum=0, ret;`
 `for(i=0; i<100; i++)`
 `{ printf("teclea cantidad de dinero a retirar");`
 `scanf("%d", &ret);`
 `sum= sum + ret;}`
 `Printf("total retirado del dia=%d",sum);`
- C) `int i, sum=0, ret, op;`
 `do{`
 `printf("teclea cantidad de dinero a retirar");`
 `scanf("%d", &ret);`
 `sum= sum + ret;`
 `printf("¿otro retiro? /n 1)SI /n 0)NO");`
 `scanf("%d", &op);`
 `}while(op==0);`
 `Printf("total retirado del dia=%d",sum);`
- D) `int i, sum=0, ret, op;`
 `while(op==0){`
 `printf("teclea cantidad de dinero a retirar");`
 `scanf("%d", &ret);`
 `sum= sum + ret;`
 `printf("¿otro retiro? /n 1)SI /n 0)NO");`

<pre>scanf("%d", &op);} Printf("total retirado del dia=%d",sum);</pre>
4.1 TIEMPO ESTIMADO DE EJECUCIÓN 2 minutos
<p>4.2 CONGRUENCIA COMPETENCIA DEL ÍTEM – COMPETENCIA DE LA UNIDAD O DEL CURSO</p> <p>Elaborar programas aplicando la metodología de programación y la teoría de ciclos para dar solución a problemas que requieran procesos repetitivos, con actitud positiva, de manera eficiente y organizada.</p>

Figura 52 - Formato para elaborar especificaciones de reactivos – Reactivo #29

Anexo B – Diagramas UML

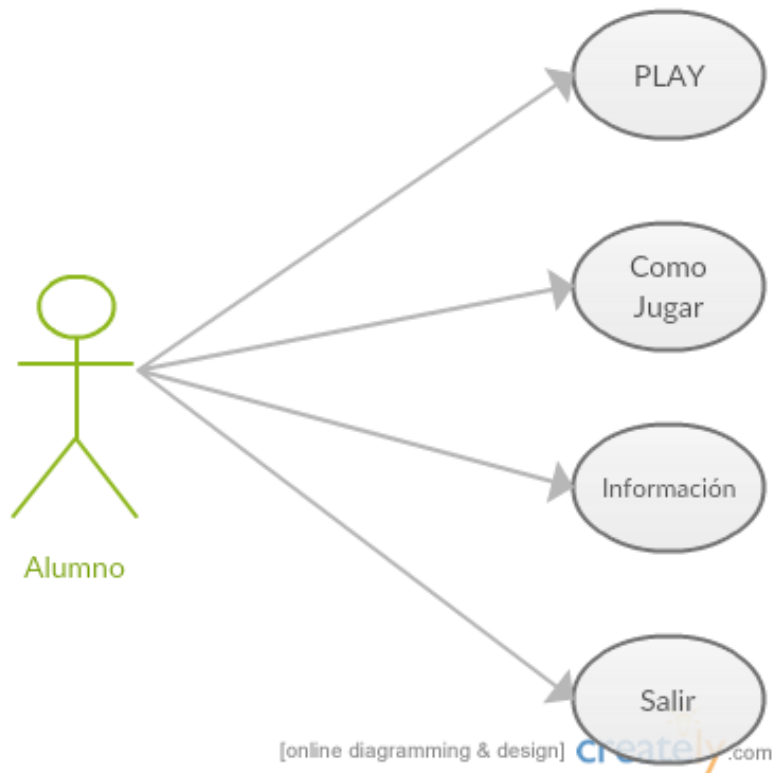


Figura 53 - Diagrama de casos de uso para aplicación móvil El Fox & Robot (Fuente Propia)

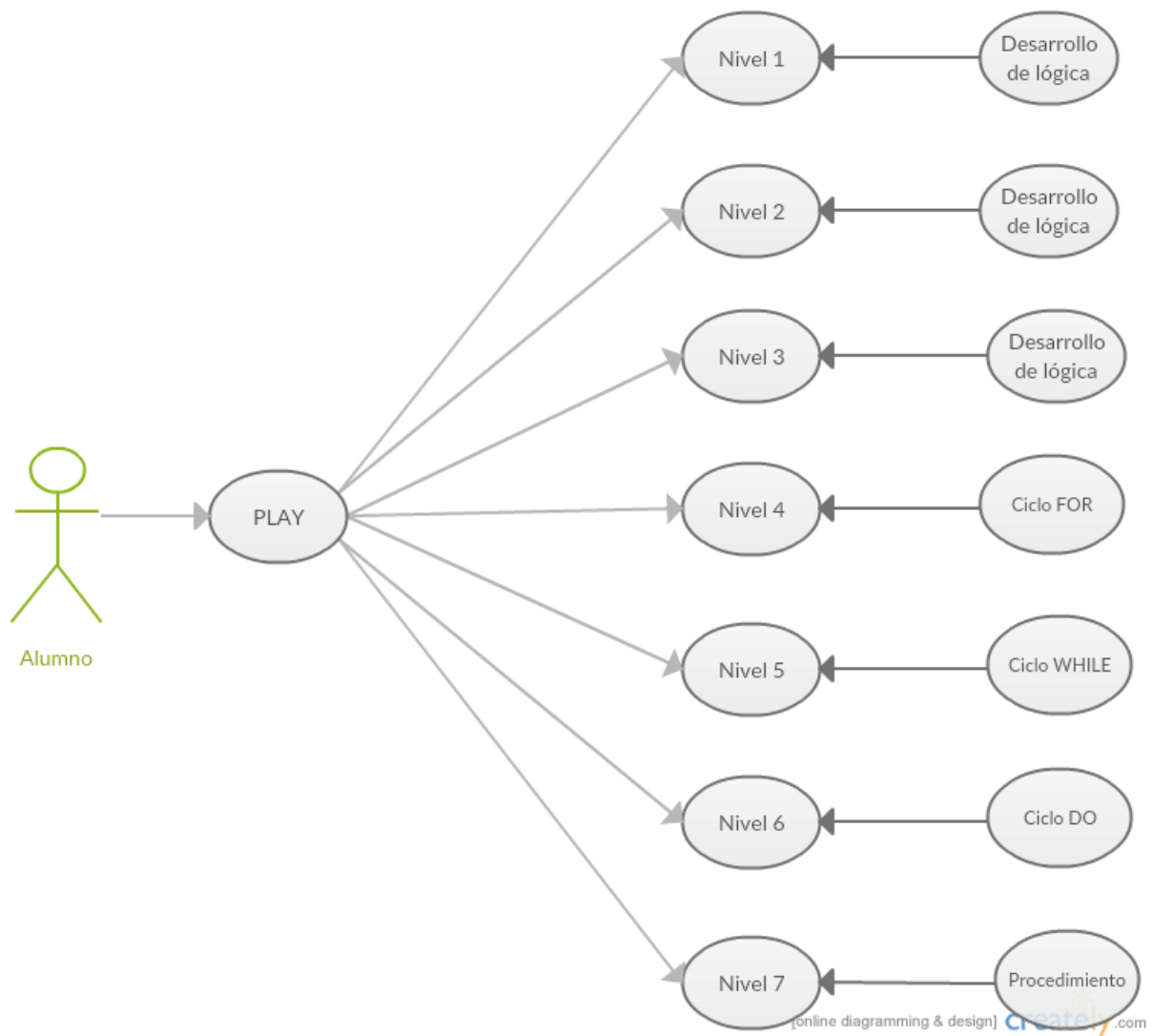


Figura 54 - Diagrama de casos de uso de niveles de la aplicación móvil El Fox & Robot (Fuente Propia)

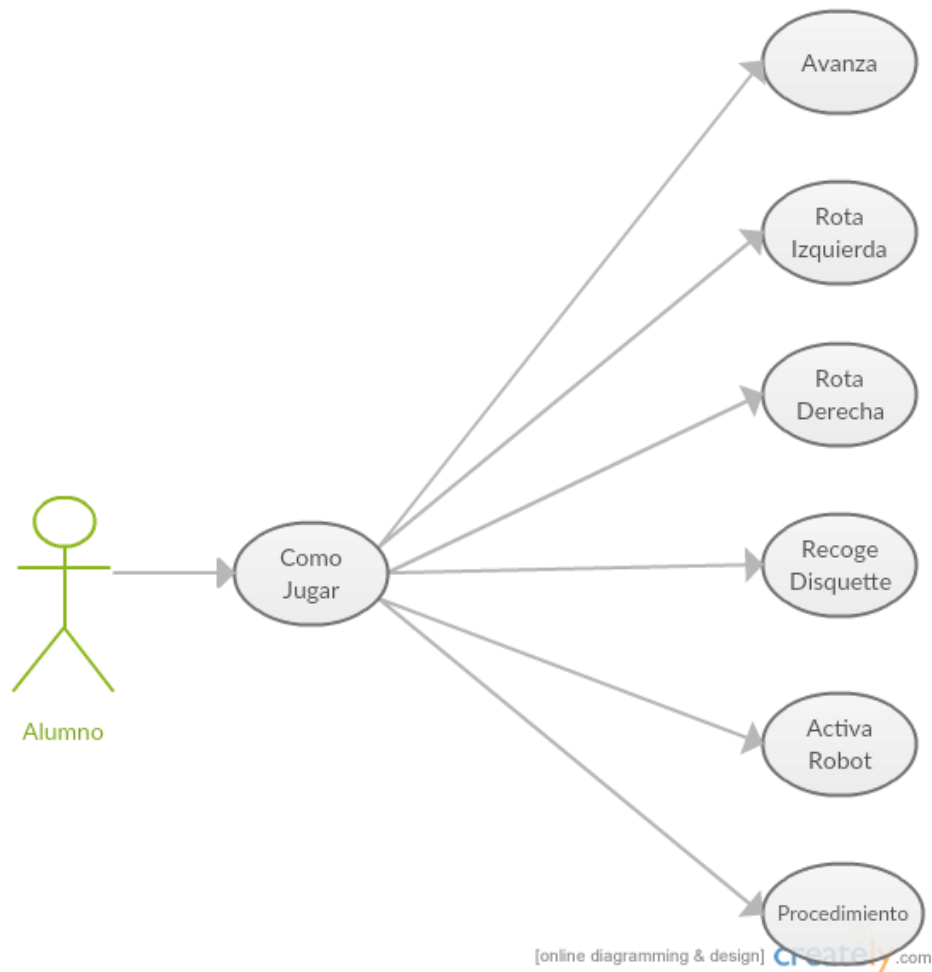


Figura 55 – Diagrama de Casos de uso de como jugar la aplicación móvil El Fox & Robot (Fuente Propia)

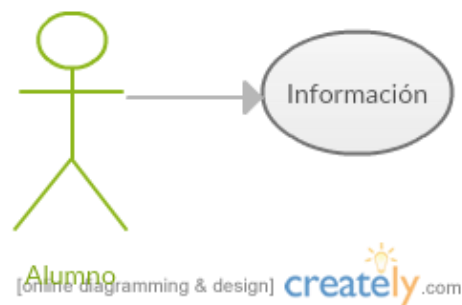


Figura 56 - Diagrama de casos de uso para ver la información de la aplicación móvil El Fox y el Robot (Fuente Propia)

Anexo C – GameMaker: Studio

Como ya se mencionó en el Capítulo 4, para el desarrollo de la aplicación móvil se utilizó la herramienta GameMaker: Studio, que es una herramienta de creación de videojuegos muy completa que permite crear videojuegos de una forma sencilla e intuitiva. La limitación que tiene es que la versión gratuita solo permite la exportación de aplicaciones a Windows y no cuenta con herramientas avanzadas.

A continuación se muestra parte del código de los principales procedimientos de la aplicación móvil El Fox & Robot:

- Objetos declarados en la herramienta GameMaker (ver Figura 57)

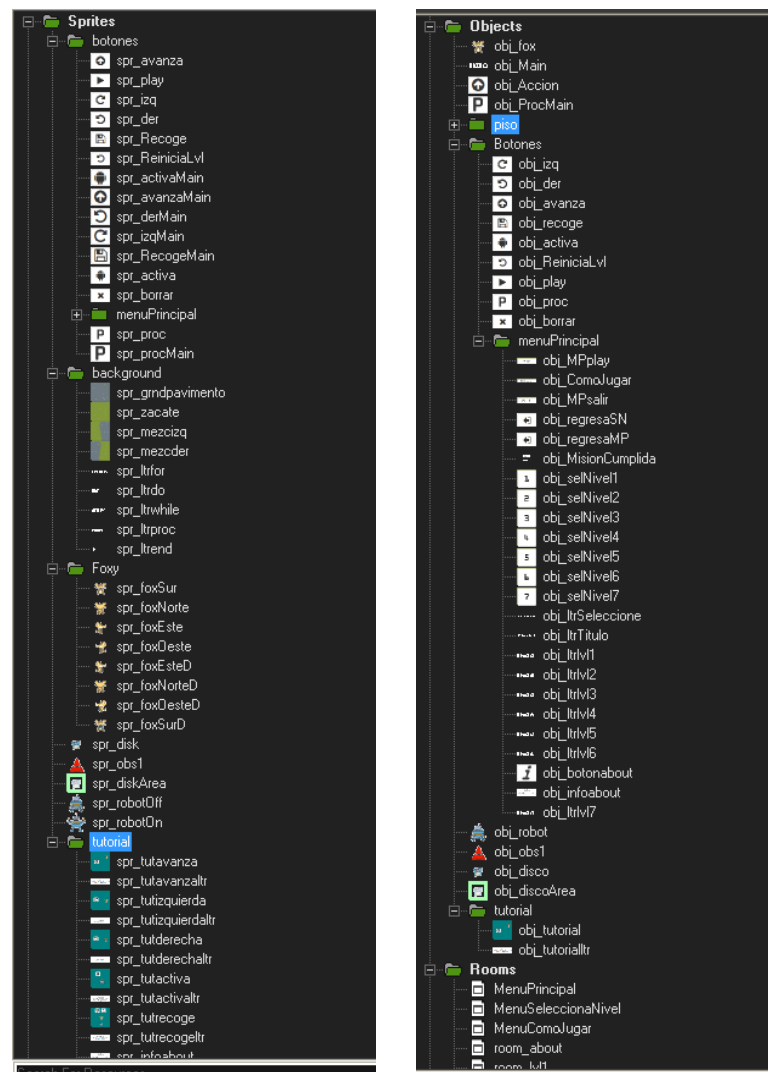


Figura 57 - Sprites y Objetos utilizados en la aplicación móvil El Fox & Robot (Fuente Propia)

- **Objeto Main:** Es la función principal de toda la aplicación, es la que manda llamar todos los eventos programados (ver Figura 58 - Figura 61).

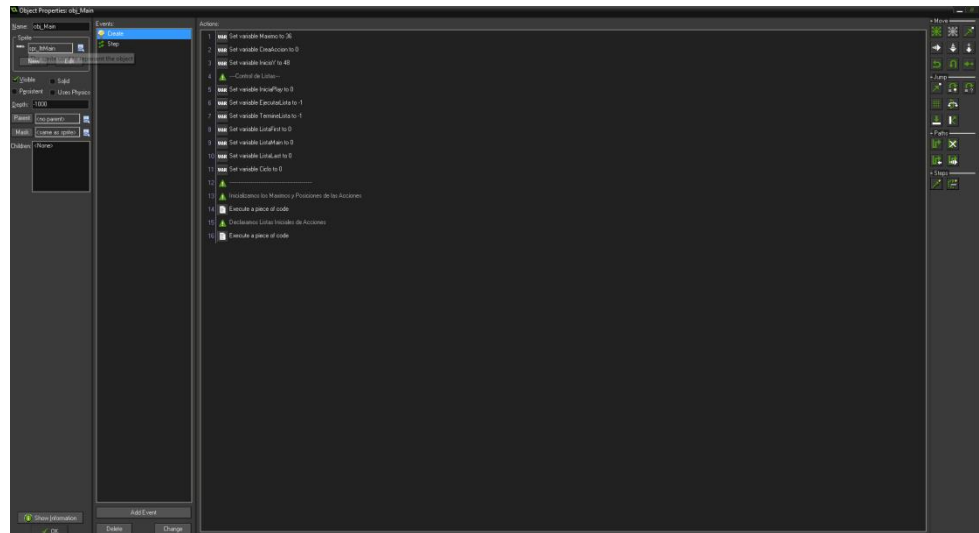


Figura 58 - Objeto Main: Código de evento “crear” (Fuente Propia)

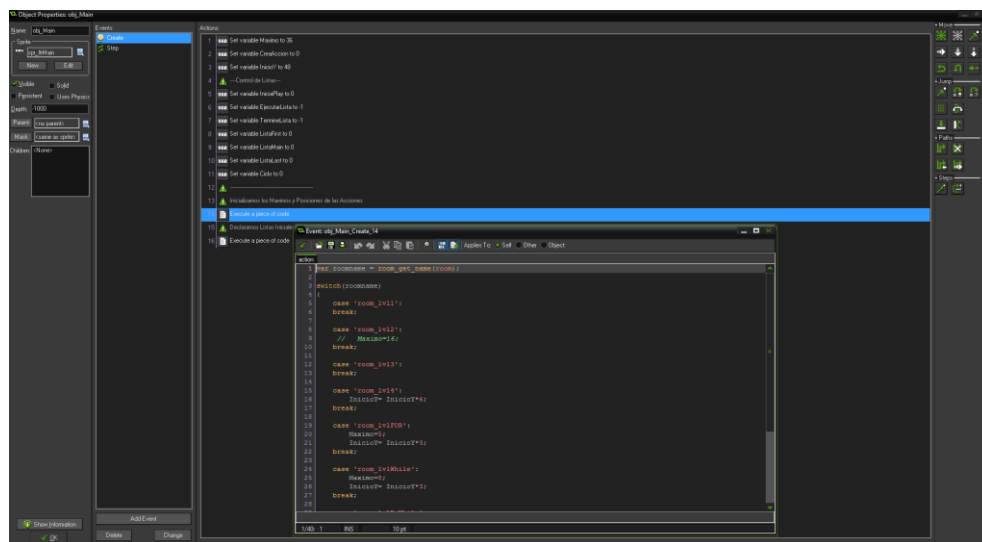


Figura 59 - Objeto Main: Código de procedimiento en evento “crear” (Fuente Propia)





- Objeto Fox: Es el personaje principal, tomará las órdenes ingresadas por el alumno para tratar de llevar el Diskette así el Robot y activarlo (ver Figura 62 y Figura 63).

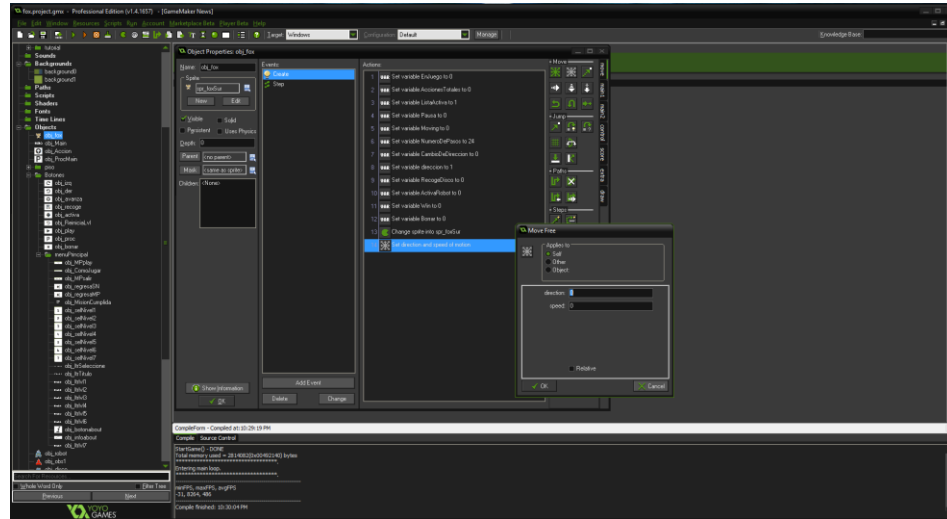


Figura 62 - Objeto Fox: Programación de función del objeto en evento “crear” (Fuente Propia)

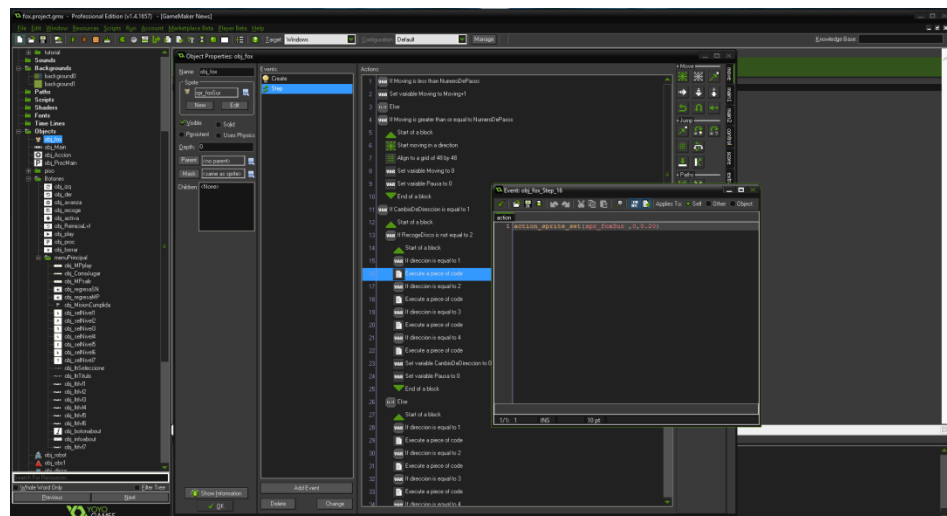
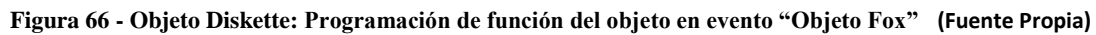
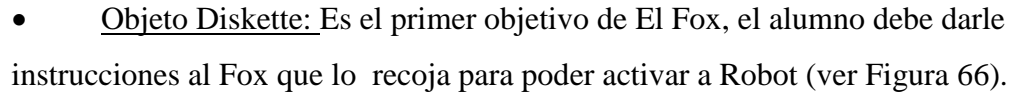


Figura 63 - Objeto Fox: Código de procedimiento en evento “paso” (Fuente Propia)



-
- The screenshot displays the Xilinx ISE 10.1 Professional Edition software interface. The top menu bar includes File, Edit, Window, Project, Tools, and Help. The Project Navigator on the left shows a project named 'myproj' with various files and folders. The Hierarchy View in the center shows a block diagram with a 'myproj' block. The Change Sprite dialog box is open, showing the 'Sprite' field set to 'myproj', and the 'Number' and 'Speed' fields set to 0 and 1 respectively. The 'OK' button is highlighted.

Figura 67 - Objeto Tutorial: Programación de función del objeto en evento “paso” (Fuente Propia)

- ¿Cómo guardar la aplicación para Android? (ver Figura 68 y Figura 69)

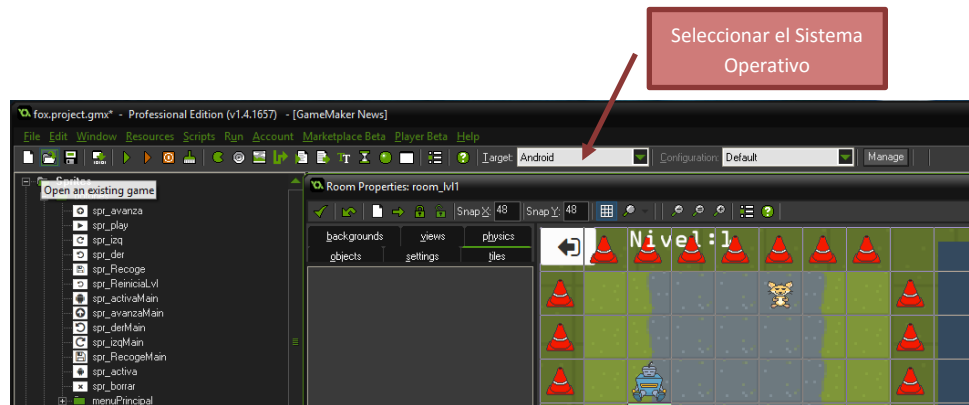


Figura 68 - Selección de Sistema Operativo para exportar la aplicación (Fuente Propia)

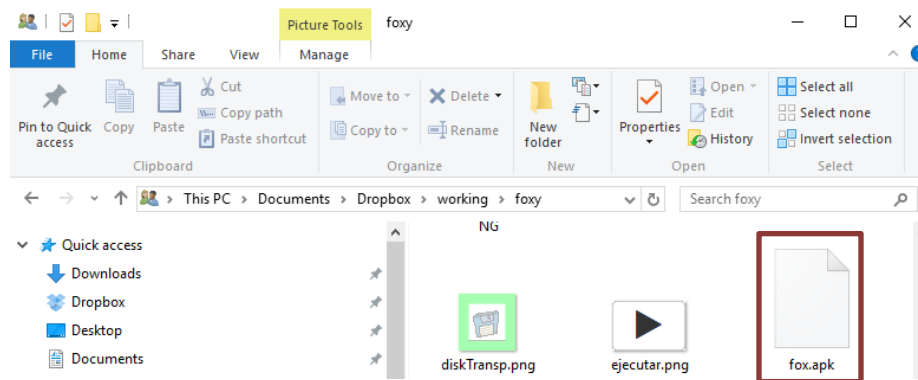


Figura 69 - Archivo para dispositivos con Sistema Operativo Andorid (*.apk) (Fuente Propia)