



UNAP E C
UNIVERSIDAD A P E C

DECANATO DE INGENIERÍA E INFORMÁTICA

ESCUELA DE INFORMÁTICA

**ANÁLISIS E IMPLEMENTACIÓN DE LA PROGRAMACIÓN DE SOFTWARE
COMO ASIGNATURA BÁSICA EN LAS ESCUELAS PÚBLICAS Y PRIVADAS A
NIVEL DE EDUCACIÓN MEDIA, EN LA CIUDAD DE SANTO DOMINGO
DURANTE EL PERIODO SEPTIEMBRE-DICIEMBRE 2020. CASO DE ESTUDIO:
INSTITUTO TECNOLÓGICO FABIO AMABLE MOTA**

PROYECTO DE TRABAJO DE GRADO PARA OPTAR POR EL TÍTULO DE:

INGENIERO EN SISTEMAS DE COMPUTACIÓN.

SUSTENTANTES:

Alan E. Rodríguez Pérez / 2013-1209

Luis J. Florentino Veloz / 2006-0336

ASESOR:

Ing. Freddy Jiménez

Santo Domingo, D.N.

Noviembre 2020

Los conceptos emitidos en el presente trabajo de investigación son de la exclusiva responsabilidad de quien (es) lo sustenta (n)

ÍNDICE

Contenido	Página
AGRADECIMIENTOS	I
DEDICATORIA	II
ASPECTOS METODOLÓGICOS	III
RESUMEN EJECUTIVO	IV
INTRODUCCIÓN	V
CAPÍTULO I: ASPECTOS GENERALES DEL INSTITUTO FABIO AMABLE MOTA	29
1.1 Introducción.....	30
1.2 Descripción del Instituto Fabio Amable Mota.....	31
1.3 Reseña histórica.....	32
1.4 Filosofía.....	34
1.4.1 Misión.....	34
1.4.2 Visión.....	35
1.4.3 Valores.....	35
1.5 Objetivos del Instituto Fabio Amable Mota.....	35
1.5.1 Objetivo General.....	35
1.5.2 Objetivos Específicos.....	35
1.6 Organigrama.....	36
1.7 Conclusión Capítulo I.....	36
CAPÍTULO II: CONCEPTOS SOFTWARE	38
2.1 Introducción.....	39
2.2 Definición de Programa de Software.....	40
2.3 Programación.....	40
2.4 Lenguajes de programación.....	43
2.4.1 Tipos de Lenguajes de Programación.....	43
2.4.2 Lista de Lenguajes de Programación.....	47
2.5 Paradigmas de Programación.....	47
2.5.1 Programación Estructurada y Modular.....	48
2.5.2 Programación Orientada a Objetos.....	50
2.5.3 Programación Concurrente.....	53
2.5.4 Programación Declarativa.....	54
2.6 Metodologías de Desarrollo de Software.....	54
2.6.1 Método Waterfall o Cascada.....	55
2.6.2 Método Agile.....	55
2.6.2.1 SCRUM.....	56

2.6.3	Método Espiral.....	57
2.6.5	Método RAD (Rapid Applications Development)	59
2.6.7.	Método XP (Extreme Programming)	59
2.7	Ventajas de Aprender a Programar	60
2.8	Beneficios de Aprender a Programar en educación Media	61
2.8.1	Creatividad	62
2.8.2	Trabajo Colaborativo o Trabajo en equipo	62
2.8.3	Razonamiento Computacional.....	63
2.8.4	Constructivismo	63
2.9	Herramientas para enseñar a programar en escuelas.....	63
2.9.1	MakeCode de Microsoft.....	64
2.9.2	Swift Playgrounds	66
2.10	PSeint	68
2.11	Python.....	69
2.12	Conclusión Capítulo II:.....	71
CAPÍTULO III: SEGURIDAD DE LOS SISTEMAS DE INFORMACIÓN		73
3.1	Introducción.....	74
3.2	Definición de Sistemas de Información	75
3.4	Manejo y Tratamiento Seguro de la Información	76
3.4.1	Como se clasifica la información.....	78
3.4.2	Como se almacena la información.....	79
3.4.3	Como se Respalda la información	81
3.5	Gestión de Riesgos de la Información	83
3.6	Firewalls.....	86
3.6.1	Tipos de Firewalls	87
3.7	Virus Informático.....	90
3.7.1	¿Qué es un virus informático?	90
3.7.2	Tipos de Virus Informático	91
3.7.3	Protección contra virus informáticos	92
3.8	Hacking Ético.....	92
3.8.1	Hackers	94
3.8.2.	¿Qué se necesita para ser un buen Ethical Hacker?.....	94
3.8.3	Ciclo del Hacking.....	95
3.8.4	Tipos de hackers.....	97
3.9	Ciberataques.....	97
3.9.1	Tipos de Ciberataques	97
3.10	Malwares.....	99

3.10.1	Tipos de Malwares	99
3.10.2	Protección contra Malwares	103
3.11	Seguridad Biométrica.....	105
3.11.1	Tipos de Seguridad Biométrica.....	105
3.11.1.1	Seguridad de lector de huellas.....	106
3.11.1.2	Verificación por voz.....	107
3.11.1.3	Patrones Oculares.....	108
3.12	Conclusión Capítulo III	108
CAPÍTULO IV: PROPUESTA DE IMPLEMENTACIÓN DE LA ENSEÑANZA DE PROGRAMACIÓN DE SOFTWARE COMO MATERIA BASE A NIVEL MEDIO.....		110
4.1	Introducción.....	111
4.2	Descripción de la Situación Actual del INSTITUTO FABIO AMABLE MOTA	112
4.3	Diseño de la red actual del INSTITUTO FABIO AMABLE MOTA.....	113
4.4	Descripción General de la Propuesta	115
4.4.1	Análisis FODA.....	118
4.4.2	Estudio de Factibilidad y Económica.....	119
4.4.3	Objetivos de la Propuesta.....	120
4.5	Desarrollo de la Propuesta.....	121
4.5.1	Plan de Estudios para aprender a programar a nivel medio de escuela	121
4.5.2	Herramientas de software para aprender a programar	127
4.5.3	Equipos (hardware) necesarios para programar	128
4.5.4	Ciclo de Inversión del Proyecto	129
4.5.5	Beneficios del Proyecto.....	129
4.6	Retorno de Inversión (ROI)	130
4.7	Cronograma de actividades de implementación del proyecto:	131
4.8	Formulario para el análisis de investigación de campo.....	131
4.9	Análisis y Resultado.....	133
4.10	Conclusión Capítulo IV.....	135
CONCLUSION.....		137
RECOMENDACIONES.....		140
BIBLIOGRAFIA.....		143
GLOSARIO.....		148
ANEXOS.....		155
Anexo. 1 – ENCUESTA.....		156
Anexo. 2 – ANTEPROYECTO.....		161
I - Título del tema.....		164

II- Introducción	165
III- Justificación	166
IV - Delimitación del Tema y Planteamiento del Problema.....	166
IV.I - Delimitación del Tema.....	166
IV.II - Planteamiento del problema	166
V - Objetivos	167
V.I - Objetivo General.....	167
V.II - Objetivos Específicos	167
VI - Marco Teórico.....	167
VI.I - Marco Teórico Referencial.....	167
VII - Hipótesis.....	174
VIII – Diseño metodológico: metodología y técnicas de investigación cuantitativa y/o cualitativa .	174

ÍNDICE DE FIGURAS

Contenido	Página
Instituto Fabio Amable Mota [Figura 1.1].....	31
Instituto Fabio Amable Mota [Figura 1.2].....	33
Organigrama Instituto Fabio Amable Mota [Figura 1.3]	36
Programación [Figura 2.1].....	40
La esencia de lógica de la programación: Algoritmos [Figura 2.2].....	41
PSeint Pseudocódigo [Figura 2.3]	42
La esencia de lógica de la programación: Lenguaje de Bajo Nivel [Figura 2.4].....	44
Lenguaje Ensamblador Figura [2.5].....	44
Los 10 lenguajes de programación más populares en 2018 Figura [2.6].....	45
La esencia de lógica de la programación: Interpretados, [Figura 2.7].....	46
La esencia de lógica de la programación: Compilación [Figura 2.8].....	47
Programación Modular. [Figura 2.9].....	49
Características de la Programación Modular. [Figura 2.10].....	50
Programación Orientada a Objetos: Creación de Clases y Objetos. [Figura 2.11].....	51
Programación Orientada a Objetos: Elementos [Figura 2.12].....	53
Metodología en Cascada [Figura 2.13].....	55
Metodología Agile [Figura 2.14].....	56
Scrum Methodology [Figura 2.15].....	57
Modelo Espiral [Figura .16].....	58
Prototipos [Figura 2.17].....	59

RAD [Figura 2.18].....	59
Metodología XP [Figura 2.19].....	60
Creatividad [Figura 2.20].....	62
Enseñanza Constructivista [Figura 2.21].....	63
Dispositivo Micro:bit de Microsoft. [Figura 2.22].....	65
Dispositivo Circuit Playground Express. [Figura 2.23].....	65
Minecraft. [Figura 2.24].....	66
Dispositivo LEGO MINDSTORMS Education EV3. [Figura 2.25].....	66
Swift Playgrounds. [Figura 2.26].....	67
Swift Playgrounds: Página de Práctica. [Figura 2.27].....	68
Seguridad de la Información. [Figura 3.1].....	76
Sistemas de Almacenamiento de información [Figura 3.2].....	80
Gestión de Riesgo, [Figura 3.3].....	86
Firewall, [Figura 3.4].....	87
Encaminador con filtrado de paquetes [Figura 3.5].....	88
Pasarela a nivel de circuito [Figura 3.6].....	89
Pasarela a nivel de aplicación o servidor proxy: [Figura 3.7].....	89
Fases del Hacking Ético p.11. [Figura 3.8].....	93
Etapas del hacking. [Figura 3.9].....	96
Ciberataques. [Figura 3.10]	98
Número de paquetes de instalación maliciosos detectados, primer trimestre de 2019 – primer trimestre de 2020. [Figura 3.11].....	100
Distribución por tipo de los nuevos programas móviles detectados, cuarto trimestre 2019 y primer trimestre de 2020. [Figura 3.12].....	101

Distribución de exploits utilizados por los ciberdelincuentes por tipo de aplicaciones atacada, tercer trimestre de 2019. [Figura 3.13].....	102
Distribución por país de las fuentes de ataques web, primer trimestre de 2020. [Figura 3.14].....	102
Minucias de Huellas Dactilares. [Figura 3.15].....	107
Patrones Huellas Dactilares. [Figura 3.16].....	107
Diagrama de laboratorios de informática del Instituto Fabio Mota. [Figura 4.1].....	114
Nuevo Diagrama de laboratorios de informática del Instituto Fabio Mota. [Figura 4.2].....	117
Matriz de análisis FODA Programación en escuelas. [Figura 4.3].....	118
Diagrama de casos de uso Programación de Software Instituto Fabio Mota [Figura 4.4].....	126
Requerimientos de Software. [Figura 4.5].....	129
Retorno de Inversión (ROI). [Figura 4.6].....	130
Cronograma de actividades de implementación del proyecto. [Figura 4.7].....	131
Cronograma de actividades de implementación del proyecto. [Figura 4.8].....	131

ÍNDICE DE TABLAS

Contenido	Página
Clasificación de la información. [Tabla 3.1].....	80
Costos de los recursos humanos. [Tabla 4.1].....	119
Costos de los recursos humanos. [Tabla 4.2].....	120
Requerimientos de Hardware. [Tabla 4.3].....	128
Requerimientos de Software. [Tabla 4.4].....	129
Encuesta ANÁLISIS E IMPLEMENTACIÓN DE LA PROGRAMACIÓN DE SOFTWARE - INSTITUTO TECNOLÓGICO FABIO AMABLE MOTA. [Tabla 4.5].....	132
Resultados de encuesta ANÁLISIS E IMPLEMENTACIÓN DE LA PROGRAMACIÓN DE SOFTWARE -INSTITUTO TECNOLÓGICO FABIO AMABLE MOTA. [Tabla 4.6].....	133
Resultados de encuesta ANÁLISIS E IMPLEMENTACIÓN DE LA PROGRAMACIÓN DE SOFTWARE INSTITUTO TECNOLÓGICO FABIO AMABLE MOTA. [Tabla 4.7].....	134
Resultados de encuesta ANÁLISIS E IMPLEMENTACIÓN DE LA PROGRAMACIÓN DE SOFTWARE -INSTITUTO TECNOLÓGICO FABIO AMABLE MOTA. [Tabla 4.8].....	134

**ANÁLISIS E IMPLEMENTACIÓN DE LA PROGRAMACIÓN DE
SOFTWARE COMO ASIGNATURA BÁSICA EN LAS ESCUELAS
PÚBLICAS Y PRIVADAS A NIVEL DE EDUCACIÓN MEDIA, EN LA
CIUDAD DE SANTO DOMINGO DURANTE EL PERIODO
SEPTIEMBRE-DICIEMBRE 2020. CASO DE ESTUDIO: INSTITUTO
TECNOLÓGICO FABIO AMABLE MOTA**

AGRADECIMIENTOS

AGRADECIMIENTOS

Quiero empezar agradeciéndole a Dios por haberme guiado siempre durante estos años y también por haberme dado la fortaleza, paciencia y sabiduría necesaria para llegar a este punto de culminación de mi carrera. A mis padres, María F. Pérez Batista y Félix Rodríguez De Oca por haberme dado palabras de aliento para seguir luchando sin importar las dificultades, y también por haberme apoyado en lo económico y emocional para poder culminar mi carrera.

A mis hermanas Vianny Rodríguez Pérez y Marlenny Rodríguez Pérez por haberme brindado su ayuda y apoyo en todo lo que estuvo dentro de sus capacidades.

A mi novia Teany Isabel Soto Ramírez, por también haber estado conmigo durante estos años y por haberme motivado y apoyado en los momentos y decisiones más difíciles, en lo personal, académico y profesional.

A nuestro asesor de trabajo de grado, el Ing. Freddy Jiménez, por siempre estar en disposición de orientarnos por el mejor camino y por haber brindado su ayuda durante este proceso.

Finalmente quiero agradecer a mis queridos amigos que de forma directa e indirecta me han ayudado en varios puntos de mi carrera, hasta llegar a este punto de culminación.

Alan E. Rodríguez Pérez

AGRADECIMIENTOS

Primero que todo agradecerle a Dios el permitirme llegar a esta meta en la vida, ser un profesional. Me tomó mucho tiempo, pero desde que me senté en el 2019 a analizar que quería lograr en la vida pues entendí que el primer paso era este.

A mis padres, Bernardina Rosario y Felipe Florentino por siempre haber estado ahí para mí, por siempre motivarme a terminar lo que había empezado, por ser mi soporte en todos los aspectos de mi vida.

A mi novia Ana Issa Diaz, por haberme dado palabras de aliento a través de este último año, siempre motivándome a seguir adelante y apoyarme en todo lo concerniente a este último tramo de mi carrera.

Gracias totales al Ing. Freddy Jiménez, por siempre estar en la mejor disposición de guiarnos y proveernos su excepcional asesoría. Sus consejos durante este tramo nos fueron de gran ayuda.

Finalmente quisiera agradecer a mis amigos cercanos que siempre me motivaron y siempre me dieron su apoyo para lograr esta meta. En especial quiero mencionar a Edwin Ramos, Francisco Rodríguez, Jonathan Gonzalvo, Rafael Medrano, León Aquino, Edwin Nolasco, Luis Mejía entre otros. Eternamente agradecido por su apoyo.

Luis J. Florentino Veloz

DEDICATORIA

DEDICATORIAS

Empezando con Dios por la guía constante que obtenemos al leer su palabra, por haberme dado la fortaleza requerida para seguir durante estos años.

También le dedico este trabajo que representa años de sacrificio y esfuerzo a mis padres María F. Pérez Batista y Félix Rodríguez De Oca por haberme dado una excelente educación y por ser ejemplos de honestidad e integridad, que son los valores que me inculcaron siempre, por ser las personas que más se preocupan por mí en este mundo y que siempre han estado conmigo, anteponiendo sus necesidades para que mis hermanas y yo tengamos todo lo necesario siempre.

A mis hermanas Vianny Rodríguez Pérez y Marlenny Rodríguez Pérez por haber sido un punto de apoyo en varios momentos de mi vida, tanto en lo personal como en lo profesional.

A mi novia Teany Isabel Soto Ramírez por su apoyo y motivación constante al momento de tomar decisiones importantes y también por estar ahí siempre para prestar su ayuda durante mis años de estudio.

Y finalmente al resto de mi familia de que una forma u otra también han contribuido con mi desarrollo personal y profesional, por estar ahí en los momentos más críticos de mi vida.

Alan E. Rodríguez Pérez

DEDICATORIAS

Primero que todo a Dios por darme la fortaleza, la sabiduría y la determinación y perseverancia necesaria a través de todos estos años.

A mis padres por haber sido pilar y soporte en este peldaño de mi formación profesional. Por haber sido soporte en todo momento sin escatimar esfuerzos, por ser mí ejemplo para seguir en la vida tanto profesional como privada. Sus valores inculcados siempre tendrán un espacio en toda decisión que tome en la vida.

A mi novia Ana Issa Diaz, tu paciencia y apoyo durante este tramo fue de gran ayuda. Gracias por ayudarme a tomar las decisiones correctas que en algún momento tuvieron impacto en el curso de mi desarrollo personal y profesional.

A mis amigos, gracias por haber estado ahí siempre para cualquier consejo, por haberme dado fuerzas para seguir y compartir mis logros. A todos los profesores que tuvieron un impacto a lo largo de estos años de formación, sus enseñanzas perdurarán para siempre.

Finalmente quiero dedicar esta tesis a todos los que en algún momento de mi aportaron su granito de arena. Gracias totales.

Luis J. Florentino Veloz

ASPECTOS METODOLÓGICOS

Justificación

La programación como materia en escuelas a nivel medio es conveniente en muchos sentidos (social, económico, etc.), pero lo principal es en la parte educativa, con esta materia los estudiantes adquieren habilidades como pensamiento crítico, resolución de problemas, desarrolla el espíritu emprendedor al crear programas, entre otras.

Además de obtener beneficios, al terminar la escuela media o bachillerato tendrán un conocimiento por el que se paga muy bien en las empresas y es un área con mucho desarrollo en la actualidad así que esta materia en las escuelas capacita a los jóvenes de mejor manera para el mundo real.

En República Dominicana una gran parte de los estudiantes no tienen la formación necesaria en cuanto a contenido en el pensum escolar. Esto deja lagunas en el pensamiento lógico de ellos, con la entrada de la programación como parte integral de su formación, dichas lagunas se pueden mitigar.

Esta investigación ayudará a tener los puntos claros cuando nos referimos a la programación de software en escuelas de nivel medio. Al tener documentación como esta se puede entender por qué la incorporación de estas materias en escuelas de nivel medio puede ser de beneficio para las estudiantes, por ende también para la sociedad ya que esta materia ayuda bastante a crear más jóvenes capaces que aporten valor a la sociedad dominicana.

Delimitación del Tema y Planteamiento del Problema

Delimitación del Tema

El Instituto Tecnológico Fabio Amable Mota., ubicado en la Prolongación Avenida Venezuela #2, Santo Domingo Este, 11906, Rep. Dom. en el periodo mayo-agosto 2020.

Planteamiento del problema

El Instituto Tecnológico Fabio Amable Mota no contempla actualmente con la programación de software como materia base para todos sus estudiantes, solamente está incluida en el bachillerato técnico en desarrollo y administración de aplicaciones informáticas. Bajo el pensum actual los estudiantes no cuentan con las herramientas necesarias para dominar los conceptos básicos de la programación en general.

La programación es necesaria en todos los niveles, no solamente en el área técnica de informática. Las demás modalidades del recinto necesitan incorporar esta materia para ayudar a sus estudiantes a desarrollar su mente y la lógica detrás de cada proyecto que quieran emprender. Al incorporar esta materia podemos mejorar la calidad educativa del instituto por ende mejorará el nivel académico de sus estudiantes.

Objetivos de la investigación

Objetivo General

Implementar la enseñanza de la programación de software como materia base a nivel medio, en el INSTITUTO FABIO AMABLE MOTA, en la ciudad de Santo Domingo, República Dominicana, en el periodo septiembre-diciembre 2020.

Objetivos Específicos

1. Mostrar quienes serían los beneficiados al integrar la programación en las escuelas.
2. Determinar las herramientas y métodos que se usarán para enseñar programación de software en las escuelas.
3. Evaluar y presentar la eficiencia de las herramientas y métodos que se usarán para enseñar programación en las escuelas.
4. Evaluar y describir el impacto económico, social y educativo que tendría la enseñanza de programación de software en las escuelas en la sociedad dominicana.

Elaboración de Hipótesis

Aprender a programar puede ser de mucho beneficio para los jóvenes que están cursando el bachillerato, y más ahora en esta época digital, la programación los ayudará a ser más creativos, a asumir compromisos cuando se desarrolla un proyecto, a aprender sobre el trabajo colaborativo, además de que están aprendiendo una habilidad muy demandada en la actualidad, con la programación estarán mejor preparados para la sociedad una vez se gradúen.

Identificación de las variables

- **Variable Independiente:** Programación de software como materia base en escuelas.
- **Variable Dependiente:** mejora de habilidades cognitivas del estudiante.

Enfoque de Investigación

Utilizaremos como enfoque para esta investigación los métodos de investigación cuantitativa y la investigación cualitativa.

Tipo de estudio

Para llevar a cabo la investigación, emplearemos los siguientes tipos de estudio:

Descriptivo: El tipo de estudio será el descriptivo, en el que vamos a describir cómo se implementará la enseñanza de programación de software, que herramientas y métodos que utilizarán para este propósito, también se describirán los beneficios que adquirirán los estudiantes al aprender programación como parte del pensum escolar.

Exploratorio: también usaremos el tipo de estudio exploratorio, que se usará para recolectar información de libros, artículos de periódicos, artículos de revistas y por último trabajos de grados que tengan relación al nuestro.

Método de investigación

Método analítico: este método nos ayudará a identificar las mejores técnicas y herramientas a usar para incluir la programación de software en el pensum de las escuelas, en este caso el Instituto Tecnológico Fabio Amable Mota.

Método descriptivo: este método se utilizará para poder brindar una descripción bien detallada de todo lo relacionado con la implementación de la enseñanza de programación de software en el Instituto Tecnológico Fabio Amable Mota y sus beneficios.

Fuente y técnicas para recolección de información

Recolección de información: esta se obtendrá de libros de programación, educación u otros temas relacionados a nuestro trabajo de grado, también de trabajos de grado relacionados al nuestro y de artículos (de periódicos, revistas y web) que podamos encontrar.

Grupo de enfoque: se harán grupos de enfoques con estudiantes del centro educativo, esto es, con el fin de poder obtener información fidedigna sobre el pensum escolar del centro y saber que piensan los estudiantes sobre la propuesta.

RESUMEN EJECUTIVO

RESUMEN EJECUTIVO

El presente trabajo de grado trata de la implementación de la enseñanza de la programación de software como materia básica a nivel medio en las escuelas de República Dominicana, teniendo como objeto de estudio el INSTITUTO FABIO AMABLE MOTA.

Este trabajo de grado tiene como objetivo principal formular y presentar un plan que permita la incorporación de la programación de software como materia en las escuelas que permitirá a los estudiantes desarrollar ciertas habilidades que lo preparen de mejor manera al momento de culminar la escuela media.

Es una propuesta novedosa ya que en la República Dominicana no contamos con este tipo de materias a nivel medio en las escuelas. Por lo que la programación como materia base podrá ser de gran aporte a los estudiantes.

Dicha implementación mejoraría la calidad de la educación en nuestro país, porque esta materia desarrolla la capacidad de resolución de problemas, la capacidad analítica/crítica y también el razonamiento, por ende, mejoraría la capacidad de los estudiantes.

La calidad de la educación debe ser un factor importante en nuestro país si se quiere seguir el camino del desarrollo, por lo que agregar una materia como la programación que se convertirá en una habilidad fundamental en el futuro es sumamente importante para todas las naciones en especial la nuestra. Los estudiantes mientras más preparación tengan de cara a sus estudios universitarios serán más efectivos a la hora de resolver problemas. Esto se traduce a beneficios para el país ya que tendrán más personas capacitadas.

En la actualidad vivimos en un mundo digital casi en su totalidad por lo que aprender a programar es algo fundamental para las personas. Es por esto por lo que debemos enfocarnos en preparar

jóvenes con los conocimientos necesarios para poder enfrentar los retos del mañana, y esto es lo que puede brindar la programación en las escuelas.

Los jóvenes son el futuro de toda nación progresista y en vías de desarrollo. Si logramos capacitar a nuestros estudiantes esto se traducirá en un mejor país, mejores ingresos, mejores oportunidades, personal capacitado, etc. La programación es necesaria para abrir la mente y ayudará tanto a estudiantes como maestros a poder identificar soluciones en cualquier problema práctico que se les pueda presentar.

La programación puede abrirnos las puertas que necesita nuestra República Dominicana para ser considerado un país futurista y acorde con los nuevos tiempos. Nuestros jóvenes tendrían mejor calidad de vida y sus aportes podrían ser implementados en nuestro país.

INTRODUCCIÓN

INTRODUCCIÓN

El mundo ha cambiado mucho desde el siglo anterior hasta la fecha, nos encontramos viviendo en la llamada era digital en la que no saber manipular las nuevas tecnologías no es opción para muchas personas, y en muchas profesiones se pide cada vez más ser más diestro con el uso la tecnología actual y la demanda de profesionales con más capacidad de análisis y resolución de problemas a aumentado considerablemente. Las instituciones educativas deben ser capaces de educar a estudiantes más capaces en muchos sentidos, actualmente es una necesidad que los estudiantes de las escuelas se gradúan teniendo conocimiento de las famosas TICS (Tecnologías de Información y Comunicación) para así poder desempeñarse mejor y tener más oportunidad en la sociedad en la cual se desenvuelven al terminar sus estudios.

El poder educar jóvenes con una buena capacidad de análisis y capacidad de resolver problemas puede ser difícil si no se usa el método apropiado, por eso la programación de software brinda una solución a esta situación. El aprender a programar en la juventud puede aportar muchos beneficios a los estudiantes, porque la programación se trata precisamente de analizar y resolver problemas usando la lógica, además de que es un método en el que se está creando algo, y una de las mejores formas de aprender es mediante la creación.

Para la implementación de la programación como materia en escuelas, en los primeros niveles nos enfocaremos en el uso de herramientas didácticas como

MakeCode de Microsoft, SwiftPlaygrounds de Apple. En cursos con estudiantes de más edad (13 años en adelante) se usaron ya lenguajes de programación reales como Python. Básicamente se usarán herramientas y métodos que permiten al estudiante aprender mientras va creando sus propios programas, lo que irá aumentando la creatividad, capacidad de análisis y pensamiento crítico del estudiante.

El Instituto Tecnológico Fabio Amable Mota, enseña programación solo a los estudiantes de informática, pero proponemos enseñar esto en todos los niveles de educación media, y en todas las áreas técnicas, con el objetivo final de que todos los estudiante graduados de bachiller puedan desarrollar su habilidad para resolver problemas tengan un buen nivel de conocimiento de programación de software, que le permita hacer carrera en esta área si así lo desea.

**CAPÍTULO I: ASPECTOS GENERALES DEL
INSTITUTO FABIO AMABLE MOTA**

1.1 Introducción

El Instituto Fabio Mota es un instituto politécnico en el cual los jóvenes aprenden diversas habilidades desde carreras técnicas como electrónica hasta carreras como enfermería. Este instituto se enfoca en educar a los jóvenes tanto en materias básicas como matemáticas y también en otras áreas como electricidad o informática. Pero el hecho de que sea un instituto tecnológico no significa que no se pueda mejorar lo que se imparte en ese instituto.

En las escuelas siempre se ha buscado orientar y educar a sus estudiantes de la mejor manera posible, en muchos casos se ha tenido que innovar, es decir, agregar algo nuevo para enseñar o cambiar el programa completamente, en este trabajo lo que proponemos es agregar una nueva materia al pensum del Instituto Fabio Amable Mota con el objetivo de mejorar la calidad y la capacidad de los estudiantes de este instituto, la nueva materia es la Programación de Software, aunque el instituto ya cuenta con un plan de estudios a nivel técnico para los estudiantes y en una de esas áreas técnicas (informática) ya se da esta materia, pero nuestra propuesta se base en agregar esta materia para todos los estudiantes del instituto.

1.2 Descripción del Instituto Fabio Amable Mota

Es un centro educativo técnico profesional, de carácter público designado con el nombre de Dr. Fabio Amable Mota, como un homenaje póstumo al educador y humanista Dr. Fabio Amable Mota Medrano, durante el gobierno constitucional presidido por el Ex. presidente de la República Dominicana, el Dr. Joaquín Balaguer Ricardo, período 1974-1978. El instituto, en la actualidad tiene una matrícula de más de 648 estudiantes, 24 aulas, 10 laboratorios y se imparten 7 carreras técnicas. (Instituto Fabio Amable Mota, 2019, p.9).



Los Autores: Rodríguez, A. y Florentino, L. (2020). Instituto Fabio Amable Mota. [Figura 1.1]. República Dominicana

El Instituto Tecnológico Fabio A. Mota, forma jóvenes de escasos recursos económicos en las carreras técnicas, potenciando capacidades, inculcando valores y aptitudes para ofrecer sus servicios cualificados a las empresas e instituciones que promueven el desarrollo del país. (Instituto Fabio Amable Mota, 2019, p.10).

El Instituto Fabio Amable Mota (2019, p.9), como institución pública tiene las puertas abiertas a jóvenes de diferente sexo, edad, religión e intereses políticos y sociales. Cumpliendo con el

sagrado derecho que le corresponde a cada joven de recibir una educación formal y dual para que en el futuro:

- Pueda desarrollar su capacidad técnica, su personalidad y respetar los derechos y deberes democráticos fundamentales.
- Sea consistente sobre el significado de la libertad, ame a su patria y coopere con el desarrollo tecnológico, científico y humanístico para una sociedad más justa.
- Sea un sujeto crítico y creativo, con valores en el marco de la familiarización local e internacional, procurando la paz y el respeto a los derechos ciudadanos y del medio ambiente.
- Desarrolle la estructura del país mediante la aplicación de los conocimientos científicos, técnicos y humanísticos que demanda una sociedad en pleno proceso de desarrollo y de crecimiento.
- Sea capaz de aprender a aprender, aprender a hacer, aprender a ser durante toda su vida.
- Pueda poner en práctica capacidades técnicas y humanas para transformar positivamente su entorno.

1.3 Reseña histórica

Los orígenes del Instituto Fabio Amable Mota datan del año 1970. El 19 de octubre de dicho año mediante la ordenanza 1-70, se estableció una nueva estructura y un nuevo currículo en la educación media nacional que tiene la finalidad de “el desarrollo integral de la personalidad del adolescente mediante la adquisición del patrimonio cultural y la comprensión del mundo en que viven, la preparación para el cambio social y la participación en el mundo del trabajo”. (Instituto Fabio Amable Mota, 2019, p.3).



Los Autores: Rodríguez, A. y Florentino, L. (2020). Instituto Fabio Amable Mota. [Figura 1.2]. República Dominicana

Por disposición de la Secretaría de Estado de Educación, el 1ero de octubre de 1972, se inició en el liceo Ramon Emilio Jiménez el primer curso del plan acordado para la reforma con 12 cursos y una población de unos 400 estudiantes. Durante el año escolar 1977-1978 y conformado por el excedente de estudiantes y profesores del liceo Ramon Emilio Jiménez se funda el liceo de educación media Dr. Fabio Amable Mota que funcionaba para el ciclo superior del plan de reforma. En este mismo recinto funcionaba también el liceo Domingo Faustino Sarmiento. (Instituto Fabio Amable Mota, 2019, p.3).

Desde el año 1978 y hasta el 1992 ambos liceos se complementaron para servir de ciclo superior y ciclo básico del plan de reforma contemplado por la Secretaría de Educación. En el año 1992 se inicia la modalidad técnico profesional en el liceo Fabio Amable Mota y adquiere la categoría de politécnico con 3ro / 4to de bachillerato. Esto se logra con la ordenanza 1-95. (Instituto Fabio Amable Mota, 2019, p.3).

Para el año 2001-2002 se inicia un profundo proceso de transformación y reestructuración de toda la estructura de gestión y de administración para relanzar la institución. Para tales fines se implementaron 8 periodos de clases de 55 min cada uno, este proceso inició con los 3ros de

bachillerato durante el primer año y en el segundo año fue expandido hasta 4to de bachillerato. (Instituto Fabio Amable Mota, 2019, p.3).

En el año 2002-03, con la llegada de un equipo de gestión colegiado, el Instituto se relanza y presenta su primer proyecto educativo de centro y su nuevo proyecto curricular de centro 2003-2008. Estos, combinados con el apoyo estatal y la integración de los estudiantes, profesores, padres/madres, iglesias, grupos empresariales etc. Lograron devolver el prestigio a este reconocido centro y hoy se presenta como uno de los más importantes del país con inversiones en su infraestructura ascendente a los 100 millones de pesos. (Instituto Fabio Amable Mota, 2019, p.4).

1.4 Filosofía

El Instituto Tecnológico Fabio Amable Mota es un centro educativo de carácter público, abierto a todas las corrientes del pensamiento social. Tiene el compromiso de formar en competencias técnicas y humanísticas a jóvenes de diferentes sexos, religiones, estratos sociales e intereses políticos. Como entidad educativa ofrece educación formal, capaz de desarrollar la personalidad del sujeto creando en ellos personas capaces de ejercer sus derechos y deberes con honestidad y responsabilidad, crítico, creativo, innovador y participativo. (Instituto Fabio Amable Mota, 2019, p.14).

1.4.1 Misión

Formar técnicos competentes, capaces de integrarse a la producción de bienes y servicios de manera eficiente como lo demanda la sociedad, produciendo mejoras en su calidad de vida, la de su familia y la del país, dispuestos a producir e insertarse en los cambios y transformaciones científicas, tecnológicas y humanísticas que requieren el mundo actual. (Instituto Fabio Amable Mota, 2019, p.13).

1.4.2 Visión

Ser el modelo de institución educativa Técnico-Profesional inclusiva que brinda una formación por competencias técnicas e integrales de calidad, preferida por la comunidad, las empresas e instituciones de bienes y servicios que requieren profesionales de unos altos estándares para aumentar su eficiencia y su competitividad. (Instituto Fabio Amable Mota, 2019, p.13).

1.4.3 Valores

- Respeto por la condición humana.
- Confiabilidad como actos que aseguran el éxito en los procesos.
- Búsqueda de la verdad mediante la reflexión y la investigación.
- La responsabilidad en los actos que realiza
- La calidad como objetivo de esfuerzo constante.
- La solidaridad como principio universal para construir una sociedad más justa y equitativa.
- La honestidad como principio que enaltece la condición y la actuación de los actos humanos.
- Compañerismo como hábito y ejercicio de vida que nos permite convivir.

1.5 Objetivos del Instituto Fabio Amable Mota

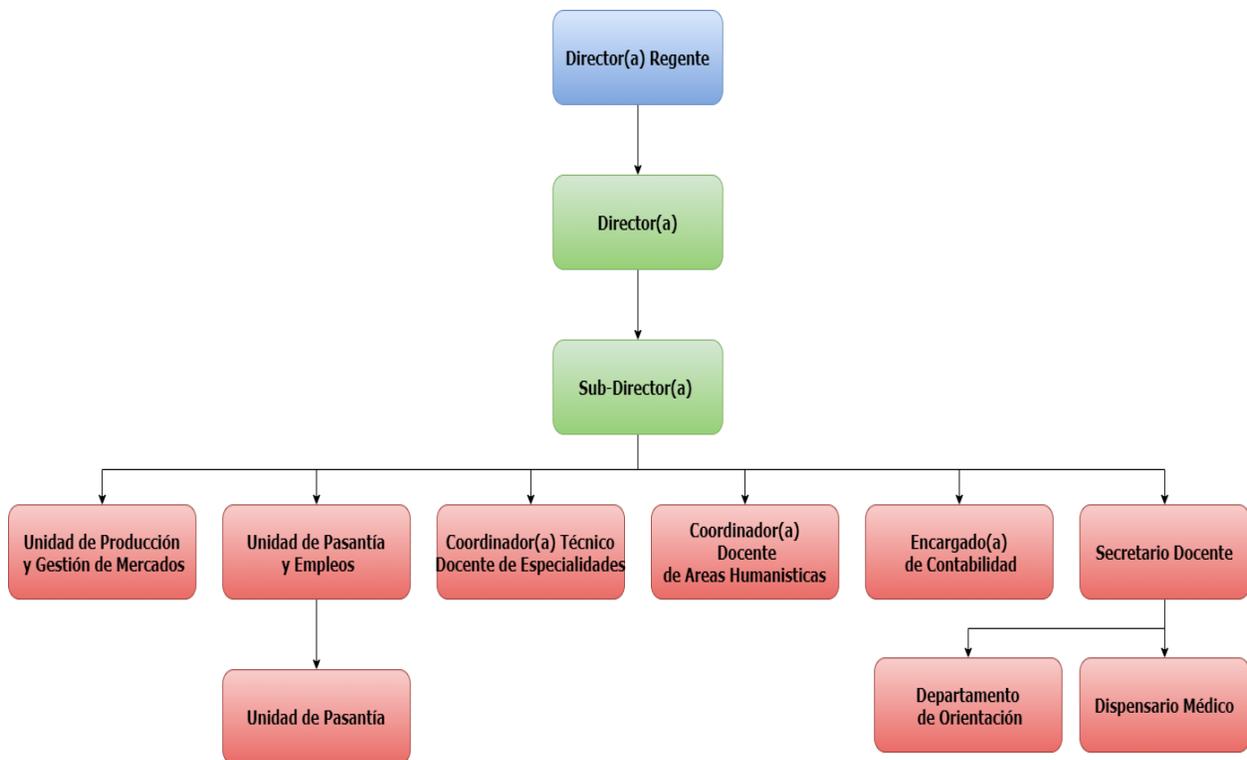
1.5.1 Objetivo General

Ofrecer una educación formal, capaz de desarrollar la personalidad del sujeto, creando en ellos personas capaces de ejercer sus derechos y deberes con honestidad y responsabilidad, crítico, creativo, innovador y participativo.

1.5.2 Objetivos Específicos

- Formar jóvenes de escasos recursos económicos en carreras técnicas, potenciando capacidades, inculcando valores y aptitudes para ofrecer sus servicios cualificados a las empresas e instituciones que promueven el desarrollo del país.
 - Lograr la equidad desde un punto de vista educativo, y dar sentido al sistema de evaluación educativa y elevar la calidad de la educación tecnológica y humanística de los estudiantes.
- (Instituto Fabio Amable Mota, 2019, p.15).

1.6 Organigrama



Los Autores: Rodríguez, A. (2020) Organigrama Instituto Fabio Amable Mota [Figura 1.3]

1.7 Conclusión Capítulo I

En resumen, podemos decir que el Instituto Fabio Amable Mota está muy bien preparado y decidido a mejorar la calidad de los estudiantes que pasan por dicha institución, debido a su historia y plan de educación, además hemos visto los aspectos que componen el Instituto Tecnológico Fabio Amable Mota y que es un centro que tiene un buen plan de capacitación, pero se puede mejorar aún más al agregar la programación de software como materia base ya que las habilidades y los beneficios que se pueden adquirir por medio de esta ayudaran en gran parte a que se puedan cumplir los objetivos propuestos por el instituto.

El Instituto busca ser un agente de cambio para la sociedad dominicana por lo que el implementar nuevas ideas que ayuden a los estudiantes es de las mejores maneras para seguir avanzando.

El organigrama del Instituto está sólidamente unido para perseguir estos objetivos y poder monitorear el progreso de ellos en cada departamento. Los organigramas son parte esencial de una estructura sólida y enfocada en conseguir los resultados deseados.

CAPÍTULO II: CONCEPTOS SOFTWARE

2.1 Introducción

La programación es una ciencia que consiste en la creación de programas con el objetivo de solucionar problemas. Una ciencia que te permite automatizar y también simplificar varias tareas a niveles que antes no se creían posibles. Al ser una ciencia tan efectiva se debe de entender ciertos conceptos sobre la programación, como el hecho de que existe diferentes formas de programar no solo no referimos al estilo de codificar de cada programador, sino también al paradigma que este use, aunque el más usado en la actualidad es el paradigma de la programación orientada a objetos, conceptos que veremos más adelante.

Existen diversas formas de aprender a programar, también diferentes lenguajes y herramientas que nos ayudan a esto, pero la naturaleza misma de esta ciencia hace que aprender a programar sea una actividad muy beneficiosa porque en la programación se aprende creando cosas que es la mejor manera de poder aprender una nueva habilidad. Además de que la programación es un área de la informática que aporta a nivel académico también aporta beneficios competitivos a nivel laboral y también económico ya que es de las áreas mejores pagadas y con más salida laboral.

2.2 Definición de Programa de Software

De manera sencilla podemos decir que un programa es un conjunto de instrucciones que se realizan para cumplir un propósito, pero si llevamos esto al terreno de la informática Llopis Pascual, Pérez López y Ortuño Ortin (2000) afirman que:

“Podemos definir un programa como un conjunto ordenado de instrucciones que guían a la computadora para que realice una tarea específica a través de sus operaciones. El programa, por tanto, indica a la máquina lo que debe hacer, en qué orden y con qué datos trabajar”.

Otra definición relaciona a este punto es la de Software, a primera instancia pensamos en un software como un programa de computadora o celular, pero una definición más acertada sería “Es un conjunto de instrucciones de programa detalladas que controlan y coordinan los componentes hardware de una computadora y controlan las operaciones de un sistema informático” (Aguilar, 2000, p.31).

2.3 Programación

La programación como ciencia es un tanto complejo de definir si no se tienen algunos conceptos básicos que involucran la parte de programar como tal, se debe saber sobre algoritmos, variables, operadores, etc.



Rafino, M. (2020). Programación. [Figura 2.1]. Recuperado de <https://concepto.de/programacion/>

Algoritmo: Es un conjunto de pasos secuenciales y ordenados que permiten lograr un objetivo. Que sean pasos secuenciales significa que deben ser ejecutados uno después de otro y que sean pasos ordenados quiere decir que deben llevar un orden casi obligatorio. (Trejos Buritacá, 1999, p.18).

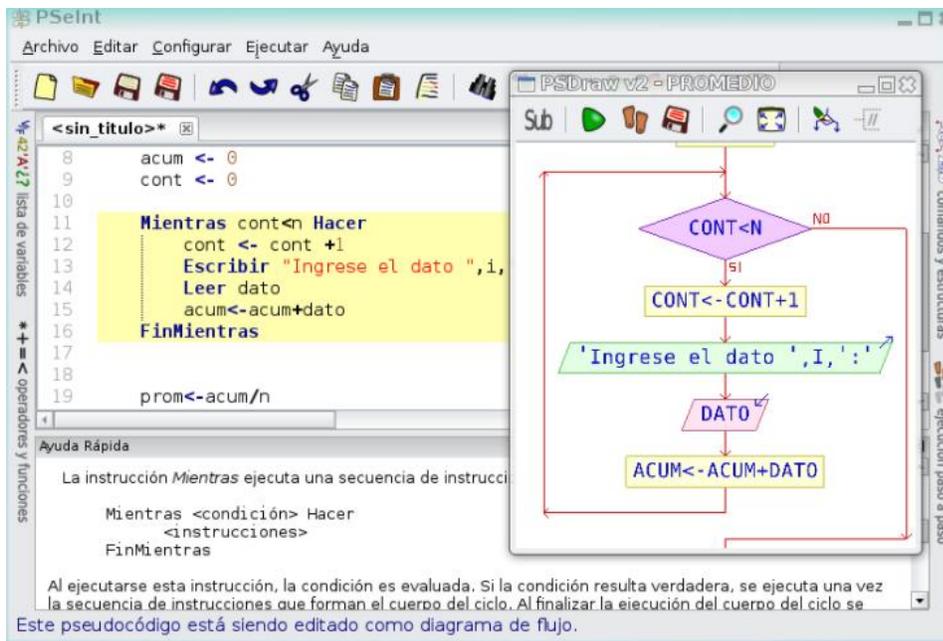
```
Algoritmo Adquisicion_Libro
Inicio
    1. Saber cuál es el libro que se quiere adquirir
    2. Desplazarnos hacia una librería
    3. Preguntar si tienen el libro que necesitamos
    4. Si lo tienen
        adquirirlo y Parar allí (dentro de este algoritmo)
        Si no lo tienen
            ir al paso 2
Fin
```

Trejos, O. (1999). La esencia de lógica de la programación: Algoritmos, Pág. 19. [Figura 2.2]

Variable: Una variable es como lo indica su nombre un valor cambiante en un sistema, “Técnicamente una variable es un campo de memoria al que se le puede cambiar su contenido cuantas veces sea necesario” (Trejos Buritacá, 1999, p.27).

Operadores: “Los operadores son signos que nos permiten expresar relaciones entre variables y/o constantes, relaciones de las cuales normalmente se desprende un resultado” (Trejos Buritacá, 1999, p.35).

Pseudocódigo: La palabra seudo significa falso, por lo que pseudocódigo se define como sigue, “Es la representación textual de un algoritmo de manera que dicho texto se encuentre enmarcado en algunas normas técnicas que faciliten su posterior transcripción a un lenguaje de Programación” (Trejos Buritacá, 1999, p.66).



PSeint (2020). PSeint Pseudocódigo. [Figura 2.3]. Recuperado de <http://pseint.sourceforge.net/slide/pseint.html>

Compilación: “Es el proceso a través del cual el computador revisa que el programa que hemos digitado se ajuste a las reglas sintácticas de un determinado Lenguaje de Programación” (Trejos Buritacá, 1999, p.22).

Ahora podemos decir que la programación es el proceso en el cual se toma un algoritmo y se modifica en alguna notación o más bien un lenguaje de programación que forma que este algoritmo ya codificado en un lenguaje pueda ser compilado y ejecutado por una computadora. Hay una inmensa variedad de lenguajes de programación y muchos tipos de computadores, lo primero que debemos tener al empezar a programar es la necesidad de solucionar un problema, por lo que creamos un algoritmo para resolver dicho problema, en otras palabras, no puede existir un programa sin su respectivo algoritmo.

2.4 Lenguajes de programación

Un lenguaje de programación se define como una serie de instrucciones que usan los programadores para crear programas de software ya sean aplicaciones móviles, web o de escritorio. Según Llorens Largo (2002) un lenguaje de programación es el idioma que utiliza la persona para comunicarse con el ordenador y dar instrucciones, por lo tanto, un papel fundamental en la programación.

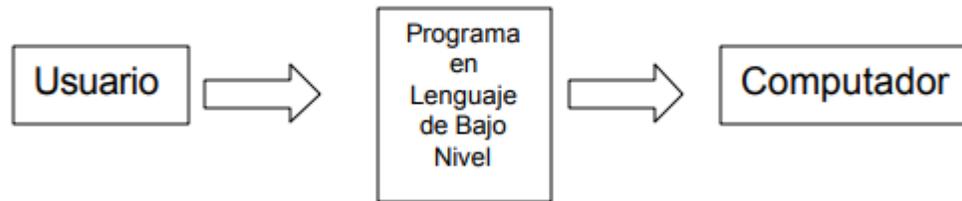
2.4.1 Tipos de Lenguajes de Programación

- **Lenguajes de Bajo Nivel**

Estos son los lenguajes cuyas instrucciones suelen ser muy complejas para entender a primera vista debido a su sintaxis y a su estructura, pero son fáciles de entender para una computadora. Un lenguaje de este tipo es el lenguaje Assembler, en este lenguaje las instrucciones están basadas en una serie de mnemónicos que dificultan la transcripción, este es el lenguaje con el que trabajan las computadoras lo que significa que las computadoras a nivel interno lo único que pueden entender es el lenguaje assembler o ensamblador.

Este tipo de lenguaje programación no es utilizado con frecuencia hoy día debido a su complejidad y su larga curva de aprendizaje, pero la programación con lenguajes de bajo nivel fue la que se usó en los primeros años de la programación, los programadores de assembler debían ser muy especializados dada la complejidad de las instrucciones que se usan en este lenguaje de bajo nivel.

Dicho esto, la programación de bajo nivel puede representarse con la siguiente figura:



Trejos, O. (1999). *La esencia de lógica de la programación: Lenguaje de Bajo Nivel*, Pág. 78. [Figura 2.4]

Ejemplo de Assamblar, código para un microcontrolador Intel 80C51:

```
ORG 8030H
include
T05SEG:
  SETB TR0
  JNB uSEG,T05SEG ;esta subrutina es utilizada
  CLR TR0 ;para realizar una cuenta de
  CPL uSEG ;0,5 segundos mediante la
  MOV R1,DPL ;interrupción del timer 0.
  INVOKE
  MOV R2,DPH
  CJNE R2,#07H,T05SEG
  CJNE R1,#78H,T05SEG
  MOV DPTR,#0
RET
```

Wikipedia (2020). *Lenguaje Ensamblador*. Figura [2.5]. Recuperado de https://es.wikipedia.org/wiki/Lenguaje_ensamblador

- **Lenguajes de Alto Nivel**

Este tipo de lenguaje se creó pensando en las complicaciones dadas en los lenguajes de bajo nivel, con el objetivo de hacer que la programación sea más sencilla de aprender. Ya que el que programa es el ser humano y no la maquina por lo que programar debería ser mucho más sencillo.

Se crearon los lenguajes de alto nivel porque son más entendibles para el ser humano, ya que es más fácil entender las instrucciones de este tipo de lenguaje, son básicamente un lenguaje fácil de comprender lo que hace la programación algo relativamente sencillo. Se les llama lenguajes de alto nivel sabiendo que el computador necesita otro programa que tomará lo que se escriba en alto nivel y lo convierta a lenguaje de bajo nivel (ya que este es el verdadero lenguaje del computador). De este viene lo que conocemos hoy como compiladores e interpretadores.

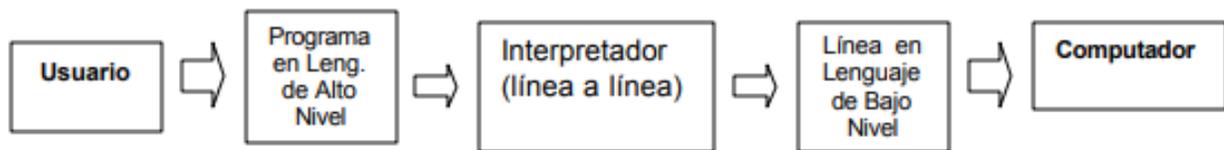


Izquierdo, R. (2018). *Los 10 lenguajes de programación más populares en 2018*. Figura[2.6]. Recuperado <https://pandorafms.com/blog/es/lenguajes-de-programacion/>

- **Lenguajes Interpretados**

Estos son los lenguajes de programación en donde se tiene un programa interpretador que se define como un programa que toma el código que creamos y lo convierte línea a línea a Lenguaje de Bajo Nivel y así mismo lo va ejecutando. El problema con estos lenguajes es que si el código tenía un error en algunas de las líneas, este solo era detectado en el momento que el intérprete llegaba a esta línea, lo que representa pérdida de tiempo y recursos ya que se detectaba el error después de haber interpretado las líneas anteriores a las líneas del error.

Aunque este lenguaje tiene sus desventajas, era el tipo que se usaba en los inicios de los lenguajes de alto nivel, por lo que los lenguajes de programación interpretados tomaron mucha popularidad porque el programador ya no tenía que gastar tiempo entendiendo las instrucciones del ensamblador.

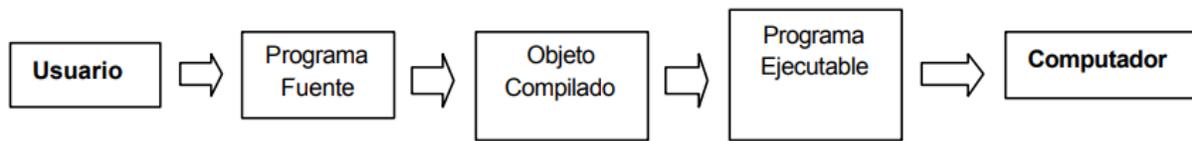


Trejos, O. (1999). *La esencia de lógica de la programación: Interpretados*, Pág. 79. [Figura 2.7]

- **Lenguajes Compilados**

Estos son los lenguajes en donde se utiliza un programa llamado compilador que toma todo el código escrito (código fuente), lo revisa en busca de errores de sintaxis (y a veces de semántica) y solo cuando todo el código esté bien escrito es cuando lo convierte a lenguaje de bajo nivel para que el computador lo ejecute. Por lo que de esta manera se reducía el riesgo de evaluación que representaban los lenguajes interpretados, ya que con uno compilado se solo se ejecuta cuando esté todo bien por lo que el programa no abortara cuando se esté ejecutando (lo que pasa con los interpretados si se encuentra un error en una línea al momento de su ejecución). En los compilados cuando se compila el código fuente se crea lo que se conoce como objeto compilado, cuando se va a ejecutar entonces este objeto compilado se organiza para que cada instrucción sea entendible y ejecutable por el computador, a este programa se le conoce como ejecutable.

El proceso de los lenguajes compilados lo podemos ver como sigue:



Trejos, O. (1999). La esencia de lógica de la programación: Compilación, Pág. 80. [Figura 2.8]

2.4.2 Lista de Lenguajes de Programación

A continuación, una lista de algunos de los lenguajes de programación más utilizados en la actualidad:

- Java
- C#
- Python
- JavaScript
- Kotlin
- Dart
- Go
- Swift
- R

2.5 Paradigmas de Programación

Al momento de programar existen diferentes maneras (paradigmas) de hacerlo, y estos paradigmas han surgido según las necesidades que surgen y también para mejorar las debilidades que tenían los paradigmas anteriores. Cada paradigma es único a su forma, algunos tienen características de paradigmas anteriores y características agregadas que los hacen únicos.

2.5.1 Programación Estructurada y Modular

Este es un paradigma de programación muy usado y más en los niveles iniciales cuando se empieza a aprender a programar. Se define como un grupo de técnicas que usan una cantidad definida de estructuras de control que suelen simplificar los problemas y también reduce los errores.

Algunas de las técnicas usadas en este paradigma son:

- Top Down o descendente: en este proceso el problema presentado se descompone en partes pequeñas para hacer que la resolución del problema sea más sencilla.
- Basic structures o estructuras básicas: un programa hecho con programación estructurada puede ser escrito estructuras secuenciales, selectivas y repetitivas.

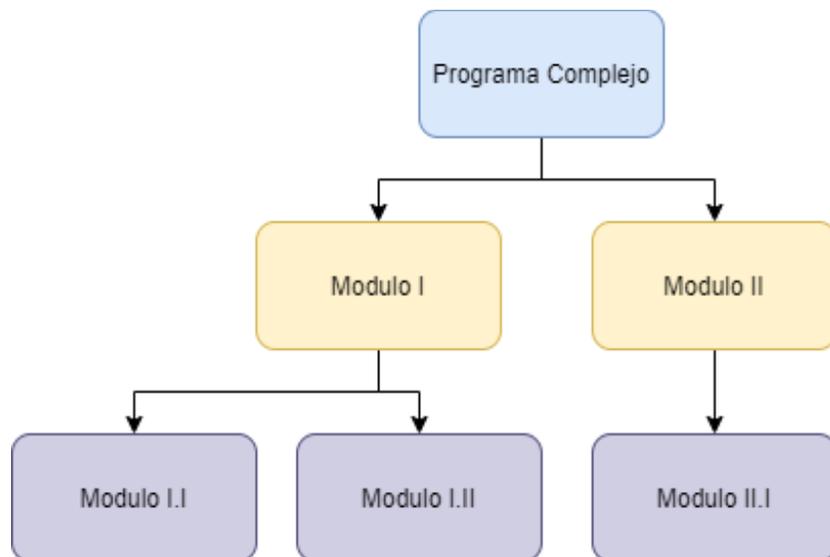
Un programa puede ser considerado estructurado si cumple con lo siguiente:

- Tiene un único entry point (punto de entrada) y un end point (punto de salida).
- Hay caminos desde el entry pitón hasta el end point que se puede seguir y pasan por todo el programa.
- Las instrucciones en el código son ejecutables y no hay bucles infinitos.

La programación modular existe desde los inicios de la programación como tal, podemos decir que esta fue incluso primera que la programación estructurada. La programación modular se define como el paradigma de programación en el que se usa el concepto de “divide y vencerás”, es decir, se toma un problema muy complejo y se divide en pequeños módulos o subproblemas, de forma

que estos subproblemas sean más fáciles de resolver que resolver todo el problema sin dividirlo, y por ende la solución de cada subproblema al combinarlas se tiene la solución general del problema más grande. En palabras simples la programación modular como tal nos permite:

- Fragmentar la complejidad de un problema complejo en pequeños subproblemas que son más simples de solucionar.
- También permite la reutilización de código de un software en cualquier momento de la ejecución de este.



Los Autores: Rodríguez, A. (2020). Programación Modular. [Figura 2.9]

Los módulos de un programa modular deben de cumplir con dos características principales, estas son llamadas cohesión y acoplamiento.

- Bajo nivel de acoplamiento: esto quiere decir que la dependencia entre los módulos del programa debe ser baja o lo más reducida posible.
- Alto nivel de cohesión: esta característica se trata de las órdenes o instrucciones que están dentro de un módulo deben de apoyar al ejercicio de la tarea.



Los Autores: Rodríguez, A. (2020). Características de la Programación Modular. [Figura 2.10]

2.5.2 Programación Orientada a Objetos

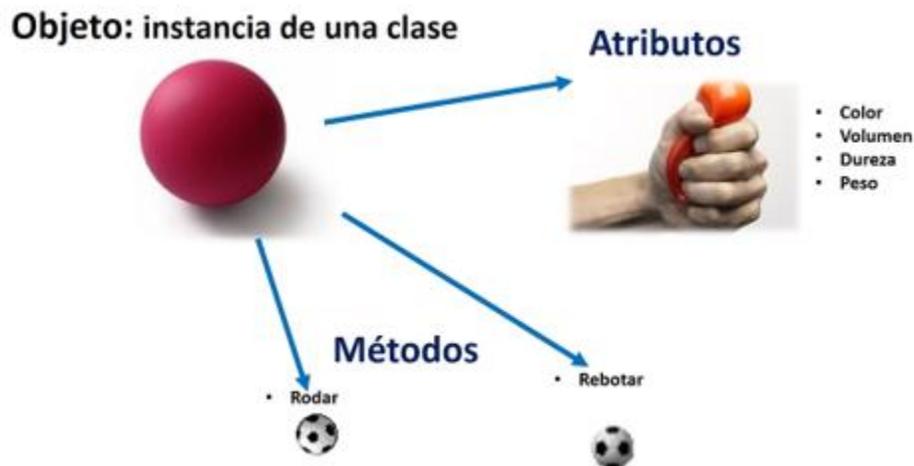
La programación Orientada a Objetos es el paradigma de programación más usado hoy día, es básicamente el paradigma en cual se basan todos los lenguajes modernos y algunos ya algo viejos, entre estos lenguajes tenemos a Java, C#, Python, JavaScript, etc.

En la **Programación Orientada a Objetos (POO)** el programa se compone de varios objetos que interactúan entre sí a través de métodos. La POO podemos definirla como una metodología de creación de software y también como paradigma de programación en el cual se define los programas como clases de objetos, estos objetos son entidades del mundo real que pueden o no cambiar de estado y comportamiento.

La POO también se define como un programa con un conjunto de objetos creados a partir de clases que se comunican entre ellos para poder llevar a cabo una tarea y además la POO es un modelo que representa un subconjunto del mundo real. En este paradigma vemos que los objetos de clases al igual que los objetos de la vida real tienen características o atributos (variables) y también tienen

comportamientos o métodos (algo parecido a las funciones). A diferencia de los demás paradigmas este define objetos y después envían mensajes a los objetos diciendo que realicen o ejecuten algún método a él que ese objeto tenga acceso.

A continuación, una imagen que refleja la esencia de la programación orientada a objetos.



Narváez, A. (2017). *Programación Orientada a Objetos en Java - #1: Creación de Clases y Objetos*. [Figura 2.11]. Recuperado de www.youtube.com/watch?v=y7eyMeEZswg

La POO tiene varias ventajas, entre estas está que los métodos están hechos para hacer programas y módulos que sean fáciles de mantener, escribir y volver a usar. Los componentes esenciales en la POO son: los objetos, las clases y los métodos.

- **Objeto:** se pueden definir como las unidades básicas de construcción, para la conceptualización, diseño o programación, esto es que son instancias agrupadas en clases con características en común y que son los atributos y procedimientos, conocidos como operaciones o métodos (Arenas Hernández, 2004, p.3).

También podemos decir que los objetos son una abstracción encapsulada de datos, es decir, una entidad con atributos y comportamiento (métodos).

- **Métodos:** Los métodos son en sí el comportamiento de un objeto básicamente es una acción que determina cómo deberá actuar un objeto, esto es un poco parecido a lo que se conoce como funciones en la programación estructurada.
- **Clases:** Las clases se definen como el modelo en el que se define un conjunto de variables (atributos) y métodos (comportamiento) que son apropiados para operar con dichos datos. Todo objeto que se cree a partir de una clase se le denomina instancia de la clase.

La programación orientada a objetos tiene 5 elementos principales, que son:

- **Polimorfismo:** esta es la capacidad de un tener varios métodos (varias formas de comportarse) con el mismo nombre, pero se implementan de forma diferente.
- **Herencia:** Esta característica permite a una clase padre heredar sus atributos y métodos a una clase hija.
- **Encapsulamiento:** esto se refiere a que la información de un objeto debe estar encapsulada por su comportamiento.
- **Abstracción:** Tiene dos vertientes, una es la de descripción simplificada de un sistema que solo muestra los detalles significativos de un objeto y la otra vertiente es que se trata de la forma en la que abstraemos objetos del mundo real y lo plasmamos en un programa.
- **Reutilización de código:** En la POO la reutilización de código es algo muy común.



Los Autores: Rodríguez, A. (2020). Programación Orientada a Objetos: Elementos. [Figura 2.12]

2.5.3 Programación Concurrente

“La programación concurrente se define como el área de la ciencia de computación que trata sobre las metodologías, lenguajes, técnicas y herramientas de programación necesarias para la construcción de programas reactivos. Los programas reactivos, frente a los programas transformacionales, interaccionan continuamente con el entorno intercambiando información; por el contrario, los programas transformacionales tienen el objetivo de producir un resultado final a partir de unos datos de entrada” (Suarez Sarmiento, 1996, p.22).

Los programas concurrentes se diferencian de los programas tradicionales en que en estos últimos un programa tiene un conjunto de instrucciones que se ejecutan secuencialmente. Los programas concurrentes vistos de otra forma se pueden definir como un conjunto de programas secuenciales ordinarios que se ejecutan en concurrencia abstracta. Según Suarez Sarmiento, (1996) “La concurrencia es abstracta porque no se requiere que exista un procesador físico separado para cada

proceso. Incluso si el programa es ejecutado compartiendo un solo procesador, la programación concurrente nos puede facilitar el entendimiento de determinados problemas suponiendo que el programa está compuesto por un conjunto de procesos. Los programas transformacionales también se codifican mediante la concurrencia abstracta sincronizando y comunicando procesos que pueden ejecutar subconjuntos o cálculos del programa original”.

2.5.4 Programación Declarativa

La programación declarativa la definimos como un estilo de programación en el cual el programador especifica que debe computarse más que el cómo debe realizarse los cómputos. Este tipo de programación tiene como característica principal el uso de la lógica para el uso de lenguajes de programación. Según Iranzo (2007), este tipo de programar debe de cumplir con ciertos requisitos en la lógica que se emplee a la hora de programar y estos son:

- Disponer de un lenguaje lo suficientemente expresivo.
- Disponer de una semántica operacional.
- Disponer de una semántica declarativa que permita dar un significado a los programas de forma independiente a su posible ejecución.

2.6 Metodologías de Desarrollo de Software

Según Avison y Fitzgerald (1995, p. 70), una metodología es una colección de procedimientos, técnicas, herramientas y documentos auxiliares que ayudan a los desarrolladores de software en sus esfuerzos por implementar nuevos sistemas de información. Una metodología está formada por fases, cada una de las cuales se puede dividir en sub-fases, que guiarán a los desarrolladores

de sistemas a elegir las técnicas más apropiadas en cada momento del proyecto y también a planificarlo, gestionarlo, controlarlo y evaluarlo.

2.6.1 Método Waterfall o Cascada

Según Sommerville (1998, pp. 396 -398) la metodología en cascada es el enfoque metodológico que ordena rigurosamente las etapas del proceso para el desarrollo de software, de tal forma que el inicio de cada etapa debe esperar a la finalización de la etapa anterior. A continuación, las fases de este método:

1. Análisis de requisitos.
2. Diseño del sistema.
3. Diseño del programa.
4. Codificación.
5. Pruebas.
6. Implementación o verificación del programa.
7. Mantenimiento.



Digital Guide, (2019): Metodología en Cascada [Figura 2.13]. Recuperado de <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/el-modelo-en-cascada/>

2.6.2 Método Agile

De acuerdo con Kendall y Kendall (2004, p.20) la metodología ágil es una metodología de desarrollo de software que se basa en valores, principios y prácticas básicas. Los cuatro valores son comunicación, simpleza, retroalimentación y valentía.



Progressa, (2020). Metodología Agile [Figura 2.14]. Recuperado de <https://www.progressalean.com/metodologia-agile/>

2.6.2.1 SCRUM

Schwaber (2004, p. 163) habla de Scrum como un marco de trabajo que define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto.

Los roles principales en Scrum son el Scrum Master, que procura facilitar la aplicación de Scrum y gestionar cambios, el Product Owner (dueño del producto), que representa a los stakeholders (interesados externos o internos), y el Team (equipo) que ejecuta el desarrollo y demás elementos relacionados con él.

Durante cada sprint, un periodo entre una y cuatro semanas (la magnitud es definida por el equipo

y debe ser lo más corta posible), el equipo crea un incremento de software potencialmente entregable (utilizable). El conjunto de características que forma parte de cada sprint viene del Product Backlog, que es un conjunto de requisitos de alto nivel priorizados que definen el trabajo a realizar (PBI, Product Backlog Item). Los elementos del Product Backlog que forman parte del sprint se determinan durante la reunión de Sprint Planning. Durante esta reunión, el Product Owner identifica los elementos del Product Backlog que quiere ver completados y los da a conocer al equipo. Entonces, el equipo conversa con el Product Owner buscando la claridad y magnitud adecuadas (Cumpliendo el INVEST) para luego determinar la cantidad de ese trabajo que puede comprometerse a completar durante el siguiente sprint.



Drew, (2019). Scrum Methodology [Figura 2.15]. Recuperado de <https://blog.wear Drew.co/ventajas-y-desventajas-de-la-metodologia-scrum>

2.6.3 Método Espiral

Boehm (1986, pp. 61–72), autor de diversos artículos de ingeniería del software; modelos de estimación de esfuerzo y tiempo que se consume en hacer productos software; y Modelos de Ciclo

de Vida; ideó y promulgó un modelo desde un enfoque distinto al tradicional en Cascada: El Modelo Evolutivo Espiral. Su Modelo de Ciclo de Vida en Espiral tiene en cuenta fuertemente el riesgo que aparece a la hora de desarrollar software. Para ello, se comienza mirando las posibles alternativas de desarrollo, se opta por la de riesgo más asumible y se hace un ciclo de la espiral. Si el cliente quiere seguir haciendo mejoras en el software, se vuelve a evaluar las distintas nuevas alternativas y riesgos y se realiza otra vuelta de la espiral, así hasta que llegue un momento en el que el producto software desarrollado sea aceptado y no necesite seguir mejorándose con otro nuevo ciclo.



ASP Gems, (2019). Modelo Espiral [Figura 2.16]. Recuperado de <https://aspgems.com/metodologia-de-desarrollo-de-software-iii-modelo-en-espiral/>

2.6.4. Método Prototipo

Kendall y Kendall (2011, p. 158), define la creación de prototipos de sistemas de información como una técnica valiosa para recopilar rápidamente información específica sobre los requerimientos de información de los usuarios. En general, la creación de prototipos efectivos se

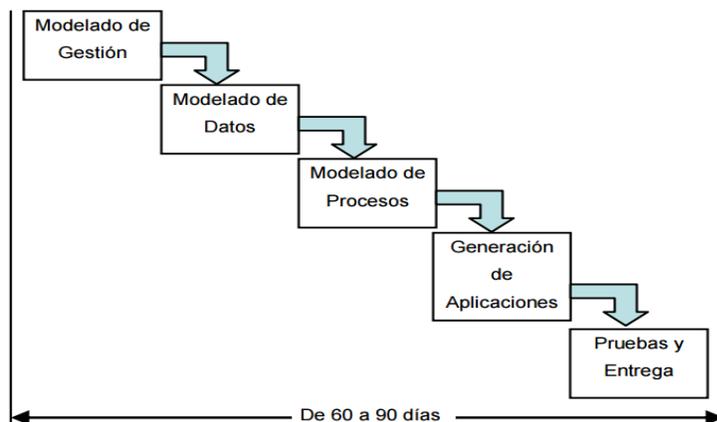
debe llevar a cabo en las primeras etapas del ciclo de vida del software, durante la fase de determinación de requerimientos.



Apuntes IS, (2018). *Prototipos* [Figura 2.17]. Recuperado de <https://alexapuntes44587.wordpress.com/metodo-del-prototipo/>

2.6.5 Método RAD (Rapid Applications Development)

Para Maurer y Martel (2002, pp. 86-91), El desarrollo rápido de aplicaciones o RAD (acrónimo en inglés de rapid application development) es un proceso de desarrollo de software que comprende el desarrollo interactivo, la construcción de prototipos y el uso de utilidades CASE (ingeniería asistida por computadora). Tradicionalmente, el desarrollo rápido de aplicaciones tiende a englobar también la usabilidad, utilidad y la rapidez de ejecución.



Ingeniería de Software, (2017). *RAD* [Figura 2.18]. Recuperado de <https://sisingblog.wordpress.com/2017/04/03/metodologia-rad/>

2.6.7. Método XP (Extreme Programming)

La Programación Extrema, mejor conocida por su nombre en inglés "Extreme Programming" (XP), es una metodología ágil. Fue formulada por Kent Beck, autor del primer libro sobre la materia, "Extreme Programming Explained: Embrace Change".

Somerville (2011 p. 364) nos da Las siguientes bases del método:

1. Desarrollo iterativo e incremental
2. Pruebas unitarias continuas,
3. Programación por parejas de personas
4. Frecuente interacción del equipo de programación con el cliente
5. Corrección de todos los errores antes de añadir nueva funcionalidad.
6. Refactorización del código
7. Propiedad del código compartida
8. Simplicidad en el código



Diego Calvo, (2018). Metodología XP [Figura 2.19]. Recuperado de <https://www.diegocalvo.es/metodologia-xp-programacion-extrema-metodologia-agil/>

2.7 Ventajas de Aprender a Programar

De acuerdo con un artículo de agosto 2020 publicado por la Universidad de Rasmussen en Estados Unidos, entre las ventajas de aprender a programar se encuentran:

1. Trabajos con salario por encima del promedio de acuerdo con las estadísticas del 2019 del Buró de estadísticas Laborales.
2. Gran demanda de programadores y trabajos que requieren conocimiento de programación.
3. Desarrollo de habilidades lógicas y resolución de problemas.
4. Estabilidad y flexibilidad laboral
5. Automatización de tareas en el ámbito laboral.

2.8 Beneficios de Aprender a Programar en educación Media

De acuerdo con un estudio publicado en el 2014 por la Revista de Pedagogía Bordón de la Sociedad Española de Pedagogía (pp. 136-138), entre las ventajas de aprender a programar en niveles de educación media se encuentran:

1. las tareas de programación exigen establecer relaciones formales, analogías, y procesos de inducción y memoria; en la línea del llamado pensamiento fluido
2. es una tarea que exige creatividad
3. demanda poner en juego habilidades en las cinco categorías de pensamiento creativo: pensamiento fluido (o generar gran cantidad de soluciones); flexible (o generar soluciones diferentes a partir de los mismos principios); original (o generar soluciones novedosas y de síntesis); elaborativo (o embellecer las soluciones obtenidas); y evaluativo (o escoger y decidir entre las soluciones en función de las apreciaciones propias y ajenas).
4. Exige altas dosis de metacognición al implicar un continuo control y evaluación de

múltiples procesos cognitivos de orden inferior que se van desplegando “en paralelo”: la programación demanda conjugar simultáneamente las tareas analíticas de escritura de código con la perspectiva sintética del producto final generado, la aplicación, que debe funcionar como un todo integrado.

5. Aprender a programar es adquirir una habilidad emergente y relevante, para los estudiantes, dado que les permite generar un producto final, la aplicación, de creciente uso, utilidad y valoración social, y que potencialmente puede ser puesto a disposición de inmensas audiencias reales y globales.

2.8.1 Creatividad

La creatividad se trata de una capacidad existente en todos los seres humanos, utilizada para la solución de problemas y que precisa de realidades ya existentes. Con base en algunas investigaciones podría definirse la creatividad como “el conjunto de aptitudes vinculadas a la personalidad del ser humano que le permiten, a partir de una información previa, y mediante una serie de procesos internos (cognitivos), en los cuales se transforma dicha información, la solución de problemas con originalidad y eficacia” (Hernández, 1999, p. 48)



Unycos, (2019). Creatividad [Figura 2.20]. Recuperado de <https://unycos.com/blog/conoce-los-distintos-tipos-de-creatividad-que-existen/>

2.8.2 Trabajo Colaborativo o Trabajo en equipo

El diccionario Merriam Webster define el trabajo en equipo es el trabajo hecho por varias personas

donde cada uno hace una parte, pero todos tienen un objetivo en común. Pero para que se considere trabajo en equipo o cooperativo según Guitert y Giménez: el trabajo debe tener una estructura organizativa que favorezca la elaboración conjunta del trabajo y no que cada uno de los miembros realicen una parte del trabajo y juntarlas.

2.8.3 Razonamiento Computacional

Según Wing (2006, p. 33), el pensamiento computacional implica resolver problemas, diseñar sistemas y comprender el comportamiento humano, haciendo uso de los conceptos fundamentales de la informática.

2.8.4 Constructivismo

Para Carretero (2009, p. 1), el constructivismo propone un paradigma donde el proceso de enseñanza se percibe y se lleva a cabo como un proceso dinámico, participativo e interactivo del sujeto, de modo que el conocimiento sea una auténtica construcción operada por la persona que aprende (por el «sujeto cognoscente»). El constructivismo en pedagogía se aplica como concepto didáctico en la enseñanza orientada a la acción.



Edacom, (2019). Enseñanza Constructivista [Figura 2.21]. Recuperado de <https://blog.edacom.mx/que-es-constructivismo>

2.9 Herramientas para enseñar a programar en escuelas

La programación como tal hoy día es una habilidad esencial que muchas personas piensan que debe ser incluida en el pensum de las escuelas, todo niño y adolescente debería aprender a programar software. Pero hay un problema respecto a la enseñanza de esta materia en las escuelas y es que mientras que enseñar programación a los niños es relativamente fácil, cuando se llega a la adolescencia muchos pueden que no tengan interés de aprender, quizás por el método o las herramientas que se usan, por lo que diversas instituciones como Microsoft, Apple y el MIT, han desarrollado un conjunto de herramientas que facilitan el enseñar y aprender a programar.

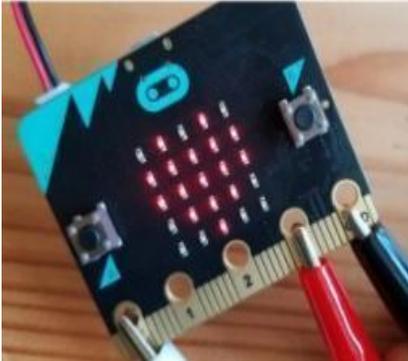
Para que los estudiantes de educación media tengan real interés en aprender a programar se les debe instruir y enseñar a construir algo real, aunque primero deben aprender lo básico con ciertas herramientas.

2.9.1 MakeCode de Microsoft

Microsoft MakeCode es una plataforma open source y gratuita que se utiliza para la creación de experiencias atractivas de aprendizaje de la programación de software, es una buena herramienta para usarse cuando se inicia a un estudiante de escuela en el mundo de la programación de software.

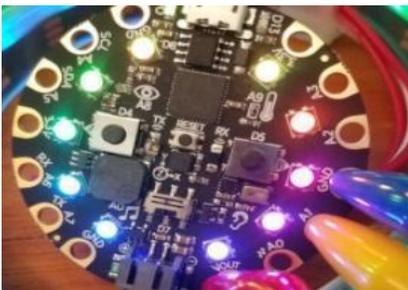
Para poder usar MakeCode se necesita es una computadora con acceso a internet, no se necesita descargar ni instalar nada ya que esta herramienta funciona de manera online, así no hay problemas de compatibilidad por versiones del sistema operativo, MakeCode es gratis y funciona en todos los navegadores y plataformas. Se puede usar los siguientes productos para programar con MakeCode:

- **micro:bit.** El micro:bit es una pequeña computadora programable que tiene una pantalla led, 2 botones, un sensor de luz, antena Bluetooth, un acelerómetro, una brújula y muchos componentes más. El micro:bit tiene todo lo necesario para crear proyectos increíbles e innovadores, desde robots hasta teclados dinámicos.



Microsoft, (2020). Dispositivo Micro:bit de Microsoft. [Figura 2.22]. Recuperado de <https://www.microsoft.com/es-es/makecode>

- **Circuit Playground Express.** Circuit Playground Express es una herramienta que sirve como excelente introducción a la electrónica y la programación. Tiene 10 NeoPixels de varios colores, sensor de movimiento, sensor de temperatura, sensor de luz, micrófono y altavoz. Crea una alarma de sirena, un reloj con cronómetro o una varita mágica con unos cuantos bloques de código, es una herramienta muy versátil para hacer muchos proyectos.



Microsoft, (2020). Dispositivo Circuit Playground Express. [Figura 2.23]. Recuperado de <https://www.microsoft.com/es-es/makecode>

- **Minecraft.** Minecraft es un juego en el que se colocan y rompen bloques, el único límite de lo que se puede hacer en minecraft es la imaginación del jugador. Pero Minecraft

también se usa como una plataforma creativa para enseñar y aprender informática, específicamente para aprender a programar de forma interactiva y viendo los resultados de tu creación en tiempo real y de forma divertida a través de un juego. Se puede crear un programa para invocar una lluvia de cerdos o crear tu propio minijuego para jugar con amigos. Esta herramienta incentiva la creatividad y el compañerismo de una forma muy divertida y atractiva para los estudiantes.



Microsoft, (2020). Minecraft. [Figura 2.24]. Recuperado de <https://www.microsoft.com/es-es/makecode>

- **LEGO MINDSTORMS Education EV3.** Con este producto la tecnología cobra vida, este producto incluye diferentes sensores y motores que se pueden programar. Los alumnos pueden crear robots que pueden recorrer laberintos, clasificar por colores o actuar como un instrumento musical.



Microsoft, (2020). Dispositivo LEGO MINDSTORMS Education EV3. [Figura 2.25]. Recuperado de <https://www.microsoft.com/es-es/makecode>

2.9.2 Swift Playgrounds

Swift Playgrounds es una aplicación revolucionaria creado por la empresa Apple Inc., es una aplicación que se usa en Ipad y Macs, esta aplicación hace que aprender a programar sea más interactivo y divertido, ya que no requiere habilidades o conocimiento de coding o programación lo que la hace perfecta para los estudiantes que están empezando en el mundo de la programación.

Esta aplicación te pone a resolver acertijos para empezar a dominar la lógica básica de programación con el lenguaje real de programación conocido como Swift. Swift es un poderoso lenguaje de programación creado por Apple y utilizado por los profesionales de programación para crear aplicaciones para dispositivos de Apple (iPhone, iPad y Mac). Swift Playgrounds te pone una serie de desafíos y a medida que avanzas en estos desafíos entras a áreas de juego más retadoras.



Apple, (2020). Swift Playgrounds. [Figura 2.26]. Recuperado de <https://www.apple.com/swift/playgrounds/>

Swift PlayGrounds puede ser usado desde un navegador si se va al siguiente link <http://online.swiftplayground.run/>.



Apple, (2020). *Swift Playgrounds: Página de Práctica*. [Figura 2.27]. Recuperado de <http://online.swiftplayground.run/>

2.10 PSeint

PSeint (Pseudo Intérprete), es una herramienta educativa que fue creada en Argentina, los usuarios de esta aplicación son los estudiantes que requiera aprender los fundamentos de la programación y el desarrollo de la lógica de programación. Este software tiene un alto nivel de popularidad en países latinoamericanos y España. Básicamente lo que hace es utilizar el pseudocódigo para la creación de algoritmos. (PSeint, 2020. Documentación Oficial de PSeint. Recuperado de <http://pseint.sourceforge.net/index.php?page=documentacion.php>)

Esta herramienta es más para asistir al estudiante en sus primeros pasos en el fascinante mundo de la programación, por lo que es más que perfecto para los estudiantes de nivel medio. No es tan interactivo como MakeCode o Swift PlayGrounds, pero este software es lo más cercano a la programación real que se hace con lenguajes como C#, Java, C, C++, etc. PSeint, mediante un simple e intuitivo pseudo-lenguaje en español y con un editor de diagramas de flujo, le permite centrar su atención en los conceptos fundamentales de la algoritmia computacional, lo que reduce las dificultades propias de un lenguaje de programación real, además proporciona un entorno de trabajo con gran cantidad de ayuda disponible y muchos recursos didácticos. (PSeint, 2020. Documentación Oficial de PSeint. Recuperado de <http://pseint.sourceforge.net/index.php?page=documentacion.php>)

Características de PSeint

- Lenguaje Autocompletado
- Ayudas Emergentes
- Plantillas de Comandos
- Soporta procedimientos y funciones
- Indentación Inteligente
- Exportación a otros lenguajes (C, C++, C#, Java, PHP, JavaScript, Visual Basic .NET, Python, Matlab)
- Graficado, creación y edición de diagramas de flujo
- Editor con coloreado de sintaxis
- Software multiplataforma sobre Microsoft Windows, GNU/Linux y Mac OS X, en diciembre de 2016 empezó un desarrollo independiente para Android.

2.11 Python

Python es un lenguaje de programación interpretado muy famoso y utilizado debido a lo fácil que es aprender a programar con Python, esto es debido a su filosofía que se enfoca en la legibilidad del código. Es un lenguaje de programación multiparadigma porque se puede soportar programación orientada a objetos, programación imperativa y programación funcional. Es un lenguaje interpretado, dinámico y lo mejor de todo es multiplataforma, así que puede usarse y puede ejecutarse en casi cualquier sistema operativo. Este lenguaje es administrado por la Python Software Foundation. Posee una licencia de código abierto llamada Python Software Foundation License. (Python, 2020. Documentación Oficial de Python. Recuperado de <https://docs.python.org/3/>)

Python tiene como característica principal la legibilidad del código y que es muy fácil aprender a programar con este lenguaje, lo que hace de este lenguaje perfecto para aprender a programar en las escuelas de nivel medio. Además de que existen muchos proyectos que se pueden hacer con Python, proyectos muy dinámicos como crear un videojuego de forma sencilla usando uno que otro framework que facilite la creación de este lenguaje. Python es una muy buena alternativa para optar por este lenguaje en el mundo de la programación, ya que es muy fácil, su sintaxis es fácil de entender porque se asemeja mucho a lo que es el lenguaje natural que usamos para hablar, además de que los programas hechos en Python parecen más pseudocódigos, lo cual brinda una gran ayuda en su mantenimiento. (Python, 2020. Documentación Oficial de Python. Recuperado de <https://docs.python.org/3/>)

El código en Python es más organizado evitando el uso de signo de punto y coma al final de cada sentencia y también evita el uso de llaves para especificar que dicho bloque de código pertenece a una sentencia. Para esto Python hace uso de la indentación, la cual está representada por un bloque de 4 espacios, teniendo así un código legible. (Python, 2020. Documentación Oficial de Python. Recuperado de <https://docs.python.org/3/>)

Python es un lenguaje de código abierto y eso nos permite crear nuestras aplicaciones e involucrarnos en su desarrollo, tiene tipado dinámico, esto se refiere a que su tipo de dato se determinará en el tiempo de ejecución, según el valor que se le asigne a una variable. (Python, 2020. Documentación Oficial de Python. Recuperado de <https://docs.python.org/3/>)

¿Por qué Python para aprender a programar en escuelas?

Porque programar te permite aprender haciendo cosas, se vuelve algo muy significativo y divertido, te da muchas herramientas y oportunidades de aprendizaje, te permite tener el control de la computadora y decirle qué hacer. Podrías crear un videojuego, una aventura multimedia, etc.

2.12 Conclusión Capítulo II:

En este capítulo definimos los conceptos clave que forman parte de nuestro marco teórico los cuales ayudarán a entender mejor la propuesta planteada.

Pudimos identificar diversos lenguajes de programación candidatos, así como también paradigmas

y metodologías de desarrollo cada una ofreciendo ciertas ventajas.

Afirmamos los beneficios de aprender a programar, de manera específica en el nivel medio, lo cual puede llevar a mejores oportunidades de trabajo, desarrollar la creatividad, el trabajo en equipo, razonamiento computacional y constructivismo, dichos conceptos claves fueron definidos de manera breve.

Vimos algunas de las herramientas más populares para aprender a programar en escuelas y por último el impacto de la programación tanto en la actualidad como a futuro en distintas áreas de la sociedad.

CAPÍTULO III: SEGURIDAD DE LOS SISTEMAS DE INFORMACIÓN

3.1 Introducción

En este capítulo trataremos conceptos claves sobre la seguridad de sistemas de información. Debido a la naturaleza de este proyecto, es de suma importancia que los sistemas de información a utilizarse estén protegidos ante ciberataques, así como también establecer una estrategia ante pérdidas de datos y una política de riesgos.

A continuación, se desglosan conceptos claves sobre:

- Sistemas de información
- Seguridad de sistemas de información
- Manejo, clasificación y almacenamiento de información
- Gestión de riesgos
- Algunos de los principales mecanismos de ataques a los que se ven expuestos los sistemas de información en la actualidad
- Algunos de los principales mecanismos de protección existentes ante dichas amenazas.

Es importante el manejo de estos temas, debido al tipo de usuario al que va destinado esta solución quizá no tenga el conocimiento y/o experiencia adecuada en temas de seguridad informática y medidas preventivas, y debido al manejo de la institución de información relacionada a menores de edad.

3.2 Definición de Sistemas de Información

De acuerdo con González, F. (2012, p. 1), un sistema de información es un medio tecnológicamente implementado para el registro, almacenando, y diseminando expresiones lingüísticas como también para dibujar conclusiones a tales expresiones.

Un sistema de información es aquel que permite recopilar, administrar y manipular un conjunto de datos que conforman la información necesaria para que los estamentos ejecutivos de una organización puedan tomar decisiones de manera informada.

3.3 Seguridad de los Sistemas de Información

Según Roa, J. (2013, p. 24): Nuestra sociedad se basa en información procesada digitalmente: hay que protegerla porque la información es poder. La seguridad de sistemas de información comprende:

- Equipos: robo, evitar intentos de conectar equipos externos a la empresa, mantenimiento preventivo.
- Aplicaciones: limitar privilegios, revisar vulnerabilidades conocidas, descargar aplicaciones de fuentes fiables.
- Datos: almacenamiento redundante, copias de seguridad y cifrado.
- Comunicaciones: utilizaremos canales cifrados para proteger la información cuando es transmitida por nuestra red interna y, sobre todo, por Internet.



SGSI, (2020). Seguridad de la Información. [Figura 3.1]. Recuperado de <https://www.pmg-ssi.com/2020/01/la-gestion-de-riesgos-en-los-sistemas-de-gestion-de-seguridad-de-la-informacion/>

3.4 Manejo y Tratamiento Seguro de la Información

Según un artículo sobre clasificación de datos publicado en el 2019 por la Organización de Estados Americanos y Amazon Web Services (2019), La implementación de la gestión de la información en general, y la clasificación de datos en particular, varía según el tipo de organización e incluso puede ser diferente según cada organización. Sin embargo, existen ciertos principios fundamentales comunes entre los gobiernos, las organizaciones no gubernamentales y las organizaciones comerciales.

De acuerdo con el artículo mencionado, estos principios son:

- **Apertura, transparencia y valores sociales:** La clasificación debe usarse con precaución y de acuerdo con la sensibilidad, el valor y la criticidad de los datos. Las restricciones de acceso solo deben elegirse para los casos en que la divulgación de información sea perjudicial para los intereses legítimos y las obligaciones legales de la propia organización, su personal o terceros. En tales casos, deben observarse estrictamente los procedimientos especificados, para garantizar que la información no se vea comprometida ya sea a propósito o inadvertidamente. El desafío será no clasificar en exceso por efectos de conveniencia o practicidad, que resultaría en detrimento de la transparencia y la confianza

pública, y así privaría a los interesados de ser los dueños de sus propias decisiones de gestión de riesgos.

- **Enfoque basado en el contenido y tecnológicamente neutral:** La información debe clasificarse en función de su contenido y los riesgos asociados con el compromiso del contenido, independientemente de su formato, medios u origen. No se debe discriminar basado en el formato o los medios de la información, ya sea analógica (en papel) o digital; almacenado en un sistema de información, en medios de almacenamiento, en dispositivos móviles o en la nube. Del mismo modo, la decisión de clasificar la información debe depender del contenido en sí y no necesariamente derivarse automáticamente de la fuente de la información en la que se basa, a la que responde o hace referencia). Por ejemplo, confiar en fuentes públicas no debería determinar automáticamente que la información agregada debería revelarse públicamente.
- **Enfoque de gestión de riesgos:** Se debe brindar protección a la información de acuerdo con el nivel de sensibilidad, valor y criticidad de esta. La protección generalmente se realiza con un enfoque graduado, basado en niveles correspondientes al valor y el riesgo. Un nivel de protección circunscribe el conjunto de medidas para reducir los riesgos a un nivel aceptable, es decir, la posible gravedad y probabilidad de que la información se vea comprometida. Al determinar el nivel de sensibilidad y el valor de la información, se deben tener en cuenta tanto el grado de daño potencial de compromiso (divulgación no autorizada, modificación o pérdida) como el valor potencial de los datos.
- **Proporcionalidad:** La información se clasificará para un nivel apropiado que debe ser lo más bajo posible, pero tan alto como sea necesario.

- **Papeles y responsabilidades claras:** Con respecto a la clasificación de datos, la política y los procesos deben asignarse para lograr la seguridad de la información dentro de la organización y confirmarse por el compromiso con y la conciencia de la seguridad de la información que tiene la gerencia.
- **Enfoque del ciclo de vida:** Como parte de un sistema de gestión de la información, el sistema de clasificación debe tener en cuenta la información durante todo su ciclo de vida: desde la creación o recepción, almacenamiento, recuperación, modificaciones, transferencia, copia y transmisión hasta la destrucción. Además, la política de gestión de información/procesamiento de datos de una organización no debe estar escrita en piedra, sino evaluarse regularmente para garantizar que se corresponde con las necesidades y expectativas de la organización.

3.4.1 Como se clasifica la información

La Guía de almacenamiento seguro de la información publicada en el 2016 por el Instituto Nacional de Ciberseguridad de España (p. 8), presenta un modelo de clasificación de la información basado en la norma ISO 27001, la cual divide la información en tres niveles. De acuerdo con esta guía, Para la clasificación se pueden considerar, además de su antigüedad y valor estratégico, las tres propiedades: confidencialidad, integridad y disponibilidad. Lo más usual es clasificar la información teniendo en cuenta solamente una de estas tres dimensiones, la confidencialidad. Se clasifica la información en tres niveles: confidencial, de uso interno e información pública. Esta aproximación es la más aceptada, pues uno de los riesgos más críticos para cualquier negocio es la fuga de información que no es más que una pérdida de la confidencialidad de esta.

CATEGORÍA	DEFINICIÓN	TRATAMIENTO
Confidencial	<ul style="list-style-type: none"> • Información especialmente sensible para la organización. • Su acceso está restringido únicamente a la Dirección y a aquellos empleados que necesiten conocerla para desempeñar sus funciones. • Se incluye la información que contenga datos de carácter personal de nivel alto. 	<ul style="list-style-type: none"> • Esta información debe marcarse adecuadamente. • Se deben implementar todos los controles necesarios para limitar el acceso a la misma únicamente a aquellos empleados que necesiten conocerla. • En caso de sacarla de las instalaciones de la empresa en formato digital, debe cifrarse. • Para los datos de nivel alto, se deben cumplir también las medidas de seguridad indicadas en el Reglamento de Desarrollo de la LOPD.
Interna	<ul style="list-style-type: none"> • Información propia de la empresa, accesible para todos los empleados. • Por ejemplo, la política de seguridad de la compañía, el directorio de personal u otra información accesible en la intranet corporativa. 	<ul style="list-style-type: none"> • Esta información debe marcarse adecuadamente y estar accesible para todo el personal. • No debe difundirse a terceros salvo autorización expresa de la dirección de la empresa.
Pública	<ul style="list-style-type: none"> • Cualquier material de la empresa sin restricciones de difusión. • Por ejemplo, información publicada en la página web o materiales comerciales. 	<ul style="list-style-type: none"> • Esta información no está sujeta a ningún tipo de tratamiento especial.

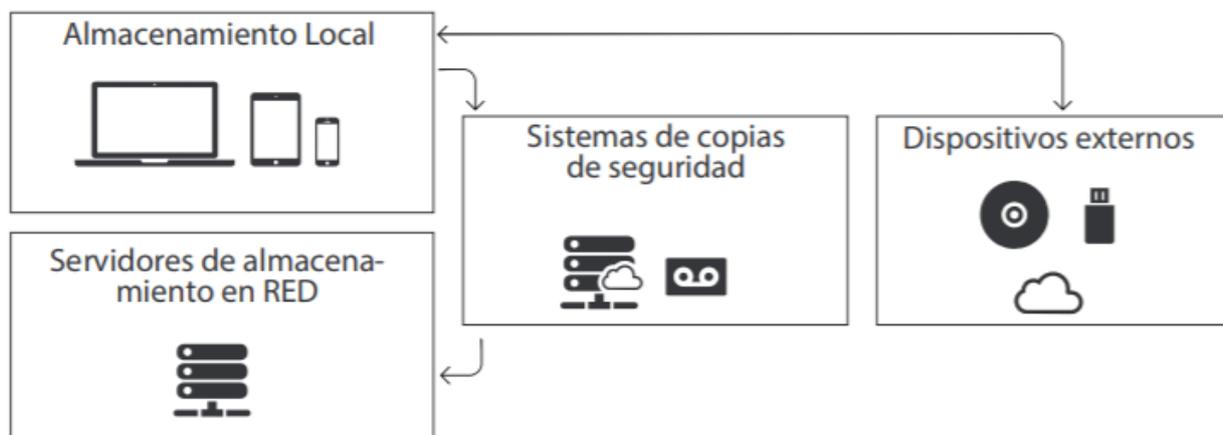
Guía de almacenamiento seguro de la información (Instituto Nacional de Ciberseguridad de España, 2016). Clasificación de la información [Tabla 3.1]. Recuperado de

https://www.incibe.es/sites/default/files/contenidos/guias/doc/guia_ciberseguridad_almacenamiento_seguro_metad.pdf

3.4.2 Como se almacena la información

La Guía de almacenamiento seguro de la información publicada en el 2016 por el Instituto Nacional de Ciberseguridad de España (pp. 11-12) define de manera sencilla los diferentes métodos de almacenamiento utilizados en los sistemas de información

En la siguiente ilustración se identifican los diferentes sistemas de almacenamiento de información



Guía de almacenamiento seguro de la información (Instituto Nacional de Ciberseguridad de España, 2016). Sistemas de Almacenamiento de información [Figura 3.2]. Recuperado de https://www.incibe.es/sites/default/files/contenidos/guias/doc/guia_ciberseguridad_almacenamiento_seguro_metad.pdf

A continuación, una breve definición de cada uno de estos sistemas de acuerdo con la Guía de almacenamiento seguro de la información:

- **Almacenamiento local:** Los usuarios utilizan equipos informáticos para realizar sus actividades. La información se genera en estos equipos y desde ellos se modifica y transmite. Cada uno de estos equipos dispone de un sistema de almacenamiento local, normalmente discos duros donde se guarda la información. También es almacenamiento local el utilizado en tabletas y dispositivos móviles interno o en tarjetas de memoria (microSD).
- **Servidores de almacenamiento en red:** Para poder disponer de un lugar común de trabajo donde almacenar el resultado de los trabajos individuales y poder compartir información entre los diferentes usuarios de la Institución se dispone de servidores de almacenamiento en red.
- **Dispositivos externos:** Adicionalmente se puede disponer de sistemas externos que,

conectados directamente a los equipos, permiten un almacenamiento extra de la información, evitando que se ocupe este espacio en el equipo. Estos pueden ser cintas magnéticas, discos duros externos, CD o DVD o pendrives conectados a través de distintos interfaces físicos. Existen también dispositivos externos que se pueden conectar de forma inalámbrica. Los distintos tipos de interfaz tienen también distinta velocidad de transferencia. Por su portabilidad es fácil que se puedan extraviar.

- **Sistema de copias de seguridad:** Es muy recomendable establecer un procedimiento para sistematizar la realización de copias de respaldo de la información generada en la empresa, en soportes externos o en otra ubicación. Algunas Instituciones optarán por sistemas de respaldo de toda o parte de su sistema de información, de manera que puedan continuar su actividad en caso de desastre.
- **Servicios de almacenamiento en la nube:** Es posible utilizar servicios de almacenamiento en la nube como medio de almacenamiento externo, para compartir la información generada o para realizar copias de seguridad. Un caso particular de almacenamiento es el asociado a la contratación de servicios externos como servicios de backup, alojamiento web o las tiendas online. Al contratar servicios en la nube se deben seguir los mismos criterios de seguridad de la institución para la información asociada a los servicios contratados, reflejándolo en los Acuerdos de Nivel de Servicio que se firmen con los proveedores.

3.4.3 Como se Respalda la información

Según Guía de almacenamiento seguro de la información del Instituto Nacional de Ciberseguridad de España (2016, p 22-23) una copia de seguridad, también conocida como backup, es un

duplicado que se realiza sobre ficheros o aplicaciones contenidas en un ordenador con la finalidad de recuperar los datos en el caso de que el sistema de información sufra daños o pérdidas accidentales de los datos almacenados.

Todo plan de contingencia de una empresa requiere contar con una planificación adecuada de las copias de seguridad que se realizan, ya que la pérdida de datos puede poner en peligro la continuidad del negocio.

Algunos de los requisitos que debe cumplir la planificación de copias de seguridad de acuerdo con esta guía son:

- **Identificar los datos que requieren ser preservados.** Son aquellos cuya pérdida afectaría a la continuidad de labores.
- **Establecer la frecuencia con la que se van a realizar los procesos de copia.** Esta frecuencia influye en la cantidad de información que se puede perder con respecto a la fuente original. Este parámetro es de suma importancia y requiere de un análisis exhaustivo.
- **Disponer el almacén físico para las copias.** Este almacén se determina en función de la seguridad que requiere la información entre almacenes en el mismo edificio o remotos en edificios externos.
- **Buscar una probabilidad de error mínima.** Asegurándose de que los datos son copiados íntegramente del original y en unos soportes fiables y en buen estado.
- No se deben utilizar soportes que estén cerca de cumplir su vida útil para evitar que fallen cuando vaya a recuperarse la información que contienen.
- Controlar los soportes que contienen las copias, guardándolos en un lugar seguro y restringiendo su acceso sólo a las personas autorizadas.

- **Planificar la restauración de las copias:**
 - Formando a los técnicos encargados de realizarlas
 - Disponiendo de soportes para restaurar la copia, diferentes de los de producción
 - Estableciendo los medios para disponer de dicha copia en el menor tiempo posible
- Probar el sistema de forma exhaustiva para comprobar su correcta planificación y la eficacia de los medios dispuestos.
- Definir la vigencia de las copias, estableciendo un periodo en el que dicha copia deja de tener validez y puede sustituirse por una copia más actualizada de la información.
- Controlar la obsolescencia de los dispositivos de almacenamiento. Para el caso de aquellas copias que almacenan información histórica de la organización,
- Cuando se desechen los soportes de almacenamiento, porque hayan llegado al límite de vida útil fijado en la política de copias de seguridad, es importante realizar un proceso de borrado seguro o destrucción para asegurar que la información que contiene no podrá ser recuperada posteriormente.

3.5 Gestión de Riesgos de la Información

La plataforma ISO Tools (2019) en su artículo sobre Seguridad de Información algunos puntos clave sobre La norma ISO/IEC 27001, la cual proporciona los requisitos para el establecimiento, implementación mantenimiento y mejora continua de un sistema de gestión de seguridad de la información (SGSI), el cual debería ser una decisión estratégica para una organización, y desplegar todas sus capacidades para promover el crecimiento y la consolidación de una propuesta de valor.

La organización que desee implantar un Sistema de Gestión de Seguridad de la Información

(SGSI) según la norma ISO 27001 deberá llevar a cabo una evaluación de los riesgos de la seguridad de la información para poder establecer controles que aseguren un entorno invariable bajo los criterios de disponibilidad, confidencialidad e integridad.

Para conseguir dicho objetivo, habrá que definir el alcance del sistema de implementado o a implementar. Una vez que este se conoce habrá que establecer cuáles son los activos de información que afectan a la institución.

Se considera un “activo de la información” a todo aquello que las entidades consideran importante o de alta validez para la misma, ya que puede contener información importante como lo puede ser bases de datos con usuarios, contraseñas, números de cuentas, etc. Los activos de la información deberán ser valorados con el objetivo de conocer cuál es su impacto dentro de la organización. Para conocer ese impacto se utiliza la herramienta de análisis de riesgos. Este procedimiento implica la identificación de las amenazas, vulnerabilidades y riesgos de la información.

De acuerdo con ISO Tools (2019) se puede definir el riesgo como la probabilidad de sufrir daños o pérdidas. Este puede ser humano o no. Por su parte, la amenaza se trataría más bien de un componente del riesgo provocado por una acción que puede ofrecer un resultado inesperado o no deseado. Los impactos provocados por una determinada amenaza pueden provocar múltiples daños: pérdida de ingresos o clientes, pérdida de diferenciación de mercado, costos de respuesta y recuperación por el incidente y el costo de pagar multas o sanciones regulatoras, etc.

En el ámbito de la seguridad de la información se usa lo que se conoce como el plan de tratamiento de riesgos de seguridad de la información. Este se trata del proceso mediante el cual se identifica,

comprende, evalúa, y mitiga cualquier tipo de riesgo o amenaza en la información de una determinada organización. Este aporta dos aspectos clave a las instituciones en las que la información es clave para sus operaciones según ISO Tools:

- Identificar todos los activos críticos de información. Un programa de gestión de riesgos que se puede ampliar para identificar a las personas críticas, los procesos de negocio y la tecnología.
- Comprender por qué los activos críticos escogidos son necesarios para la operación, la realización de la misión y la continuidad de las operaciones.

El plan de tratamiento de riesgos de seguridad de la información debe de estar apoyado por una serie de elementos que incluyan de acuerdo a ISO Tools:

- Un programa de gestión de activos
- El programa de gestión de la configuración
- Un programa de gestión del cambio

ISO 27001, de acuerdo con ISO Tools nos dice que Un plan de tratamiento de riesgos de seguridad de la información adecuado debe de tener en cuenta varios componentes clave para evaluar todo riesgo. Estos son los siguientes:

- Agentes de amenazas, que puede provocarnos determinados riesgos. Habrá que identificarlos perfectamente para conocer sus características y aplicar los controles oportunos.
- Vulnerabilidad. Antes esta solo se debe de tener una cosa clara: Hay que eliminarlo cuanto antes. En el caso de que no pueda ser eliminada habrá que tomar tres acciones: O se reduce al máximo la probabilidad de explotación de la vulnerabilidad; o se intenta reducir la

gravedad del impacto resultante de la explotación de la vulnerabilidad, o bien, no se hace nada y se acepta el riesgo.

- Los resultados esperados por cada vulnerabilidad. Estos dependerán de los impactos que se activen en cada vulnerabilidad.
- Impacto o impactos provocados por una determinada vulnerabilidad. Estos pueden ser tantos como dicha vulnerabilidad genere y pueden afectar a diversos factores dentro de la organización.
- Los activos de la información. Se trata del objetivo fundamental a proteger por todo plan de seguridad de la información. Habrá que intentar evitar a toda costa que estos activos sufran cualquier tipo de impacto negativo que pueda suponer una situación de compromiso para estos.



Startup (2019), Gestión de Riesgo, [Figura 3.3]. Recuperado de <https://www.startuptraining.com/curso/analisis-y-gestion-de-riesgos-en-seguridad-de-la-informacion/>

3.6 Firewalls

De acuerdo con Navarro, Guillermo (2012 p. 9) un Firewall o cortafuegos es un sistema de red

encargado de separar redes informáticas, efectuando un control del tráfico que transcurre entre ellas. Este control consiste, en última instancia, en permitir o denegar el paso de la comunicación de una red a otra mediante el control de los protocolos de red.



Tecnología + Informática (2020), Firewall, [Figura 3.4]. Recuperado de <https://www.tecnologia-informatica.com/que-es-firewall-como-funciona-tipos-firewall/>

3.6.1 Tipos de Firewalls

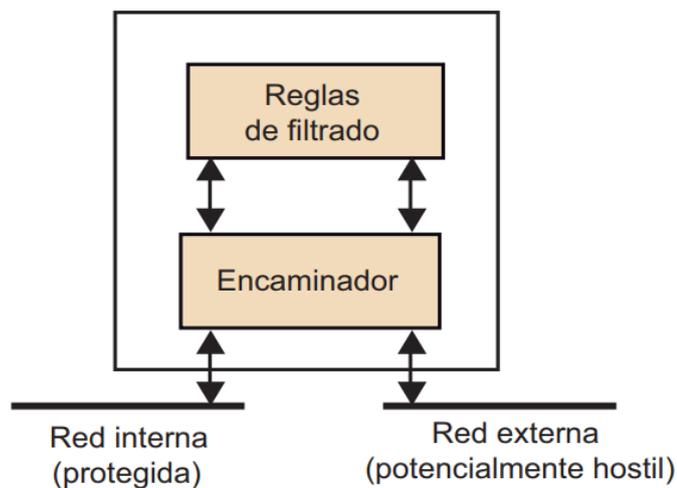
De acuerdo con Navarro, Guillermo (2012 pp. 12-16) Tres de las tecnologías más utilizadas a la hora de construir sistemas firewalls son las siguientes:

- Encaminadores con filtrado de paquetes.
- Pasarelas a nivel de circuito.
- Pasarelas a nivel de aplicación.

Estos tipos de firewalls a veces reciben el nombre de cortafuegos de primera, firewalls de segunda y firewalls de tercera generación respectivamente, debido al orden en el que han aparecido a lo largo de la historia. De manera muy genérica, su diferencia radica en el nivel al

que realizan el filtrado, es decir, a la capa de red sobre la que actúan. Asimismo, hoy en día existen sistemas de firewalls que pueden combinar el filtrado de paquetes a múltiples capas. Este tipo de filtrado multicapa recibe el nombre de stateful multi layer inspection, a continuación, algunos de estos:

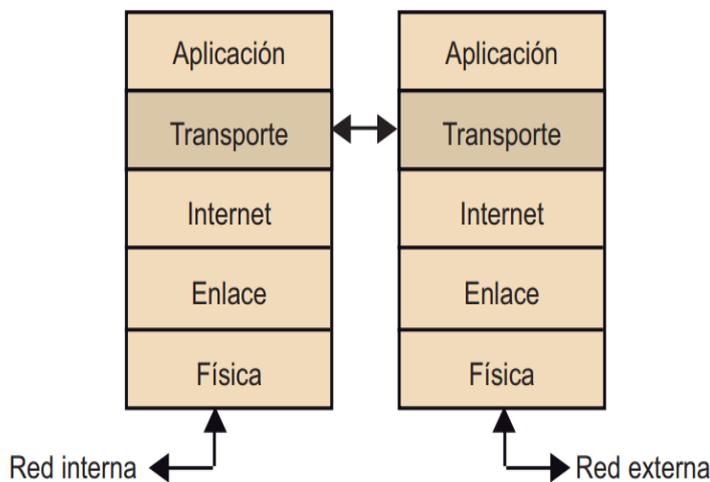
- **Encaminadores con filtrado de paquetes:** Las reglas de filtrado se encargan de determinar si a un paquete le está permitido pasar de la parte interna de la red a la parte externa, y viceversa.



Navarro, Guillermo. Encaminador con filtrado de paquetes [Figura 3.5]. Recuperado de

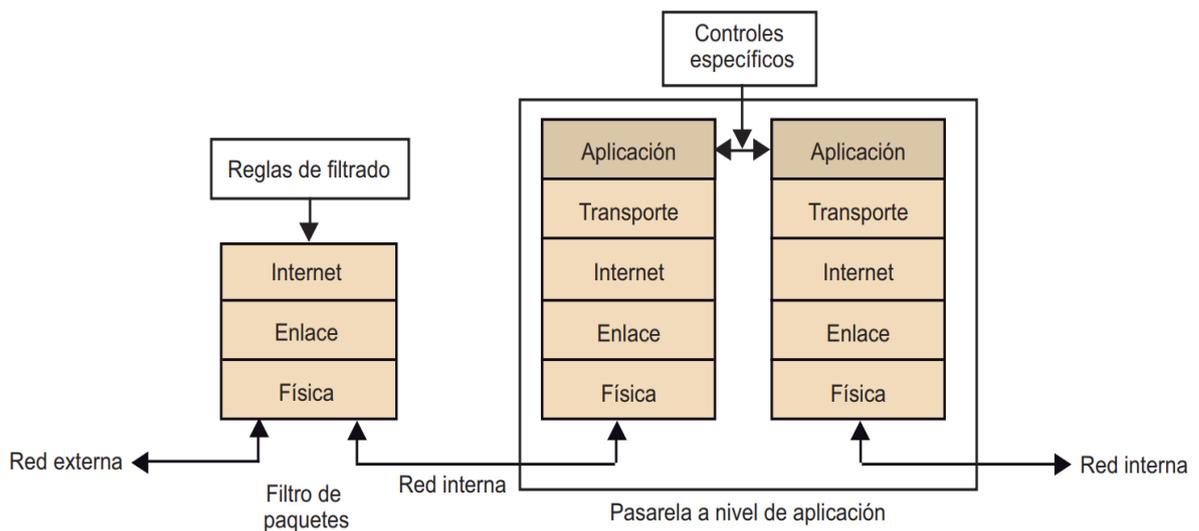
http://openaccess.uoc.edu/webapps/o2/bitstream/10609/62865/6/Seguridad%20en%20redes_M%C3%B3dulo%201_Sistemas%20de%20cortafuegos.pdf. 2012

- **Pasarelas a nivel de circuito:** Una pasarela a nivel de circuito es un dispositivo que hace de pasarela a nivel de capa de transporte entre dos extremos. Establece una conexión con cada uno retransmitiendo los datos entre las dos conexiones.



Navarro, Guillermo. Pasarela a nivel de circuito [Figura 3.6]. Recuperado de http://openaccess.uoc.edu/webapps/o2/bitstream/10609/62865/6/Seguridad%20en%20redes_M%C3%B3dulo%201_Sistemas%20de%20cortafuegos.pdf. 2012

- Pasarelas a nivel de aplicación o servidor proxy:** El servicio proxy se encargará de realizar las conexiones solicitadas con el exterior y, cuando reciba una respuesta, se encargará de retransmitirla al equipo que había iniciado la conexión. Así, el servicio proxy ejecutado en la pasarela aplicara las normas para decidir si se acepta o se rechaza una petición de conexión.



Navarro, Guillermo. Pasarela a nivel de aplicación o servidor proxy: [Figura 3.7]. Recuperado de http://openaccess.uoc.edu/webapps/o2/bitstream/10609/62865/6/Seguridad%20en%20redes_M%C3%B3dulo%201_Sistemas%20de%20cortafuegos.pdf. 2012

3.7 Virus Informático

Cuando de seguridad informática se refiere uno de los temas más hablados son los famosos virus informáticos. Como definición general según la Real Academia Española “Programa introducido subrepticamente en la memoria de un ordenador que, al activarse, destruye total o parcialmente la información almacenada”.

3.7.1 ¿Qué es un virus informático?

Un virus informático es fácilmente definido como un programa informático que tiene la capacidad de copiarse a sí mismo automáticamente en un ordenador sin que el usuario de dicho computador lo sepa y pueden venir de distintas fuentes como internet (la más común) y también por medios de almacenamientos. El objetivo de todo virus es el de alterar o cambiar el funcionamiento normal de un ordenador, todo va a depender del objetivo que tenga la persona al insertar un virus en nuestro ordenador, esto puede ser obtener información personal, o también para tomar control desapercibido de un ordenador para usarlo como ordenador “zombi”.

Dependiendo del objetivo los virus pueden ser peligrosos o no, puede que el virus solo dañe funcionalidades mínimas de un ordenador, o puede que arruine todo el hardware como también puede tomar control total de un computador con la meta de tener la información del usuario.

Existen 3 características que debe de tener un programa para poder considerarse un virus informático, entre estas decimos que un virus debe de ser perjudicial, también autorreproducible y subrepticio.

- **Perjudicial:** Generalmente todos los virus son perjudiciales de alguna forma, hay virus que borran archivos, ocultan archivos e incluso hay virus que modifican la información de los archivos.

- **Autorreproducible:** Un virus debe si o si tener la capacidad de autorreplicarse en un ordenador, para poder expandir el rango de daño que requiera hacer. Podemos decir que esta característica es la que diferencia un virus informático de cualquier otro programa informático.
- **Subrepticio:** Esta también es una característica importante, ya que es la que permite que un virus pueda ocultarse de forma que puede hacer daño y que el usuario no lo sepa hasta que ya es demasiado tarde.

3.7.2 Tipos de Virus Informático

Hay varias categorías de virus informáticos, y generalmente se clasifican según su función , nivel de daño, de qué forma llegan a un computador, que tan efectivo es su forma de esconderse, etc.

- **Virus que afectan el arranque o boot:** Estos virus son muy peligroso porque generalmente se usan para afectar el hardware. Estos se caracterizan porque estos no afectan archivos, más bien suelen afectar directamente a la unidad de almacenamiento que contienen esos archivos, específicamente el sector de arranque de la unidad de almacenamiento. En otras palabras, si una computadora arranca con un virus de este tipo, este último infectará todas las unidades de almacenamiento del sistema.
- **Virus “FAT”:** este es un tipo de virus que ataca el FAT de un sistema operativo, un FAT es la tabla de asignación de ficheros, que es la tabla que une la información de los discos duros, por lo que si un virus ataca esta tabla se impide el acceso a varios archivos o carpetas importantes para el funcionamiento del sistema operativo.
- **Virus Residentes:** estos son virus bastante incómodos porque se ocultan en las RAM de la computadora, de forma tal que son capaces de manipular las operaciones de un sistema operativo a nivel de memoria, lo que le permitiría a este virus infectar todos los archivos del sistema operativo.

- **Virus de Sobreescritura:** estos son muy molestos ya que reescriben un archivo de forma que estos pasan a ser totalmente inservibles, la única forma de deshacerse de este es eliminando por completo el archivo infectado.

Existen varios tipos de virus, pero en esencia todos funcionan de forma parecida, se autorreplican, se ocultan y hacen daño al hardware o software.

3.7.3 Protección contra virus informáticos

Una de las formas más comunes de protegerse de un virus es usando antivirus que puedan proteger nuestro sistema de estas amenazas, hay varios antivirus famosos, como Avast, McAfee y el mismo Windows Defender. Pero en realidad la mejor manera de protegerse debe ser estar informado sobre las amenazas informáticas, el combinar estos conocimientos con técnicas y software de protección es lo mejor para protegerse.

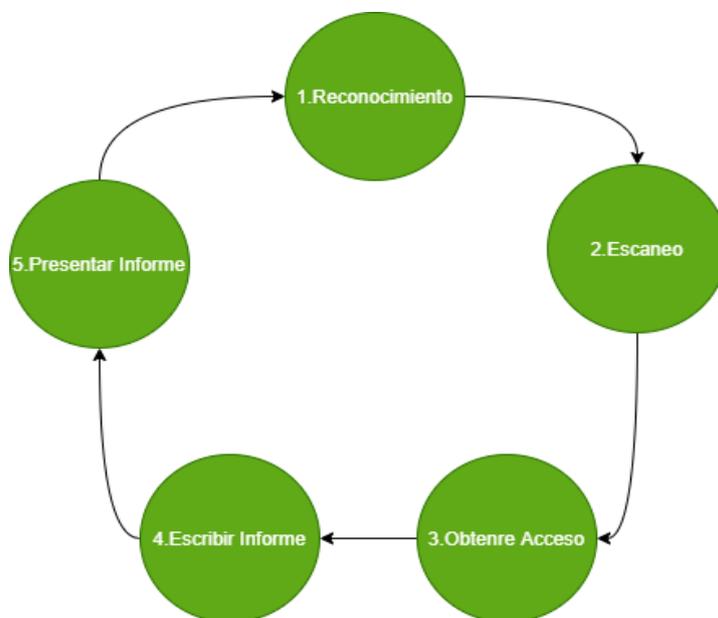
Una recomendación para cuidarse de los virus es tener una carpeta designada para todas las descargas de internet, y que sea una carpeta que nuestro antivirus analice constantemente para prevenir que los virus que se puedan descargar se esparzan en nuestro sistema. Otra cosa es que se debe hacer que nuestra secuencia de booteo siempre inicie con nuestro disco duro o cualquier unidad de almacenamiento, lo que evitará que se introduzca un virus a través del lector de CD o DVD.

3.8 Hacking Ético

Según Astudillo B., K. (2013), cuando hablamos de hacking ético nos referimos a la acción de efectuar pruebas de intrusión controladas sobre sistemas informáticos; es decir que el consultor o

pentester, actuará desde el punto de vista de un cracker, para tratar de encontrar vulnerabilidades en los equipos auditados que puedan ser explotadas, brindándole - en algunos casos - acceso al sistema afectado inclusive; pero siempre en un ambiente supervisado, en el que no se ponga en riesgo la operatividad de los servicios informáticos de la organización cliente. Es importante enfatizar que aunque es indudable que el pentester debe poseer conocimientos sólidos sobre tecnología para poder efectuar un hacking ético, saber de informática no es suficiente para ejecutar con éxito una auditoría de este tipo. Se requiere además seguir una metodología que nos permita llevar un orden en nuestro trabajo para optimizar nuestro tiempo en la fase de explotación, además de aplicar nuestro sentido común y experiencia. Y aunque lamentablemente la experiencia y el sentido común no se pueden transferir en un libro, haré mi mejor esfuerzo por transmitirles la metodología y las buenas prácticas que he adquirido a lo largo de los años de ejercer la profesión de auditoría de seguridad informática.

De acuerdo con Astudillo B., K. (2013), el hacking ético está dividido en varias fases, como se describe en la siguiente imagen:



Astudillo, K. (2013). *Fases del Hacking Ético* p.11. [Figura 3.8]

3.8.1 Hackers

Según Rodríguez, J. (2011), los hackers tienen diversos motivos para hacer lo que hacen, ese motivo puede depender de la personalidad o ambiente, a continuación, las razones que motivan a los hackers:

- Reconocimiento
- Retos
- Propósitos Personales
- Sistemas Políticos
- Diversión

Sin importar el motivo que impulse a los hackers a entrar en un sistema y hacer lo que quieran con él, entrar sin autorización es un crimen por el que se puede ser juzgado y encarcelado.

3.8.2. ¿Qué se necesita para ser un buen Ethical Hacker?

De acuerdo con Rodríguez, J. (2011), hay conocimientos que se deben tener para un buen ethical hacker, como son:

- Redes: Wifi y cableadas.
- Sistemas Operativos: Windows 8, Windows 8.1, OS/400, GNU/Linux (Kali Linux, Parrot, Ubuntu, etc.), Mac OS, Unix, etc.
- Programación: JavaScript, Java, C#, Python, Kotlin, etc.
- Pentesting.
- Bases de Datos: Oracle, SQL Server, MySQL, PostgreSQL.

Además de habilidades mencionadas arriba, es ideal contar con los siguientes puntos:

- Paciencia y leer mucho.

- Computadora con acceso a internet.
- Contar con un grupo de máquinas virtuales con Windows y Linux.
- Programas para reconocimiento.
- Programas para escaneo.
- Programas para acceder a sistemas.
- Programas para mantener el acceso a sistemas.
- Programas para borrar huellas del sistema.
- Microsoft Word, y Microsoft Visio.

3.8.3 Ciclo del Hacking

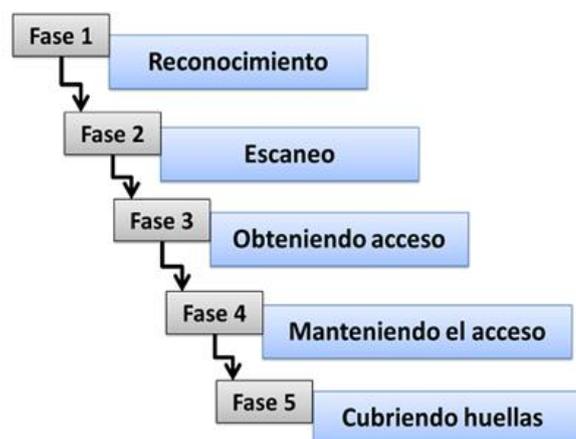
A la hora de hackear se debe estar claro de lo que se va a hacer, se debe tener una seguridad del 100% y un objetivo claro. Todo esto porque como hacker hay que entender que, si entramos a un sistema de alta seguridad sin un plan o sin conocer el terreno, el hacker podría ser descubierto y encarcelado. Por lo que los hackers generalmente siguen el siguiente ciclo a la hora de hackear un sistema. (Rodríguez, J., 2011, p.30).

Según Rodríguez, J. (2011), el hacking tiene diferentes etapas que se dividen en las siguientes:

1. **Reconocimiento:** esta fase consiste en reunir información sobre el objetivo, se hace reconocimiento de 2 formas:
 - **Activo:** el reconocimiento activo consiste en obtener información de forma física.
 - **Pasivo:** este consiste en buscar información de forma ligera, como buscar en internet.

En esta etapa el hacker debe ser capaz de conseguir la siguiente información: nombre, dirección física, números de teléfono, puesto en la empresa, correos electrónicos, IP de la empresa, cuántos empleados y que sistemas de seguridad utilizan.

2. **Escaneo:** en esta fase el hacker puede escanear y saber por dónde se puede entrar al sistema y por donde no se debe. Para hacer esta fase con éxito el hacker debe saber que quiere hacer, para saber por qué puerto puede hacerlo, o más bien que sería lo más conveniente según la necesidad.
3. **Ganar Acceso:** esta es la parte más excitante para un hacker el poder abrir el sistema y entrar es sin duda una de las razones por la que los hackers hacen lo que hacen.
4. **Mantener acceso:** Mantener el acceso no siempre es lo que buscan los hackers. Una técnica efectiva debería ser completar todo el proceso del ataque de una sola vez, entrar varias veces a un sistema aumenta la probabilidad de que detecten al hacker. Si se toma esto como secundario, la próxima parte sería mantener el acceso al sistema para entrar en el futuro.
5. **Borrar huellas:** definitivamente una etapa muy importante, aunque en si un hacker no debe de borrar huellas, lo que debe hacer es nunca dejar huellas. Si un hacker borra un archivo para no dejar rastros, pues ya lo está dejando, al igual que si editara los logs files porque se está editando un archivo.



González, R. (2020). *Etapas del hacking*. [Figura 3.9]. Recuperado de <https://ehack.info/las-fases-del-hacking-etico/>

Según Rodríguez, J. (2011), sí un hacker borra algún archivo para borrar sus huellas, ya está dejando una huella, si edita los logs, está dejando huellas, porque modificó un archivo. Ya muchos hackers saben esto, y lo que hacen es que se conectan a sitios públicos, por si son rastreados, lleguen a un restaurante o a un cibercafé. Los hackers saben que no pueden hacer ningún ataque desde su hogar, porque podrían cogerlo.

3.8.4 Tipos de hackers

Según Rodríguez, J. (2011), los hackers se clasifican según el objetivo y la forma en la que hacen las cosas (hacking):

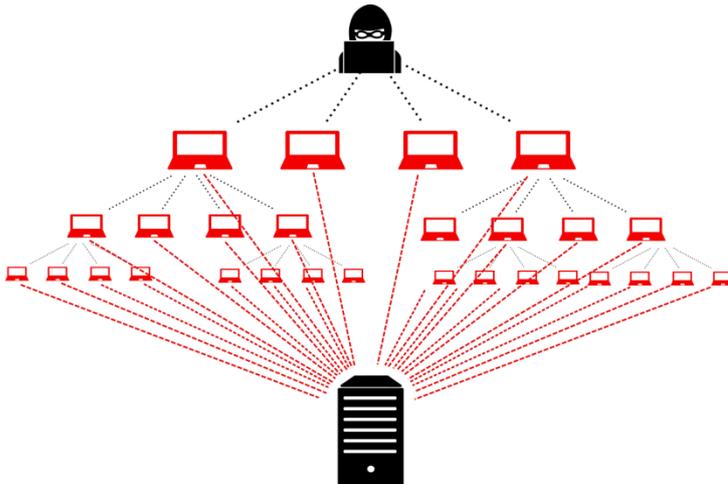
- **White Hat Hacker:** estos son los buenos, trabajan para mejorar la seguridad de una compañía con técnicas de hacking como pentesting, pero estos no lastiman a la compañía.
- **Black Hat Hacker:** este es el hacker con malas intenciones, también denominado cracker. Se enfoca en dañar sistemas, robo de información, alteración de información, etc.
- **Grey Hat Hacker:** Este es un punto medio entre White y Black Hat hacker.

3.9 Ciberataques

Los ciberataques son algo que han existido por un buen tiempo ya, son algo muy común en las empresas. Podemos definir un ciberataque como acción ofensiva hacia los sistemas informáticos de personas o empresas, y estos pueden ser ocasionados por diversas razones, desde lo personal hasta lo económico. Dependiendo del objetivo del ciberataque se puede usar uno u otro método de ciberataque.

3.9.1 Tipos de Ciberataques

- **Ataque de Denegación de Servicios Distribuidos (DDoS):** Este es un de los ataques más comunes, pero a su vez de los más efectivos, básicamente consiste en generar múltiples peticiones a un servidor (el objetivo) hasta que este sea incapaz de atender estas peticiones, lo que provoca un colapso del servidor.



RuggedTooling (2018). Ciberataques. [Figura 3.10]. Recuperado de <https://ruggedtooling.com/what-are-ddos-attacks/>

- **Ataque de SQL injection:** Este tipo de ataque utiliza código SQL, que es el lenguaje que se usa para extraer y manipular información de una base de datos. El ataque de SQL injection consiste en usar código malicioso para extraer información de una base de datos.
- **Phishing:** Este es un tipo de ataque realmente común y fácil de llevar a cabo, y que solo consiste en enviar un correo a nombre de alguna entidad de forma que el receptor del correo cree que es un correo legítimo, pero en realidad el objetivo del correo es obtener información de acceso de las víctimas, como contraseñas, números de cuenta, usuarios de redes sociales, etc.
- **Ataques con gusanos o worms:** este programa maligno es usado muy comúnmente para llevar a cabo ciberataques, consisten en replicarse a sí mismo en el equipo residente y a otros equipos.
- **Cross Site Scripting:** este tipo de ataque se centra en el usuario y no en el servidor como

en el SQL injection, consiste en inyección de código malicioso en un sitio web que es ejecutado en el navegador del usuario.

3.10 Malwares

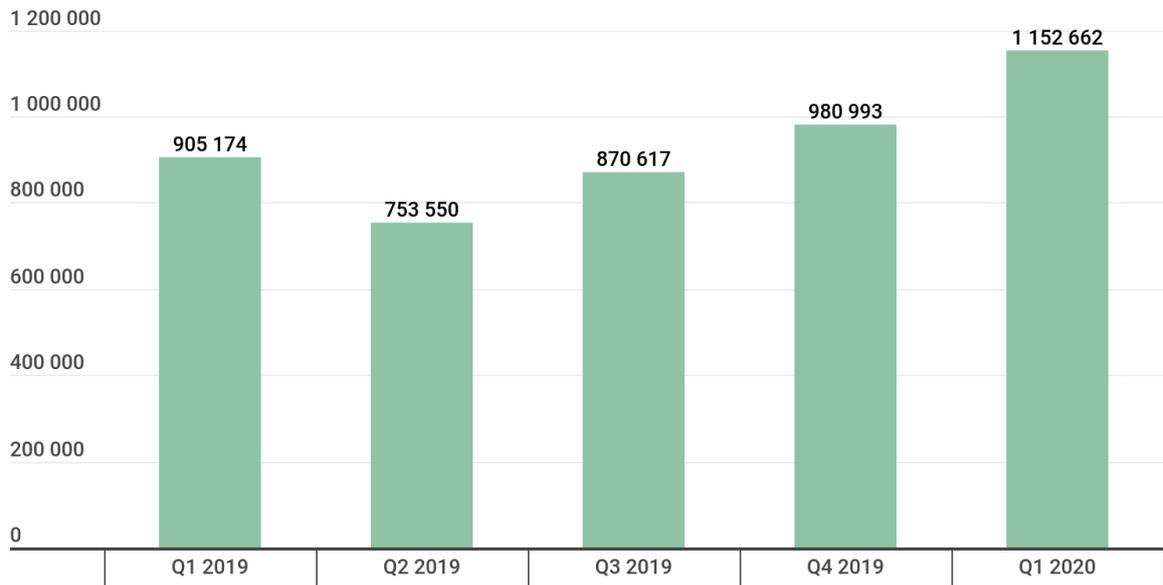
Según Rivero, S. un malware es un “Término que engloba a todo tipo de programa o código informático malicioso cuya función es dañar un sistema o causar un mal funcionamiento.”

Es importante comprender este tipo de amenazas que son usadas en contra de los sistemas de información, los malwares se pueden usar de diferentes maneras dependiendo del objetivo.

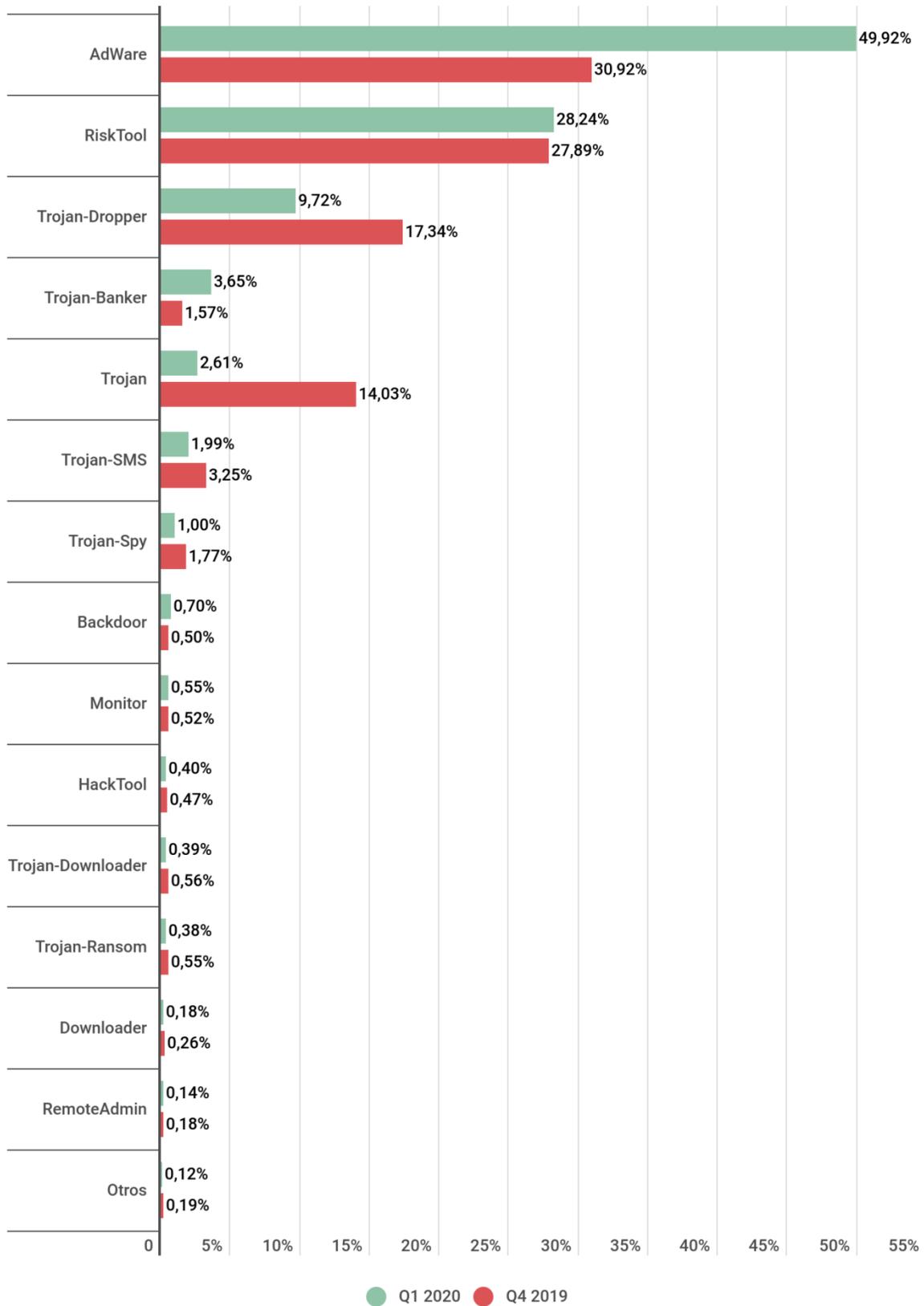
3.10.1 Tipos de Malwares

- **Ransomware:** este es un tipo de programa maligno peligroso porque lo que hace es tomar posesión de un sistema informático mientras se exige algo a cambio del sistema de información. Lo que hace el ransomware es quitarle el acceso del usuario al equipo, usualmente a través de un cifrado de las unidades de almacenamiento.
- **Adware:** es un programa maligno invasivo y molesto que entrega y carga anuncios en el pc del usuario.
- **Troyano:** este programa maligno se camufla y se ve como un archivo o programa normal con el fin de engañar a los usuarios para que descarguen el malware.
- **Spyware:** es de los más usado y lo que haces es espiar las actividades del usuario sin el conocimiento de este.

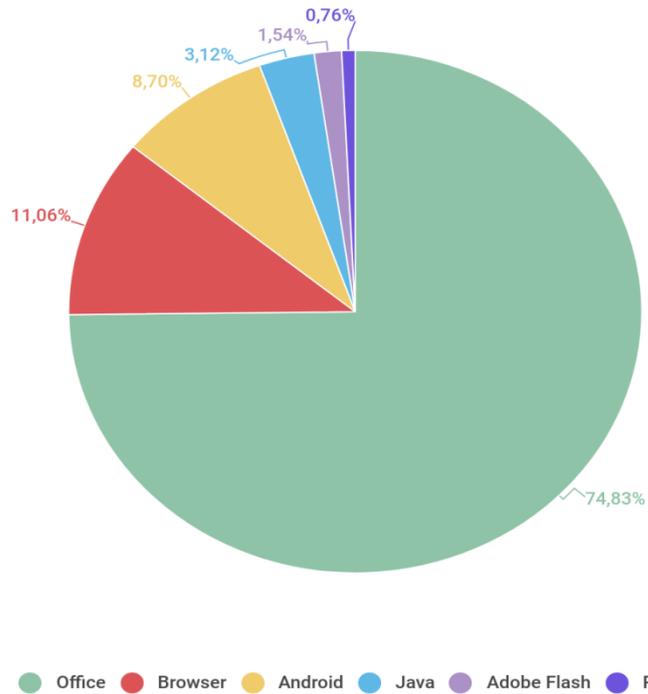
A continuación, se muestran algunos datos estadísticos obtenidos de la compañía de seguridad contra malwares conocida como Kaspersky. Estos gráficos fueron obtenidos a través de las diferentes herramientas de análisis y monitoreo que tiene Kaspersky para protección de malwares:



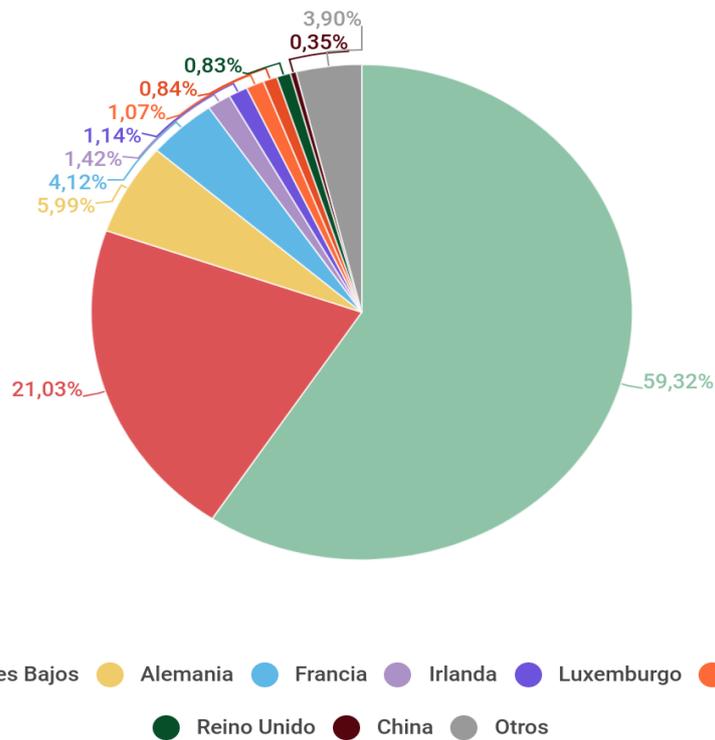
Kaspersky. (2020) Número de paquetes de instalación maliciosos detectados, primer trimestre de 2019 – primer trimestre de 2020. [Figura 3.11]. Recuperado de <https://securelist.lat/it-threat-evolution-q1-2020-statistics/90344/>



Kaspersky. (2020) Distribución por tipo de los nuevos programas móviles detectados, cuarto trimestre 2019 y primer trimestre de 2020. [Figura 3.12]. Recuperado de <https://securelist.lat/it-threat-evolution-q1-2020-statistics/90344/>



Kaspersky. (2020) Distribución de exploits utilizados por los ciberdelincuentes por tipo de aplicaciones atacada, tercer trimestre de 2019. [Figura 3.13]. Recuperado de <https://securelist.lat/it-threat-evolution-q1-2020-statistics/90344/>



Kaspersky. (2020) Distribución por país de las fuentes de ataques web, primer trimestre de 2020. [Figura 3.14]. Recuperado de <https://securelist.lat/it-threat-evolution-q1-2020-statistics/90344/>

3.10.2 Protección contra Malwares

Según la compañía Cisco (2014), hay 5 pasos muy efectivos que pueden servir para evitar o reducir los daños que puedan ocasionar los malwares, estos pasos son:

- 1. Cómo detectarlo y bloquearlo en el perímetro:** Administrar las amenazas tan cerca del perímetro como sea posible para evitar que el programa maligno ingrese a la red y potencialmente infecte los dispositivos terminales. Un dispositivo de detección de programa maligno es una estrategia importante de seguridad perimetral. Estos dispositivos aprovechan los servicios en la nube o utilizan sandboxes incorporados para evaluar no solo a qué se parece el programa maligno, sino que hace. Con la búsqueda de programa maligno conocido, el dispositivo basado en red para la detección de malware debería funcionar en línea para bloquear el ataque sin influir en el rendimiento de la red ni agregar latencia.
- 2. Cómo proteger el interior de la red:** Busque el programa maligno (y otros ataques) en los segmentos protegidos de la red que contienen activos de tecnología sensibles, si se supone que ningún ataque se origina desde un elemento interno de confianza. Los atacantes avanzados adoptan un enfoque sistemático para obtener acceso a las redes de las organizaciones. Comienzan comprometiendo cualquier dispositivo para obtener un punto de apoyo dentro de la organización. Después de integrarse en la red, los atacantes penetran cada vez más profundo y comprometen dispositivos adicionales hasta alcanzar su objetivo.
- 3. Cómo evaluar y proteger los terminales:** Es poco aconsejable suponer que cualquier dispositivo de seguridad perimetral será un 100% eficaz para bloquear todas las amenazas. Las defensas sofisticadas contra el programa maligno avanzado en cada terminal son otra

capa fundamental de la defensa contra ataques. Debido a que muchos dispositivos no siempre están conectados a una red corporativa y no siempre tienen las protecciones de programa maligno basado en red, esos terminales necesitan protección en sí mismos. Dado que a los empleados no les agrada ver comprometida su experiencia como usuarios, cualquier agente de terminal que se instale debe ser ligero y no dificultar el rendimiento.

4. Cómo analizar las amenazas: Mantenga la visibilidad de toda la actividad de archivos y realice el seguimiento del tráfico de salida para detectar la potencial exportación no autorizada de datos críticos e identificar el tráfico de comando y control para indicaciones de dispositivos comprometidos. A pesar de los grandes esfuerzos realizados a través de los controles de protección en las redes y en los terminales, todavía es necesario analizar y evaluar los ataques de programa maligno a fin de determinar su grado de amenaza para las organizaciones.

5. Cómo eliminar el malware e impedir la reinfección: Ponga en cuarentena y limpie el dispositivo infectado a fin de minimizar el riesgo para otros dispositivos de la red. Sin embargo, eso no es suficiente para realmente eliminar el programa maligno o evitar la reinfección. El análisis de datos masivos para realizar un seguimiento de cada archivo de cada dispositivo (de dónde provino y qué ocasionó su ejecución) ofrece visibilidad en toda la empresa e identifica todas las instancias del archivo de malware. Esto garantiza que se encuentre al paciente cero (el primer dispositivo víctima de programa maligno) y que se controle cualquier infección subsiguiente. La protección eficaz contra el malware avanzado también requiere la capacidad de etiquetar los archivos que más adelante se identifiquen como malware. Esto permite identificar los objetivos de ataques para la corrección específica. Después de la corrección, las organizaciones deben implementar

normas integradas en el gateway del perímetro de seguridad, dentro de los dispositivos que protegen redes internas y en los terminales para eliminar el riesgo de reinfección

3.11 Seguridad Biométrica

Según el Instituto Nacional de Ciberseguridad (2016) de España, la biometría es un método de reconocimiento de personas basado en sus características fisiológicas o de comportamiento. Se trata de un proceso similar al que habitualmente realiza el ser humano reconociendo e identificando a sus congéneres por su aspecto físico, su voz, su forma de andar, etc.

Las tecnologías biométricas se definen como métodos automáticos utilizados para reconocer a personas sobre la base del análisis de sus características físicas o de comportamiento. Dependiendo de la técnica biométrica empleada, los parámetros considerados son diferentes: los surcos de la huella dactilar, la geometría de la mano, la voz, la imagen facial, etc. De estos parámetros se extrae un patrón único para cada persona, que será el que se utilice para posteriores comparaciones. Las tecnologías biométricas se aplican en dos fases: registro y autenticación. (Instituto Nacional de Ciberseguridad de España, 2016, p. 4).

3.11.1 Tipos de Seguridad Biométrica

Según el Instituto Nacional de Ciberseguridad (2016) de España, los tipos de tecnología biométricas son:

- **Tecnologías biométricas fisiológicas:** Las tecnologías biométricas fisiológicas se caracterizan por considerar parámetros derivados de la medición directa de algún rasgo estrictamente físico del cuerpo humano a la hora de identificar personas.

- **Reconocimiento facial:** El reconocimiento facial es una técnica mediante la cual se reconoce a una persona a partir de una imagen o fotografía. Para ello, se utilizan programas de cálculo que analizan imágenes de rostros humanos.
- **Reconocimiento de la geometría de la mano:** Esta tecnología utiliza la forma de la mano para confirmar la identidad del individuo. Para la captura de la muestra se emplean una serie de cámaras que toman imágenes en 3-D de la mano desde diferentes ángulos.
- **Reconocimiento de retina:** El escáner biométrico de la retina se basa en la utilización del patrón de los vasos sanguíneos contenidos en la misma. El hecho de que cada patrón sea único (incluso en gemelos idénticos al ser independiente de factores genéticos) y que se mantenga invariable a lo largo del tiempo, la convierten en una técnica idónea para entornos de alta seguridad.
- **Reconocimiento vascular:** En la biometría vascular se extrae el patrón biométrico a partir de la geometría del árbol de venas del dedo (o de las muñecas). A diferencia de la huella dactilar el patrón biométrico es interno, por esta razón no deja rastro y sólo se puede conseguir en presencia de la persona. Es por tanto muy difícil el robo de identidad.

3.11.1.1 Seguridad de lector de huellas

Según el Instituto Nacional de Ciberseguridad (2016) de España la identificación basada en huella dactilar es la más antigua de las técnicas biométricas y ha sido utilizada en un gran número de aplicaciones debido a que se considera que las huellas dactilares son únicas e inalterables. Existen dos tipos de técnicas de búsqueda de coincidencias entre muestras de huella dactilar:

- **Basadas en minucias:** Esta técnica se basa en la autenticación de minucias en otras palabras se basa en determinadas formas fácilmente identificables existentes en la huella dactilar. (Instituto Nacional de Ciberseguridad de España, 2016, p. 7).



Instituto Nacional de Ciberseguridad de España (2016). *Minucias de Huellas Dactilares*. [Figura 3.15]

- Basadas en correlación:** Mediante la utilización de esta técnica se analiza el patrón global seguido por la huella dactilar, es decir, el esquema general del conjunto de la huella en lugar de las minucias. Esta técnica requiere un registro preciso, pero su principal inconveniente es que se ve afectada por la traslación y la rotación de la imagen. (Instituto Nacional de Ciberseguridad de España, 2016, p. 7).



Instituto Nacional de Ciberseguridad de España (2016). *Patrones Huellas Dactilares*. [Figura 3.16]

3.11.1.2 Verificación por voz

De acuerdo con el Instituto Nacional de Ciberseguridad (2016) de España las aplicaciones de reconocimiento de voz usan sistemas de inteligencia artificial (en concreto redes neuronales) para aprender a identificar voces. Los algoritmos deben medir y estimar la similitud entre las muestras

para devolver un resultado o una lista de posibles candidatos. La identificación se complica debido a factores como el ruido de fondo, por lo que siempre es necesario considerar un margen de error. A pesar de que siguen existiendo dificultades para reconocer la forma natural de hablar de ciertos individuos, esta tecnología cuenta con la ventaja de que el dispositivo de adquisición es simplemente un micrófono por lo que no requiere de inversiones adicionales.

3.11.1.3 Patrones Oculares

El escáner biométrico de la retina se basa en la utilización del patrón de los vasos sanguíneos contenidos en la misma. El hecho de que cada patrón sea único (incluso en gemelos al ser independiente de factores genéticos) y que se mantenga invariable a lo largo del tiempo, la convierten en una técnica idónea para entornos de alta seguridad. Pese a que su tasa de falsos positivos sea prácticamente nula, esta tecnología tiene un inconveniente considerable ya que es necesaria la total colaboración por parte del usuario al tratarse de un proceso que puede resultar incómodo. La toma de la muestra se realiza a partir de la pupila, lo que requiere que el usuario permanezca inmóvil y muy cerca del sensor durante la captura de la imagen. No obstante, el uso de una cámara de infrarrojos para la captura evita el riesgo de que el ojo pueda resultar dañado en el proceso. (Instituto Nacional de Ciberseguridad de España, 2016, p. 9).

3.12 Conclusión Capítulo III

En este capítulo hemos podido visualizar lo importante que es la seguridad de la información, debido a que la misma es uno de los activos más valiosos que puede tener una empresa, por el alto valor que esta tiene para los usuarios, clientes, administradores, etc. La información se compone de todos los datos recopilados del entorno de la empresa y también de los procesos que se llevan

a cabo en esta, para así proveer facilidad de uso de sistemas, informes, reportes, etc.

Por lo anteriormente mencionado es que se debe de salvaguardar la información a través de todos los medios posibles. Ya que los datos de una empresa tienen información sensible en diferentes ámbitos.

Para proteger la información existen diferentes métodos y mecanismos como vimos en este capítulo, tenemos los firewalls, antivirus, herramientas de monitoreo, gestión de usuarios y contraseñas, etc. Para evitar cualquier tipo de problema que involucre o que ponga en riesgo la información, lo ideal es implementar robustas políticas de seguridad de manera que sea posible tener un control adecuado de los datos que son manipulados en la empresa u organización. Siempre que se mantenga la integridad, seguridad y disponibilidad de la información, las empresas pueden contar con esta en el momento que sea necesaria.

**CAPÍTULO IV: PROPUESTA DE
IMPLEMENTACIÓN DE LA ENSEÑANZA DE
PROGRAMACIÓN DE SOFTWARE COMO
MATERIA BASE A NIVEL MEDIO**

4.1 Introducción

Con el objetivo de poder definir un alcance de la implementación de programación de software en escuelas de República Dominicana, posterior al análisis y a la documentación que se ha llevado a cabo hasta ahora en esta investigación.

Este capítulo consiste en presentar el sistema completo de la problemática o la situación actual del Instituto Fabio Amable Mota, que en parte dio origen al mismo, también se mostrará el sistema que se requiere para la implementación de este proyecto y los demás elementos que son muy necesarios para el desarrollo de esta propuesta.

En este apartado veremos diagramas de diseños de algunos aspectos de la propuesta, se verán ejemplos de países que implementaron la programación de software como parte de su programa educativo, además de tablas y gráficos que demostraran la posible mejoría de la calidad del estudiantado. Además de que también se mostrará el programa de estudio para los estudiantes y algunas guías de mejoras para los profesores.

4.2 Descripción de la Situación Actual del INSTITUTO FABIO AMABLE MOTA

En cuanto a nivel educacional no se puede decir que el INSTITUTO FABIO AMABLE MOTA este y bajo nivel, ya que actualmente enseñan lo necesario para un bachiller según el pensum escolar normal exigido por el Ministerio de Educación de República Dominicana (MINERD) y además también preparan a los jóvenes en diferentes áreas técnicas como son Informática, equipos electrónicos, mecatrónica, etc.

En cuanto a la programación de software, esta solo es enseñada a los estudiantes de informática, aunque esta materia puede ser enseñada en todas las áreas y todos los niveles de la escuela media, que es en sí lo que proponemos. Además cuentan con diferentes laboratorios asignados para cada área que hacen del aprendizaje una experiencia más práctica. Como el instituto tiene el plan de educación genérico proporcionado por el MINERD, tiene muchas oportunidades de mejora para hacer que los estudiantes se desarrollen y tengan mejores habilidades para cuando les toque salir de la escuela.

La situación actual a nivel educativo en la República Dominicana no es la mejor de todas, según un estudio realizado en el 2019 nuestro país ocupa el lugar número 90 de 149 países en cuanto a educación se refiere. Por este motivo es importante y necesario implementar nuevos métodos y nuevos temas para impulsar el desarrollo de la educación como tal, aumentando así la calidad de los estudiantes.

A nivel de infraestructura el instituto ya cuenta con laboratorios equipados para informática, en donde se tienen computadoras Dell Optiplex 9020, con 4GB de RAM y un CPU i3 de 2.7 GHz. Además de que debido al proyecto República Digital, cada estudiante tiene su laptop personal, no

de muchos recursos, pero sí los suficientes para que se puedan usar para este proyecto, que es implementar la programación como materia básica en las escuelas.

4.3 Diseño de la red actual del INSTITUTO FABIO AMABLEMOTA

El área de informática del Instituto Tecnológico Fabio Amable Mota consta de 2 laboratorios. Primero veremos un gráfico que describe cómo se distribuyen los pc, y la conexión que tienen los laboratorios.

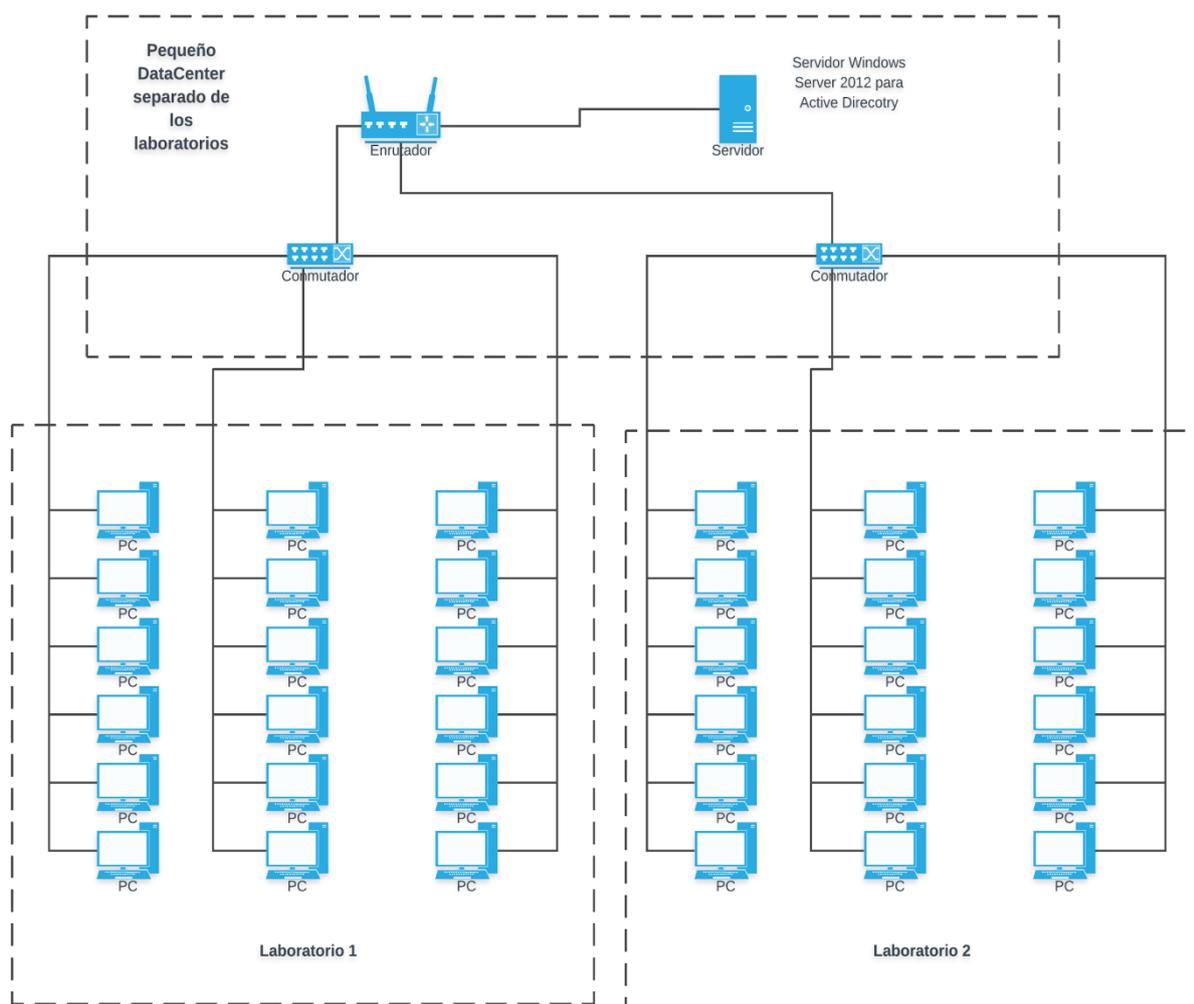
En el laboratorio 1 la red está diseñada de la siguiente manera:

- Hay un total de 16 computadoras en el laboratorio.
- Se utiliza un Switch como dispositivo de red para conectar los equipos en el laboratorio, creando una Red LAN de tipología estrella. Cada computadora de este laboratorio está conectada al switch mediante un cable directo.
- En el Instituto hay un espacio llamado Data Center en donde se encuentra el router que utiliza el switch para conectarse a internet. Mediante un cable directo el switch de este laboratorio se conecta al router, proporcionando de esta manera internet a los equipos conectados al switch. El cable que se utiliza para conectar el switch con el router atraviesa el techo del Instituto, debido a que el data center es un espacio separado del laboratorio 1.

En el laboratorio 2 el diseño de red funciona de la siguiente manera:

- El laboratorio 2 consta de 18 computadoras.
- En este laboratorio se utiliza una red basada en servidores. Hay un total de 4 equipos que funcionan como servidor, en el cual cada servidor tiene a su disposición otros 4 equipos que funcionan como clientes de ese servidor.

- Al igual que en el laboratorio 1 los equipos están conectados a un switch mediante cable directo y de igual manera el switch se conecta al router que está en el Data Center para proporcionar internet a los equipos de este laboratorio.



Los Autores: Rodríguez, A. y Florentino, L. (2020). Diagrama de laboratorios de informática del Instituto Fabio Mota. [Figura 4.1] República Dominicana

Área de administración

- En el área de administración, los equipos que se utilizan en los distintos departamentos se conectan al router que está en el Data Center. Algunos tienen una conexión mediante cable directo, pero hay otros equipos como laptops y celulares que se conectan a internet

- mediante conexión inalámbrica.
- Se tiene un servidor Windows Server 2012, para manejo de servicio, principalmente active directory para manejo de usuarios y manejo de políticas.

4.4 Descripción General de la Propuesta

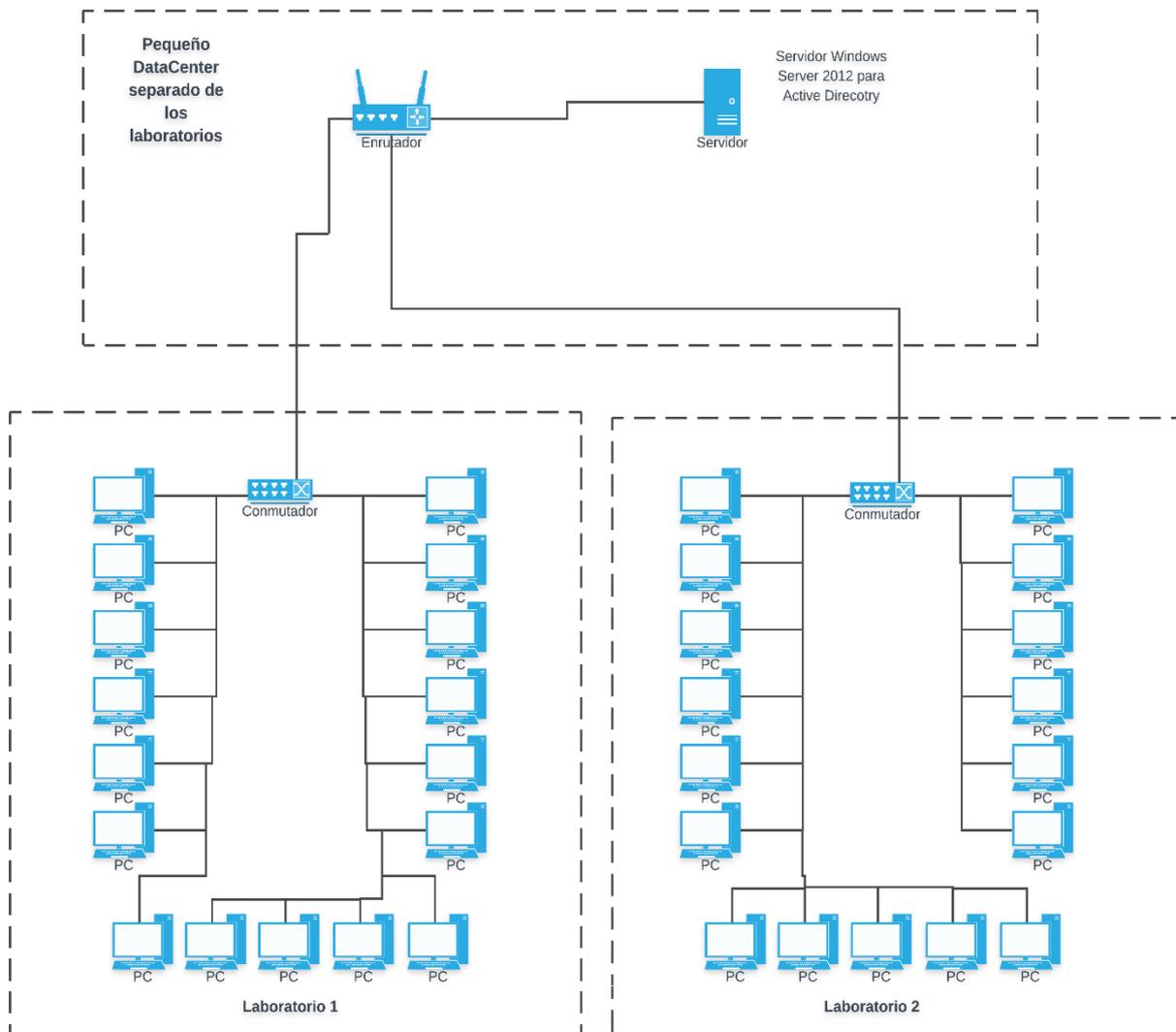
La implementación de la programación de software en escuelas a nivel medio en la República Dominicana, en este caso el Instituto Fabio Mota, permitirá a los estudiantes el aprendizaje de una de las habilidades más necesarias en este siglo debido a que todo está orientado a la tecnología, por ende el saber programar hará que los estudiantes lleven una tremenda ventaja durante sus estudios y también después de culminar la escuela, esto es debido a las capacidades que adquirirán, entre estas podemos nombrar:

- La capacidad de resolución de problemas.
- Aumento de la capacidad de abstracción a través de los análisis realizados a la hora de resolver un problema.
- Mejora en el trabajo en equipo.
- Mejora en el pensamiento cognitivo.
- Pensamiento crítico y analítico.

Las anteriores son solo algunas de las capacidades y ventajas que adquirirán los estudiantes al implementar nuestra propuesta. Mediante nuestra propuesta no solo se consigue mejorar las aptitudes y habilidades de los estudiantes, sino que también se les capacita para poder tener una mejor y más amplia respuesta ante el mundo real en el momento que les toque insertarse en el mundo laboral.

Para el desarrollo de la propuesta contamos con un plan de estudios diseñado y pensado para cada nivel de escuela media, además de enseñar los temas relevantes para aprender a programar también se empleará el uso de algunas herramientas (software) de fácil acceso y con una excelente documentación para que los estudiantes sepan que tienen un buen soporte si en un momento no cuentan con algún maestro que les aclare alguna duda. Nuestro plan de estudio consta de implementar la frase de “Aprender haciendo”, básicamente además de enseñar los conocimientos teóricos básicos a los estudiantes, el programa contempla la asignación de proyectos interactivos y divertidos que realmente llamen la atención de los jóvenes para que realmente quieran aprender a programar. Se puede decir que se utilizara un método de aprendizaje constructivista (el estudiante construye su conocimiento en base a sus propias pruebas y experiencias).

Otro de los puntos de la propuesta es el de remodelar los laboratorios y comprar equipo nuevo de más capacidad y más tecnología, además de un orden diferente en la posición de los equipos, la única parte que no cambiaría sería el pequeño datacenter que utilizan para la administración de los equipos del laboratorio y la escuela en general



Los Autores: Rodríguez, A. y Florentino, L. (2020). Nuevo Diagrama de laboratorios de informática del Instituto Fabio Mota. [Figura 4.2] República Dominicana

4.4.1 Análisis FODA



Los autores: Rodríguez, A. (2020) Matriz de análisis FODA Programación en escuelas. [Figura 4.3]

Fortalezas:

- Mejora de la calidad de los estudiantes.
- Facilidad de aprender remoto.
- Foros y documentación disponible para aprender como autodidacta.
- Mejora de las competencias y habilidades de los estudiantes.

Debilidades:

- Necesidad de una computadora o algún dispositivo inteligente.
- Necesidad de internet.

Oportunidades:

- Se puede implementar en todas las escuelas y colegios del país.

- También se puede enseñar a niños.

Amenazas:

- Costos de mantenimiento y actualización de equipos.
- Posible ciberataque.

4.4.2 Estudio de Factibilidad y Económica

A continuación, veremos un estimado de lo que puede costar tener un laboratorio funcional para que los jóvenes estudiantes del Instituto Fabio Mota puedan aprender a programar, estudiantes de todos los niveles y áreas de nivel medio en este instituto. Se propone montar un laboratorio que tenga un total de 30 equipos de capacidad promedio para trabajar y aprender programación en la escuela, esto se combinara con el uso de las laptops dadas por el proyecto República Digital para que los estudiantes puedan llevar a cabo sus asignaciones cuando estén en sus casas, además de que en el plan de estudio se utilizará una herramienta llamada Git y GitHub que permitirá a los estudiantes tener el código disponible en cualquier computadora porque almacenaran los proyectos creados en la nube.

Recursos Humanos			
Cantidad	Cargo	Costo Individual	Costo Total
2	Maestro	\$ 40,000.00	\$ 80,000.00
1	Instalador de equipos	\$ 6,000.00	\$ 6,000.00
			\$ 86,000.00

Los autores: Rodríguez, A. (2020). Costos de los recursos humanos. [Tabla 4.1]. República Dominicana

Recursos Tecnológicos			
Hardware			
Cantidad	Descripción	Costo Individual	Total
30	Computadoras Dell Optiplex 780	\$ 4,000.00	\$ 120,000.00
30	Monitores	\$ 1,600.00	\$ 48,000.00
30	Mouse	\$ 500.00	\$ 15,000.00
30	Teclado	\$ 1,000.00	\$ 30,000.00
1	TP-Link Switch Red RJ45 48 Ports Rackable 100 Mbps	\$ 16,375.00	\$ 16,375.00
1	Computadora Optiplex 9020 i7	\$ 15,000.00	\$ 15,000.00
Software			
Descripción	Costo		
Pseint	Gratis		
VS Code	Gratis		
Python 3	Gratis Licencia GPL		
React JS	Gratis		
Google Chrome	Gratis		
Git	Gratis Licencia GPL		
Linux (Ubuntu)	Gratis Licencia GPL		
Windows 10	Gratis Licencia Educacional		

Los autores: Rodríguez, A. (2020). *Costos de los recursos humanos*. [Tabla 4.2]. República Dominicana

4.4.3 Objetivos de la Propuesta

El objetivo principal de nuestra propuesta es mejorar la calidad de los estudiantes que se preparan en las escuelas de nuestro país tomando como caso de estudio el Instituto Fabio Amable Mota. Se quiere mostrar que el implementar esta materia en las escuelas no representa un gran gasto, más sin embargo nos retorna grandes beneficios en cuanto a calidad de educación se refiere y también aumenta la posibilidad de obtener un buen empleo al terminar la escuela. Buscamos seguirle los pasos a países como España que ya implementan esta materia en las

escuelas, para así en unos cuantos años tener un grupo de estudiantes altamente capacitados, no solo a nivel técnico, sino que también a nivel de razonamiento y pensamiento computacional.

4.5 Desarrollo de la Propuesta

4.5.1 Plan de Estudios para aprender a programar a nivel medio de escuela

La esencia principal de nuestra propuesta no se basa en lo físico, es decir, no se basa en el cambio de un laboratorio ni equipar a los estudiantes, aunque es parte fundamental del proyecto, pero más bien, se tiene como esencia u objetivo la implementación de la programación de software como una materia básica en las escuelas o colegios, por lo que para esto se ha desarrollado un plan de estudios pensado para asegurar el máximo aprendizaje. El plan de estudio para esto se parecerá a un pensum de universidad:

1. Primer nivel, en este nivel se verá material introductorio a los diferentes conceptos que puede involucrar la programación, para cuando los jóvenes empiezan a escribir código tengan un entendimiento general de que es lo que están haciendo. A continuación, el plan de estudios para este nivel.

Tema 1 - Algoritmos

- Concepto de algoritmo
- Características de los algoritmos
- Cómo representar un algoritmo.
 - Diagrama de flujos.
 - PseudoCódigo

- Práctica de pseudocódigo y diagrama de flujos con PSeint.

Tema 2 - Estrategias y métodos para la solución de problemas

- Entendimiento del problema
- Análisis del problema
- Creación de algoritmo en base al análisis del problema
- Representación del algoritmo a través del diagrama de flujos o pseudocódigo.
- Pruebas del algoritmo
- Crear el código del algoritmo.
- Probar el programa creado a partir del algoritmo.
- Ejecución e implementación del programa.

Tema 3 - Operadores usados en algoritmos y programas.

- Tipos de datos
- Operadores
 - Operadores lógicos
 - Operadores aritméticos
 - Operadores relacionales
- Variables y Constantes
- Prácticas de Operadores en PSeint.

Tema 4 - Lenguajes de programación

- Concepto de lenguajes de programación
- Tipos de lenguaje de programación.
- Estructura de un programa.
- Lenguaje Compilado y lenguaje interpretado.

- Paradigmas de programación.

2. Segundo nivel: En este punto los estudiantes ya tienen una buena base de la programación a modo general, pero ahora se le enseñará un paradigma de programación directamente, que es el de orientado a objetos, ya que esta es la forma de programación más usada a nivel mundial y además de que es de las mejores formas de programar.

Tema 1 - Programación Orientada a Objetos

- Teoría de Objetos
 - Que es un objeto
 - Que es una clase
 - Que es una instancia de clase
- Términos usados en la POO (Programación Orientada a Objetos)
 - Herencia
 - Polimorfismo
 - Abstracción
 - Encapsulamiento
 - Clase
 - Objeto
 - Propiedad
 - Método
 - Operadores
 - Constructores

Tema 2 - Programación con C#

- C#
 - ¿Qué es C#?
 - Características de C#
 - Framework .NET
 - Objetos en C#
 - Garbage Collector
 - Estructura de Programas en C#
 - Prácticas en C#
 - Proyectos en C#.
 - Crear un simulador de cajero automático en C#
 - Crear un sistema de inventario pequeño

Tercer nivel: En este nivel los estudiantes verán el tipo de programación que lidera hoy día, que es la programación web, para esto los estudiantes deben ver lo que es el front end y back end.

Tema 1 - Programación Front End

- HTML5
- CSS3
- JavaScript
- TypeScript
- Git
- Prácticas y Ejercicios

Tema 2 - Programación Back End

- PHP
- Node JS
- React
- Angular
- Python
- Git
- Prácticas y Ejercicios

Tema 3 - Bases de datos

- Introducción a las bases de datos relacionales.
- Modelo Entidad-Relación
- SQL en MariaDB y PostgreSQL
- Prácticas y Ejercicios

Cuarto Nivel: Este será sobre proyecto más complicados. En este nivel el estudiante ya habrá desarrollado ciertas capacidades, a través de la creación de proyectos, y también algunos puntos teóricos. En este punto, las cosas se enfocarán más en utilizar un método libre de construcción de proyectos para que así el estudiante se sienta libre de elegir y programar lo que desee, desde algo tan sencillo como una calculadora, hasta algo un poco más complejo como un videojuego.

Tema 1 - Desarrollo en ambientes de Microsoft

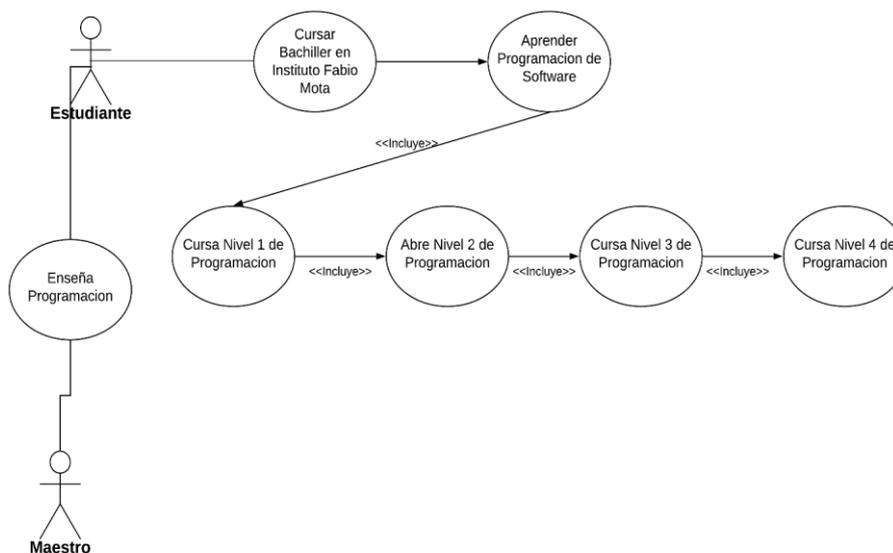
- C# .NET
- ASP.NET
- Unity

- SQL Server
- Azure a nivel básico orientado al desarrollo de software.

Tema 2 - Proyectos: En este tema, los estudiantes solo harán proyectos según el nivel y los conocimientos adquiridos, ya que al hacer proyectos algunos asignados por el profesor y otros de elección libre, los estudiantes desarrollarán mejores capacidades ya que se verán forzados a investigar en indagar.

Lo expuesto en los diferentes niveles demuestra que lo principal en esta propuesta es hacer que los estudiantes tengan que formular sus propias preguntas al momento de programar y que no se aprendan algo de memoria, sino porque tuvieron que investigar y realmente entender lo que iban haciendo.

También podemos ver todo el sistema de niveles para esta propuesta a través de un diagrama de caso de uso, en donde se ve la relación entre el estudiante, maestro y los niveles que el estudiante debe cursar para aprender a programar en el Instituto Fabio Mota.



Rodríguez, A. (2020). Diagrama de casos de uso Programación de Software Instituto Fabio Mota. [Figura 4.4]

4.5.2 Herramientas de software para aprender a programar

Para aprender a programar existen infinidad de herramientas útiles que los alumnos podrán usar, todo dependiendo del nivel del alumno.

Lo primero es que se debe de contar obviamente con una computadora y un sistema operativo, en ambiente de desarrollo lo ideal es que el estudiante utilice varios sistemas operativos, por tal razón además de usar Windows, también usarán varias distros de Linux (CentOS, Fedora, Ubuntu, Debian, etc). Los software a utilizar lo dividimos en categorías según su uso:

1. Sistemas Operativos:

- a. Windows 10 for Education
- b. GNU/Linux (Ubuntu y CentOS)

2. Lenguajes de Programación

- a. Python
- b. JavaScript
- c. C#
- d. TypeScript

3. Bases de datos

- a. MariaDB
- b. PostgreSQL

4. IDE's

- a. Visual Studio
- b. SQL Developer
- c. PyDev
- d. SQL Server Management Studio

5. Editores de texto

- a. Visual Studio Code
- b. PSeint (solo para crear algoritmos)

6. Frameworks

- a. .NET
- b. ASP.NET
- c. React

d. Angular

7. Línea de comandos

a. Bash

b. PowerShell

8. Manejo de repositorios

a. Git

b. GitHub

c. GitHub Desktop

d. Gitlab

Cada una de las herramientas anteriormente indicadas son de las herramientas que más se utilizan en el día de hoy, y además están en constante desarrollo por lo que resulta difícil que se desactualizan. Además de que al entender y aprender a usar todas las herramientas el estudiante además de desarrollar sus habilidades de razonamiento que se desarrollan de por sí con solo programar, también estará muy bien capacitado para enfrentar el mundo real si decide estudiar informática.

4.5.3 Equipos (hardware) necesarios para programar

Requerimientos de Hardware	
Detalle	Detalle
Computadoras Desktop	Modelo: Optiplex 9020 Conexión Estable a Internet de por lo menos 10Mb/s Procesador: i3 2.7GHz Memoria: 4GB Disco Duro: 150GB
Servidor para active directory	Procesador: 2.7GHz Memoria: 12GB Disco Duro: 500GB

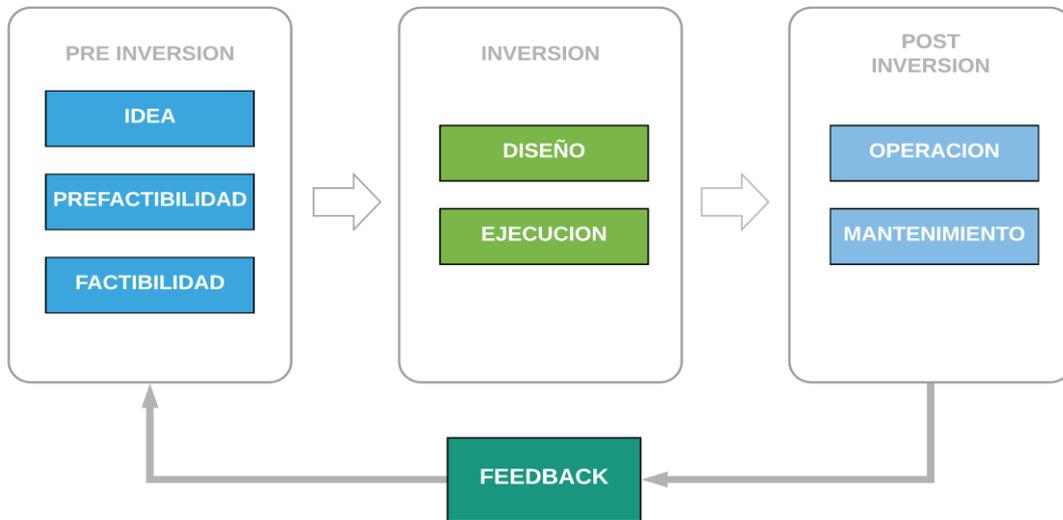
Los autores: Rodríguez, A. (2020). *Requerimientos de Hardware*. [Tabla 4.3]. República Dominicana

Requerimientos de Hardware	
Detalle	Detalle
Sistemas Operativos	Windows 10 for education GNU/Linux (Ubuntu y CentOS)
Web Browser	Google Chrome Mozilla Firefox
Bases de datos	

	MariaDB, PostgreSQL, Oracle y SQL Server
IDEs y Editores de texto	Visual Studio Code, Visual Studio 2019, SQL Developer y SSMS, PSEINT, Pydev
Otros	Git, gitlab extension, github desktop, Python, Unity, Azure

Los autores: Rodríguez, A. (2020). *Requerimientos de Software*. [Tabla 4.4]. República Dominicana

4.5.4 Ciclo de Inversión del Proyecto



Los autores: Rodríguez, A. (2020). *Requerimientos de Software*. [Figura 4.5]. República Dominicana

4.5.5 Beneficios del Proyecto

El proyecto no deja un beneficio económico de ningún tipo, al menos no directamente, debido a que este proyecto se enfoca en agregar una materia más al pensum escolar a nivel medio, con el objetivo mejorar la calidad de los estudiantes como hemos mencionado en este documento.

Beneficios del estudiante:

- Mejora capacidad de trabajar en equipo.
- Mejora capacidad de razonamiento.
- Preparación en una de las áreas mejor pagadas y más demandadas de la época

Beneficios de las escuelas:

- Tienen un mejor currículo para enseñar.
- Generan estudiantes de más calidad.

Beneficios Socioeconómicos

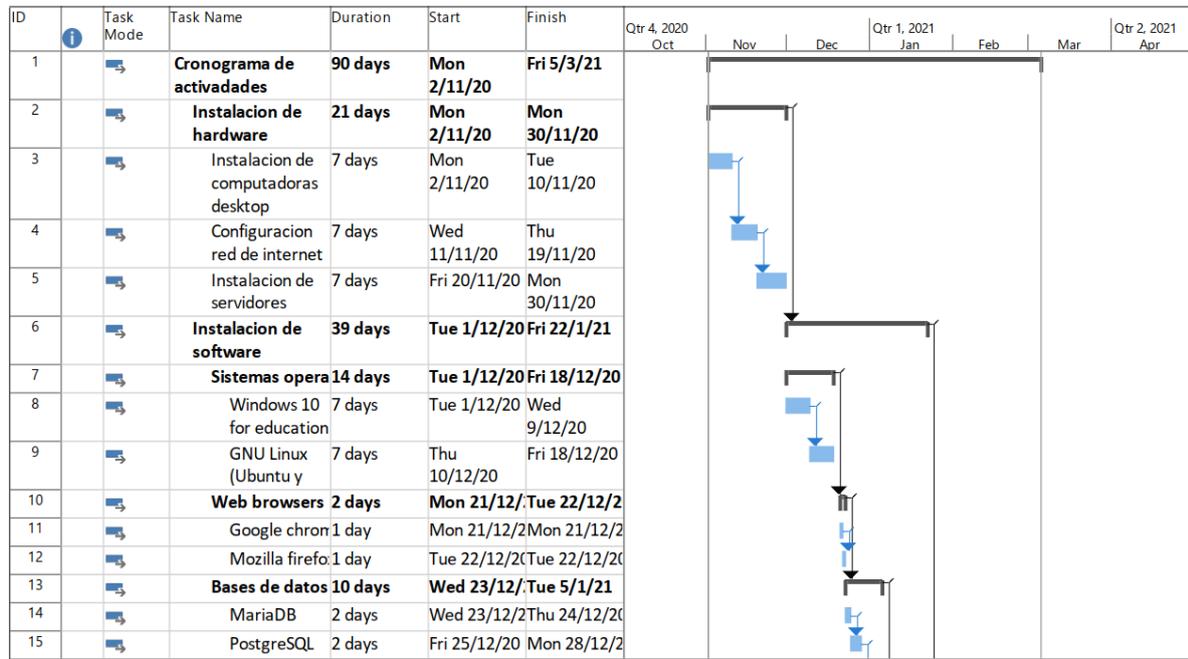
- Los jóvenes estudiantes se enfocarán más en aprender este tipo de habilidades lo que los puede volver más productivos.
- Se generan entes productivos para sociedad.
- Mejora en los niveles de educación del país.

4.6 Retorno de Inversión (ROI)

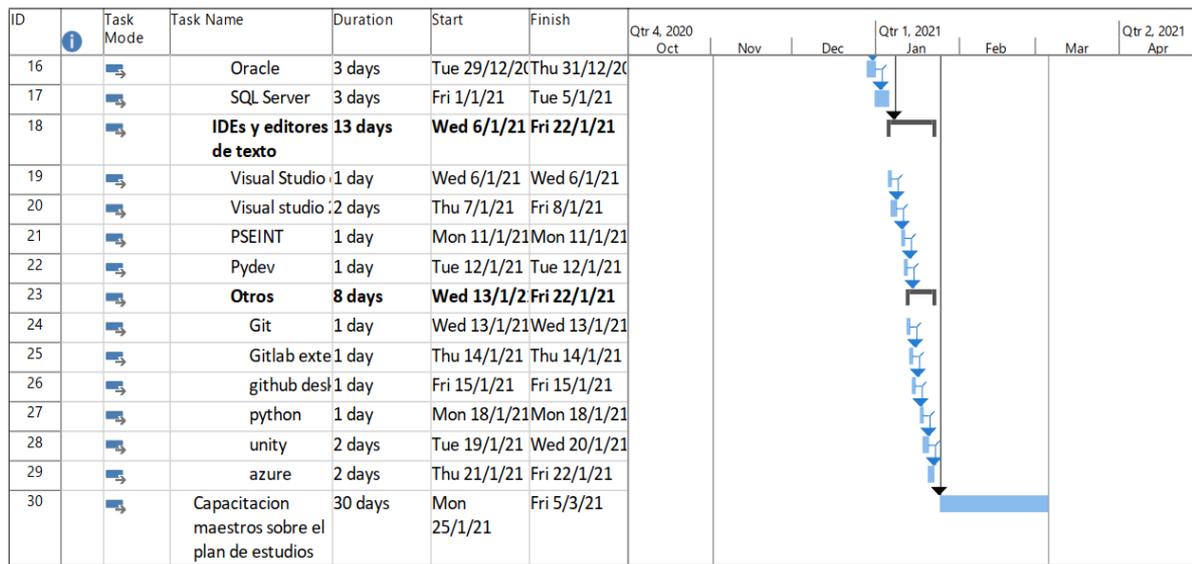
A través de la fórmula: $ROI = ([Beneficios - Inversión] / Inversión) \times 100$ calcularemos el retomo de la inversión del proyecto de implementación de la programación como materia base en escuelas. Antes de calcular el retorno de la inversión, se calculará los gastos y ahorros que se conseguirá al implementar este proyecto. Según el análisis de factibilidad del negocio, la inversión del proyecto no es grande, según nuestros cálculos la inversión total sería RD\$842,423. El tiempo de retorno de inversión de esta propuesta es de 1 año escolar en República Dominicana. Utilizando la fórmula del retorno de la inversión, se tiene que:

Los autores: Rodríguez, A. (2020). Retorno de Inversión (ROI). [Figura 4.6]. República Dominicana

4.7 Cronograma de actividades de implementación del proyecto:



Los autores: Florentino, L. (2020). Cronograma de actividades de implementación del proyecto. [Figura 4. 7]. República Dominicana



Los autores: Florentino, L. (2020). Cronograma de actividades de implementación del proyecto. [Figura 4. 8]. República Dominicana

4.8 Formulario para el análisis de investigación de campo

El objetivo de esta encuesta es medir el nivel de conocimiento de los estudiantes del área de informática dígase los estudiantes que están actualmente cursando en el instituto, recolectar

información sobre sus conocimientos de programación, lógica matemática entre otros. La encuesta está compuesta por 9 preguntas enfocadas a los estudiantes del Instituto Fabio Amable Mota.

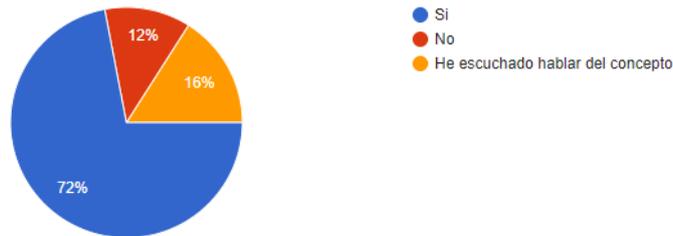
Preguntas	Posibles respuestas			
¿Cuál es su edad?	14-16	17-18		
¿Cuál es su sexo?	Masculino	Femenino		
¿Qué curso estas cursando actualmente?	4to de secundaria	5to de secundaria	6to de secundaria	
¿Sabes que es la programación?	Sí	No	He escuchado hablar del concepto	
¿Aplicas la lógica matemática para la resolución de problemas?	Siempre	Casi Siempre	Nunca	
En una escala del 1-5, ¿Cuál es tu dominio del idioma inglés?	Nulo	Básico	Intermedio	Experto
¿Te gustaría aprender programación como materia base en tu instituto?	Si	No	Definitivamente	Indeciso
¿Qué conocimientos de programación posees?	Básico	Intermedio	Avanzado	Nulo
¿Entiendes que es necesario la creación de nuevas materias en el área de informática?	Si	No		
¿Planeas seguir estudiando algo relativo al desarrollo de software al salir del instituto?	Si	No	Tal vez	

Los autores: Florentino, L. (2020) encuesta ANÁLISIS E IMPLEMENTACIÓN DE LA PROGRAMACIÓN DE SOFTWARE - INSTITUTO TECNOLÓGICO FABIO AMABLE MOTA. [Tabla 4.5]. República Dominicana.

4.9 Análisis y Resultado

¿Sabes que es la programación ?

50 respuestas



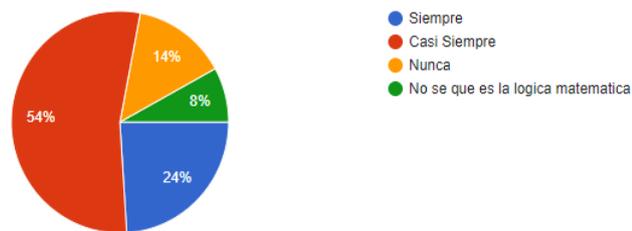
Alternativas	N	%
Si	36	72%
No	6	12%
He escuchado hablar del concepto	8	16%
Total	50	100%

Los autores: Florentino. L. (2020) Resultados de encuesta ANÁLISIS E IMPLEMENTACIÓN DE LA PROGRAMACIÓN DE SOFTWARE -INSTITUTO TECNOLÓGICO FABIO AMABLE MOTA. [Tabla 4.6]. República Dominicana.

Según los resultados, el 72% de los encuestados sabe que es la programación, el 12% no sabe a qué se refiere el termino y el 16% ha escuchado hablar del concepto.

¿ Aplicas la logica matematica para la resolucion de problemas?

50 respuestas



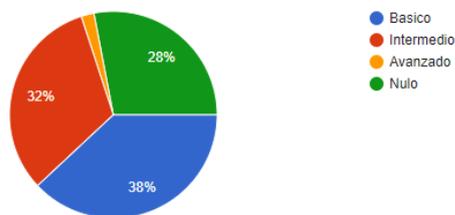
Alternativas	N	%
Siempre	12	24%
Casi Siempre	27	54%
Nunca	7	14%
No sé qué es la lógica matemática	4	8%
Total	50	100%

Los autores: Florentino. L. (2020) Resultados de encuesta ANÁLISIS E IMPLEMENTACIÓN DE LA PROGRAMACIÓN DE SOFTWARE -INSTITUTO TECNOLÓGICO FABIO AMABLE MOTA. [Tabla 4.7]. República Dominicana.

Según los resultados de la encuesta, el 24% de los encuestados siempre aplica la lógica matemática en la resolución de problemas, el 54% la utiliza casi siempre, el 14% no la utiliza nunca y el 8% no sabe que es la lógica matemática.

¿Que conocimientos de programacion posees?

50 respuestas



Alternativas	N	%
Básico	19	38%
Intermedio	16	32%
Avanzado	1	2%
Nulo	14	28%
Total	50	100%

Los autores: Florentino. L. (2020) Resultados de encuesta ANÁLISIS E IMPLEMENTACIÓN DE LA PROGRAMACIÓN DE SOFTWARE -INSTITUTO TECNOLÓGICO FABIO AMABLE MOTA. [Tabla 4.8]. República Dominicana.

Según los resultados obtenidos, el 38% de los encuestados tiene conocimientos básicos de programación, el 32% tiene conocimientos intermedios, sólo el 2% tiene conocimientos avanzados y el 28% no tiene ningún conocimiento de programación.

4.10 Conclusión Capítulo IV

En este capítulo hemos presentado el diseño de nuestra propuesta para la enseñanza de la programación como materia base en el Instituto Tecnológico Fabio Amable Mota y no solo limitada al área de informática.

Como ya hemos demostrado, la enseñanza de la programación puede ser de un gran beneficio para los estudiantes de nivel medio debido a las ventajas que ofrece y las capacidades que adquirirán los estudiantes al recibir esta enseñanza.

Hicimos una descripción de nuestra propuesta en el cual cubrimos todos los aspectos. Incluimos un plan de estudios diseñado para cada nivel de la educación media, en el cual presentamos las distintas herramientas a utilizar como pueden ser el contenido a enseñar, los lenguajes de programación a utilizar, los editores de texto, los equipos necesarios, etc. Nuestra propuesta de enseñanza se basa en el aprendizaje constructivista con el objetivo de que los estudiantes puedan construir sus propios conocimientos en base a pruebas y experiencias.

Los resultados de la encuesta que realizamos dieron un resultado favorable a favor de nuestra propuesta de que se implemente la programación como materia base en el Instituto Tecnológico Fabio Amable Mota, ya que a la mayoría de los estudiantes encuestados les gustaría que se les enseñe esta materia.

Es importante que los estudiantes tengan los conocimientos de acuerdo con la época en que vivimos actualmente, la cual es una época digital que demanda conocimientos tecnológicos. Es

por esta razón que hemos presentado esta propuesta, inicialmente para el Instituto Tecnológico Fabio Amable Mota, pero con la intención de que la misma pueda ser implementada en todas las escuelas del país en la educación media.

CONCLUSION

Conclusión

Como hemos visto hasta ahora, es importante tener ciertas habilidades de computación en estos tiempos en donde predomina la tecnología, y si se nos instruye o se nos enseña habilidades relacionadas a la computación desde muy jóvenes seremos capaces de desarrollarnos más como sociedad ya que se tendrán personas más capaces para el uso de nuevas herramientas y la aceptación de nuevas tecnologías. Por lo que la programación en las escuelas se vuelve una gran aliada por diferentes motivos que hemos visto hasta ahora.

Por esta razón en este trabajo de grado hemos utilizado el INSTITUTO FABIO AMABLE MOTA como objeto de estudio, en donde planteamos incorporar un nuevo plan de estudios que tenga presente la programación de software como una materia base, de forma que se pueda preparar mejor a los estudiantes, ya que esta materia les ayudara a desarrollarse mejor, a mejorar sus capacidades intelectuales y a adaptarse más a un mundo cambiante con tanta tecnología saliendo todos los días.

Tras la implementación de este nuevo programa de clase en el que se agrega una nueva materia y se le da mucho énfasis par que los estudiantes aprendan, esperamos poder mejorar el nivel de educación del país, aumentar las posibilidades de que los estudiantes al terminar el bachiller puedan obtener un buen empleo, y estas cosas de forma indirecta apoyan al mejoramiento de la sociedad ya que ayudan a generar entes productivos con muchas oportunidades en el mundo laboral. Además de que algunos estudiantes verán esto como una oportunidad de crear y presentar proyectos que sirvan para el mejoramiento de la sociedad o de su propia economía, ya que la programación en parte impulsa o más bien despierta el espíritu emprendedor.

Al concluir este trabajo, esperamos poder cumplir con las expectativas de nuestro asesor, los evaluadores y la universidad, pero también buscamos ayudar a aumentar los niveles de educación

actual al mejorar la calidad de los estudiantes con un pensum escolar más dinámico e interesante para estos, de forma que propuestas como esta puedan seguir surgiendo y que así exista una cultura de mejora continua en nuestra sociedad.

RECOMENDACIONES

Recomendaciones

Para una implementación exitosa del plan se recomienda tomar en cuenta los siguientes puntos de acción:

1. Mantener motivados a los estudiantes: esto puede lograrse presentando ejemplos reales de lo que pueden lograr al final del programa como:
 - Página web
 - Aplicación móvil
 - Videojuego
2. presentando las diversas áreas de trabajo en las cuales pudieran emplearse o emprender como:
 - a. Desarrollador móvil
 - b. Desarrollador web (backend, frontend o full stack)
 - c. Programador de videojuegos
3. estadística de empleabilidad y salarial en la carrera de informática.
4. Mantener el programa de clases actualizado: es de suma importancia dar seguimiento a las tendencias y cambios en el área de informática, debido a que es una disciplina en constante evolución, mantener el temario y recursos de aprendizaje actualizados garantiza que los estudiantes este aprendiendo temas de utilidad para utilizar tanto profesional como personalmente y también ayuda a que los estudiantes se mantengan motivados.

Es importante mantener actualizado los temas en cuanto a que lenguajes de programación son los más relevantes tanto local como internacionalmente, nuevas herramientas de desarrollo de

software, nuevas técnicas de enseñanza para jóvenes. Recomendamos una revisión de currículo cada 2 años

5. Crear un ambiente de pensamiento lógico y crítico: es importante que los estudiantes desarrollen confianza en sí mismos en sus habilidades de lógica y que sientan que están en un ambiente en donde aportar críticas constructivas sobre el trabajo de sus compañeros, algo presentado por el maestro y autocritica o autoevaluación del trabajo propio es algo que será tomado de buena forma.
6. Fomentar el aprendizaje por objetivos: esto garantiza que los estudiantes se mantengan motivados ya que saben que están trabajando para lograr una aplicación de utilidad.

En cuanto a ideas para escalar el proyecto, algunos temas que podrían agregarse al en un futuro son:

1. Ciencia de datos
 - a. Análisis de datos
 - b. Ciencia de datos
 - c. Big data
2. Fundamentos de Hacking ético
3. Internet de las cosas
 - a. programación de dispositivos Arduino o similares
4. Automatización de pruebas
 - a. Selenium
 - b. Katalon
 - c. Cucumber

BIBLIOGRAFIA

Bibliografía:

Arenas, L. (2004). Programación Orientada a Objetos en Java. México

Astudillo, K. (2013). Hacking Ético 101: Como Hackear Profesionalmente en 21 días o menos. Recuperado de <https://www.bibliadelprogramador.com/2017/06/hacking-etico-101-como-hackear.html>

Avison D. ; Fitzgerald, G. (1995). “Information Systems Development: Methodologies, Techniques, and Tools”. McGraw-Hill.

Aybar, V. (2012). Aplicaciones complementarias a Robocode que faciliten el aprendizaje de programación en escuelas secundarias. Recuperado de <http://sedici.unlp.edu.ar/handle/10915/47050>

Boehm, B. (1986). “A Spiral Model of Software Development and Enhancement.” ACM SIGSOFT Software Engineering Notes; ACM.

Buendía, R. & Fabian, J. (2013). “Seguridad informática”. McGraw-Hill.

Carretero, M. (2009) “¿Qué es el constructivismo?”. Federación de Educadores Bonaerenses.

Chávez, T. (2017). Programación Modular. México. Recuperado de http://ri.uaemex.mx/bitstream/handle/20.500.11799/69946/secme-19449_1.pdf?sequence=1

González, C. (2019). Estado del arte en la enseñanza del pensamiento computacional y la programación en la etapa infantil. Recuperado de <https://eds-b-ebscobhost-com.ezproxy.unapec.edu.do/eds/search/basic?vid=13&sid=c995bd88-8b7c-4195-813d-4b813d3ebaf8%40pdc-v-sessmgr02>

Gonzalez-Longatt, F. (2008) “Introducción a los sistemas de información: Fundamentos.” Universidad Veracruzana.

Guitert, M.; Giménez, F. (2008). “El trabajo en equipo en entornos virtuales: desarrollo metodológico”.

Hernández, C. (1999). “Manual de Creatividad publicitaria”. McGraw- Hill

Jiménez, C. (2019). Aprendiendo a través de la enseñanza: programa enfocado a la enseñanza de la lógica de programación desde etapas tempranas en escuelas públicas apoyado por aprendizaje-servicio. Recuperado de <https://eds-b-ebsohost-com.ezproxy.unapec.edu.do/eds/search/basic?vid=13&sid=c995bd88-8b7c-4195-813d-4b813d3ebaf8%40pdc-v-sessmgr02>

Joyanes Aguilar, L. (2000). Programación en C++: algoritmos, estructuras de datos y objetos. Recuperado de <https://eds-b-ebsohost-com.ezproxy.unapec.edu.do/eds/detail/detail?vid=7&sid=32d64fd9-55f6-4524-a6c8-018faa25e41b%40sessionmgr103&bdata=Jmxhbm9ZXMmc2l0ZT1lZHMtbGl2ZQ%3d%3d#db=cat07658a&AN=dbu.16285>

Kendall, K. & Kendall, J. (2011). “Análisis y diseño de Sistemas”. Pearson.

Llopis Pascual, F., Pérez López, E. & Ortuño Ortín, F., (2000). Introducción a la programación: algoritmos y C/C++. Recuperado de <https://eds-b-ebsohost-com.ezproxy.unapec.edu.do/eds/detail/detail?vid=10&sid=32d64fd9-55f6-4524-a6c8-018faa25e41b%40sessionmgr103&bdata=Jmxhbm9ZXMmc2l0ZT1lZHMtbGl2ZQ%3d%3d#AN=edshlc.014355434.4&db=edshlc>

Llorens Largo, F. (2002). Programación: formalización, análisis y reutilización de algoritmos matemáticos. Recuperado de <https://eds-b-ebsohost-com.ezproxy.unapec.edu.do/eds/detail/detail?vid=13&sid=32d64fd9-55f6-4524-a6c8-018faa25e41b%40sessionmgr103&bdata=Jmxhbm9ZXMmc2l0ZT1lZHMtbGl2ZQ%3d%3d#AN=edshlc.014355463.8&db=edshlc>

Martínez, M. (2018). Experiencias de programación en las escuelas. Recuperado de <https://eds-b-ebshost-com.ezproxy.unapec.edu.do/eds/search/basic?vid=13&sid=c995bd88-8b7c-4195-813d-4b813d3ebaf8%40pdc-v-sessmgr02>

Martel, S. (2002). “Extreme Programming: Rapid Development for Web-Based Applications”. IEEE Internet Computing.

Meinke, H. (2020) “Why Learn to Code? The Surprisingly Broad Benefits of Coding”. Rasmussen college online. Recuperado de rasmussen.edu.

Navarro Arribas, G. (2012). “Sistemas de Cortafuegos”. Universidad Abierta de Cataluña.

Ortega, J. (2004). Notas de Introducción al Lenguaje de Programación Java. México

Prieto, V. y Pan, R. (2006). Virus Informáticos. España. Recuperado de <http://sabia.tic.udc.es/docencia/ssi/old/2006-2007/docs/trabajos/08%20-%20Virus%20Informaticos.pdf>

Queiruga C. (2014). El juego como estrategia didáctica para acercar la programación a la escuela secundaria. Obtenido de <http://catalogo.info.unlp.edu.ar/meran/getDocument.pl?id=1352>

Queiruga, C. (2017). Programar en la escuela: nuevos desafíos en las aulas. Obtenido de <http://sedici.unlp.edu.ar/handle/10915/62341#?>

Rodríguez, J. (2011). Ética Hacker. Puerto Rico

Román González, M. (2014). “Aprender a Programar ‘Apps’ como enriquecimiento curricular en alumnado de alta capacidad”. Sociedad Española de Pedagogía; Bordón: Revista de Pedagogía.

Schwaber, K. (2004) “Agile Project Management with Scrum”. Microsoft Press.

Somerville, I. (1998) “Ingeniería del software”. Addison Wesley Iberoamericana, Wilmington.

Tapia, E. (2016). Informática VI (Programación e Implementación de Sistemas). México.

Recuperado de

http://fcasua.contad.unam.mx/apuntes/interiores/docs/20182/informatica/6/LI_1656_21117_A_Informatica_VI_Plan2016.pdf

Terres, O. (2015). Aprender a programar en las escuelas para mejora competencial en educación primaria. Obtenido de

<https://reunir.unir.net/bitstream/handle/123456789/3257/TERRES%20PEREZ%2C%20OLGA.pdf?sequence=1&isAllowed=y>

Trejos Buritacá, O. (1999). La Esencia de la Lógica de Programación. Recuperado de

http://biblioteca.inu.edu.sv/?wpfb_dl=263

Wing, J. (2006). “Computational Thinking”. ACM.

GLOSARIO

Glosario:

Programación: es el proceso en el cual se toma un algoritmo y se modifica en alguna notación o más bien un lenguaje de programación que forma que este algoritmo ya codificado en un lenguaje pueda ser compilado y ejecutado por una computadora.

Algoritmo: Es un conjunto de pasos secuenciales y ordenados que permiten lograr un objetivo. Que sean pasos secuenciales significa que deben ser ejecutados uno después de otro y que sean pasos ordenados quiere decir que deben llevar un orden casi obligatorio.

Variable: Una variable es como lo indica su nombre un valor cambiante en un sistema, técnicamente una variable es un campo de memoria al que se le puede cambiar su contenido cuantas veces sea necesario.

Operadores: Los operadores son signos que nos permiten expresar relaciones entre variables y/o constantes, relaciones de las cuales normalmente se desprende un resultado.

Pseudocódigo: es la representación textual de un algoritmo de manera que dicho texto se encuentre enmarcado en algunas normas técnicas que faciliten su posterior transcripción a un lenguaje de Programación.

Compilación: Es el proceso a través del cual el computador revisa que el programa que hemos digitado se ajuste a las reglas sintácticas de un determinado Lenguaje de Programación.

Objeto: se pueden definir como las unidades básicas de construcción, para la conceptualización, diseño o programación, esto es que son instancias agrupadas en clases con características en común y que son los atributos y procedimientos, conocidos como operaciones.

Métodos: Los métodos son en sí el comportamiento de un objeto básicamente es una acción que determina cómo deberá actuar un objeto, esto es un poco parecido a lo que se conoce como funciones en la programación estructurada.

Clases: Las clases se definen como el modelo en el que se define un conjunto de variables (atributos) y métodos (comportamiento) que son apropiados para operar con dichos datos. Todo objeto que se cree a partir de una clase se le denomina instancia de la clase.

Polimorfismo: esta es la capacidad de un tener varios métodos (varias formas de comportarse) con el mismo nombre, pero se implementan de forma diferente.

Herencia: Esta característica permite a una clase padre heredar sus atributos y métodos a una clase hija.

Encapsulamiento: esto se refiere a que la información de un objeto debe estar encapsulada por su comportamiento.

Abstracción: Tiene dos vertientes, una es la de descripción simplificada de un sistema que solo muestra los detalles significativos de un objeto y la otra vertiente es que se trata de la forma en la que abstraemos objetos del mundo real.

Metodologías de Desarrollo de Software: una metodología es una colección de procedimientos, técnicas, herramientas y documentos auxiliares que ayudan a los desarrolladores de software en sus esfuerzos por implementar nuevos sistemas de información.

Programa de Software: Software se refiere al conjunto de instrucciones preparadas para ser interpretadas por una computadora que realiza procesamiento electrónico de datos.

Paradigma de programación: Un paradigma de programación es una manera o estilo de programación de software.

Programación estructurada: La programación estructurada es la técnica de desarrollo de programas de la forma más clara posible haciendo uso de tres estructuras de control: la secuencia, la selección y la interacción.

Programación Orientada a objetos: La programación Orientada a objetos se define como un paradigma de la programación, donde se organiza el código en unidades denominadas clases, de las cuales se crean objetos que se relacionan entre sí para conseguir los objetivos de las aplicaciones.

Programación Concurrente: La programación concurrente se define como el área de la ciencia de computación que trata sobre las metodologías, lenguajes, técnicas y herramientas de programación necesarias para la construcción de programas reactivos.

Programación Declarativa: La programación declarativa es un estilo de programación en el que el programador no dice cómo llegar a obtener la solución de un problema, sino “qué” define esa solución.

Método Cascada o Waterfall: El modelo en cascada es un proceso de desarrollo secuencial, en el que el desarrollo de software se concibe como un conjunto de etapas que se ejecutan una tras otra.

Metodología Ágil: la metodología ágil es una metodología de desarrollo de software que se basa en valores, principios y prácticas básicas.

Scrum: es un marco de trabajo que define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto.

Método Espiral: El modelo de desarrollo en Espiral es una combinación entre el modelo waterfall y un modelo por iteraciones

Método Prototipo: Un prototipo es un sistema que funciona desarrollado con la finalidad de probar ideas y suposiciones relacionadas con el nuevo sistema.

Método RAD: El desarrollo rápido de aplicaciones o RAD) es un proceso de desarrollo de software que comprende el desarrollo interactivo, la construcción de prototipos y el uso de utilidades CASE.

Creatividad: La creatividad se trata de una capacidad existente en todos los seres humanos, utilizada para la solución de problemas y que precisa de realidades ya existentes.

Razonamiento Computacional: el pensamiento computacional implica resolver problemas, diseñar sistemas y comprender el comportamiento humano, haciendo uso de los conceptos fundamentales de la informática.

Constructivismo: el constructivismo propone un paradigma donde el proceso de enseñanza se percibe y se lleva a cabo como un proceso dinámico, participativo e interactivo del sujeto, de modo que el conocimiento sea una auténtica construcción operada por la persona que aprende (por el «sujeto cognoscente»). El constructivismo en pedagogía se aplica como concepto didáctico en la enseñanza orientada a la acción.

Malware: es la abreviatura de “Malicious software” o en español programa maligno, término que engloba a todo tipo de programa o código informático malicioso cuya función es dañar un sistema o causar un mal funcionamiento.

Anti-Malware: El software antimalware bloquea y elimina de forma efectiva y eficiente el malware. Este software incluye el núcleo de protección de las suites de seguridad, aunque obviando algunos extras que no resultan necesarios para combatir el malware, como el control parental o administradores de contraseñas.

Amenaza: En seguridad informática, un ordenador o red se considera bajo amenaza cuando acarrea persistentes vulnerabilidades de software, con lo cual incrementa la probabilidad de un ataque maligno.

El análisis FODA: es una herramienta de planificación estratégica, diseñada para realizar un análisis interno (Fortalezas y Debilidades) y externo (Oportunidades y Amenazas) en la empresa.

Cable Par Trenzado: este cable es de par trenzado o cable UTP, este tipo de cable cuenta con dos conductos aislados eléctricos y están entrelazados para anular las interferencias de las fuentes externas y diafonía de los cables adyacentes. Los cables pares trenzados cuentan con 4 pares de hilos de cables que, en total con 8 hilos, por el cual data de manera nativa con un flujo de cargas eléctricas.

Cableado Estructurado: esta es la forma por la cual se realiza un cableado de un edificio o una serie de edificios, este tipo de cable permite interconectar equipos activos en diferentes áreas permitiendo la integración de los diferentes servicios que dependen del tendido de cables como datos, telefonía, control, entre otros, estas deben cumplir una serie de normas y estándares antes de ser llamado cableado estructurado.

Data Center: es el espacio donde se compone y se gestiona todos los servicios y equipos de TI que utiliza la empresa para facilitar toda la gestión de ella misma

IP: estas siglas significan Internet Protocolo y este protocolo asigna dentro de una RED un número de identidad a los diversos dispositivos que ingresan a ella. con este número los diferentes dispositivos dentro de la RED son identificables y a la hora de ser enviado un paquete a alguno de estos equipos no se pierde.

Router: Este es un dispositivo de RED que permite el enrutamiento de paquetes en una RED o diferentes Redes. Servidor: se les llama así a las computadoras con capacidades superiores a la media con datos, programas u archivos conectada a la RED o varias Redes, destinada a brindar servicio a las demás computadoras o equipos que soliciten.

Spyware: este es un software malicioso que recolecta y envía información de algún usuario a otro sin previo aviso o consentimiento.

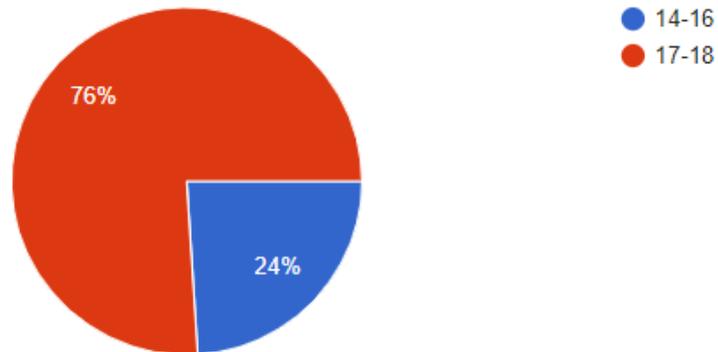
Switch: Este es un dispositivo digital lógico de interconexión que conecta diferentes equipos que opera en la capa de enlace de datos del modelo OSI.

ANEXOS

Anexo. 1 – ENCUESTA

¿Cual es su edad ?

50 respuestas



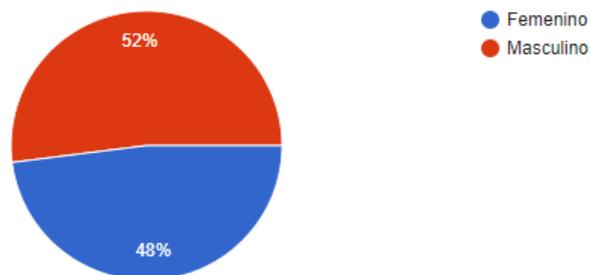
Alternativas	N	%
14-16	12	24%
17-18	38	76%
Total	50	100%

Los autores: Florentino. L. (2020) Resultados de encuesta ANÁLISIS E IMPLEMENTACIÓN DE LA PROGRAMACIÓN DE SOFTWARE -INSTITUTO TECNOLÓGICO FABIO AMABLE MOTA. República Dominicana.

Según las edades de las personas encuestadas tenemos que el 24% tiene entre 14-16 años y el 76% tiene edades que rondan entre los 17-18 años.

¿Cual es su sexo ?

50 respuestas



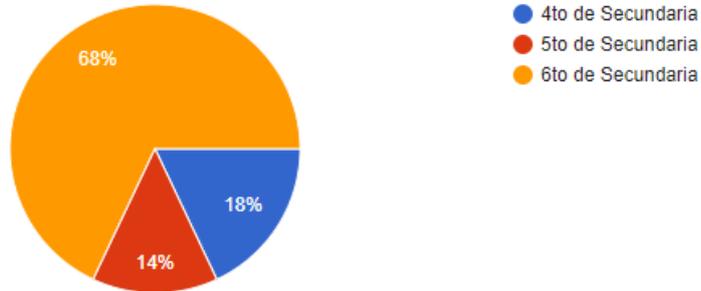
Alternativas	N	%
Masculino	26	52%
Femenino	24	48%
Total	50	100%

Los autores: Florentino. L. (2020) Resultados de encuesta ANÁLISIS E IMPLEMENTACIÓN DE LA PROGRAMACIÓN DE

Según el sexo de las personas encuestadas, tenemos que el 52% corresponde al sexo masculino, mientras que el 48% de ellas corresponde al sexo femenino

¿Que curso estas cursando actualmente ?

50 respuestas



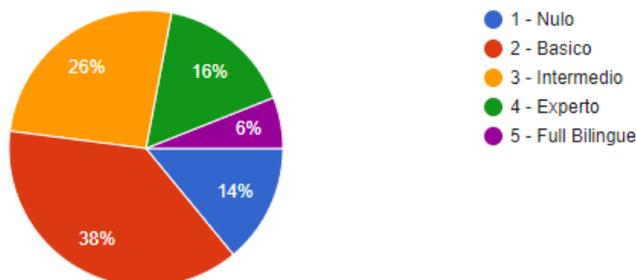
Alternativas	N	%
4to de Secundaria	9	18%
5to de Secundaria	7	14%
6to de Secundaria	34	68%
Total	50	100%

Los autores: Florentino. L. (2020) Resultados de encuesta ANÁLISIS E IMPLEMENTACIÓN DE LA PROGRAMACIÓN DE SOFTWARE -INSTITUTO TECNOLÓGICO FABIO AMABLE MOTA. República Dominicana.

Según el curso cursado actualmente por los encuestados, el 18% está en 4to de Secundaria, el 14% está en 5to de Secundaria y el 68%; siendo este el más alto porcentaje, está cursando el 6to de Secundaria.

En una escala del 1-5, ¿Cual es tu dominio del idioma ingles ?

50 respuestas



Alternativas	N	%
Nulo	7	14%

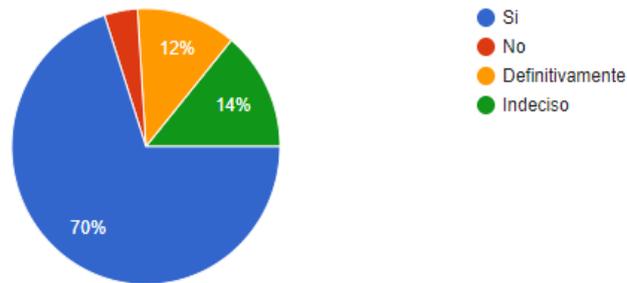
Básico	19	38%
Intermedio	13	26%
Experto	8	16%
Full Bilingüe	3	6%
Total	50	100%

Los autores: Florentino. L. (2020) Resultados de encuesta ANÁLISIS E IMPLEMENTACIÓN DE LA PROGRAMACIÓN DE SOFTWARE -INSTITUTO TECNOLÓGICO FABIO AMABLE MOTA. República Dominicana.

Según los resultados el 14% de los encuestados no tiene ningún conocimiento del idioma inglés, el 38% tiene conocimientos básicos, el 26% tiene conocimientos intermedios, el 16% es experto en el idioma y el 6% es full Bilingüe.

¿Te gustaría aprender programación como materia base en tu instituto?

50 respuestas



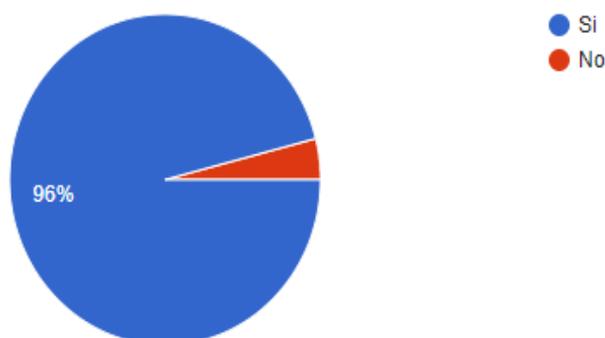
Alternativas	N	%
Si	35	70%
No	2	4%
Definitivamente	6	12%
Indeciso	7	14%
Total	50	100%

Los autores: Florentino. L. (2020) Resultados de encuesta ANÁLISIS E IMPLEMENTACIÓN DE LA PROGRAMACIÓN DE SOFTWARE -INSTITUTO TECNOLÓGICO FABIO AMABLE MOTA. República Dominicana.

Según los resultados, al 70% de los encuestados le gustaría aprender programación como materia base, al 4% no le gustaría, al 12% definitivamente le gustaría y el 14% está indeciso si desea aprender programación como materia base.

¿Entiendes que es necesario la creación de nuevas materias en el area de informatica?

50 respuestas



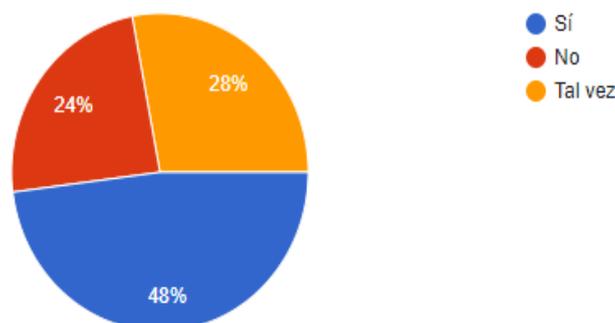
Alternativas	N	%
Si	48	96%
No	2	4%
Total	50	100%

Los autores: Florentino. L. (2020) Resultados de encuesta ANÁLISIS E IMPLEMENTACIÓN DE LA PROGRAMACIÓN DE SOFTWARE -INSTITUTO TECNOLÓGICO FABIO AMABLE MOTA. República Dominicana.

Según los resultados, el 96% de los encuestados entiende que si es necesario la creación de nuevas materias en el área de informática, mientras que el 4% entiende que no es necesario la creación de estas.

¿Planeas seguir estudiando algo relativo al desarrollo de software al salir del instituto?

50 respuestas



Alternativas	N	%
Si	24	48%
No	12	24%
Tal Vez	14	28%
Total	50	100%

Los autores: Florentino. L. (2020) Resultados de encuesta ANÁLISIS E IMPLEMENTACIÓN DE LA PROGRAMACIÓN DE SOFTWARE -INSTITUTO TECNOLÓGICO FABIO AMABLE MOTA. República Dominicana.

Según los resultados, al salir del Instituto el 48% de los encuestados planea seguir estudiando algo relativo al desarrollo de software, el 24% no, mientras que un 28% tal vez estudie algo relativo al desarrollo de software.



UNAPEC
UNIVERSIDAD APEC

DECANATO DE INGENIERÍA E INFORMÁTICA

Anteproyecto de Trabajo de Grado

Sustentantes:

Alan E. Rodríguez Pérez / 2013-1209

Luis J. Florentino Veloz / 2006-0336

Tema:

Análisis e Implementación de la Programación de Software como asignatura básica en las escuelas públicas y privadas a nivel de educación media, en la ciudad de Santo Domingo durante el periodo Septiembre-Diciembre 2020. Caso de Estudio: Instituto Tecnológico
Fabio Amable Mota

Asesor:

Ing. Freddy Jiménez

Fecha:

Diciembre 2020

Santo Domingo, Rep. Dom.

Contenido

I - Título del tema.....	3
II -Introducción.....	3
III - Justificación.....	5
IV - Delimitación del Tema y Planteamiento del Problema	5
IV.I - Delimitación del Tema.....	5
IV.II - Planteamiento del problema.....	5
V - Objetivos.....	6
V.I - Objetivo General	6
V.II - Objetivos Específicos.....	6
VI - Marco Teórico	6
VI.I - Marco Teórico Referencial	6
VI.II - Marco Conceptual.....	9
VII - Hipótesis	13
VIII – Diseño metodológico: metodología y técnicas de investigación cuantitativa y/o cualitativa.....	13
VIII.I – Enfoque.....	13
VIII.II –Tipo de estudio	13
VIII.III – Método de investigación	13
VIII.IV – Fuente y técnicas para recolección de información	14
VIII.V – Análisis y Tratamiento de la Información.....	14
X - Esquema Preliminar.....	17

I - Título del tema

Análisis e Implementación de la Programación de Software como asignatura básica en las escuelas públicas y privadas a nivel de educación media, en la ciudad de Santo Domingo durante el periodo Septiembre-Diciembre 2020. Caso de Estudio: Instituto Tecnológico Fabio Amable Mota.



Rodríguez Pérez A. (2020). Instituto Fabio Amable Mota



Florentino Veloz, L. (2020). Instituto Fabio Amable Mota

II- Introducción

El mundo ha cambiado mucho desde el siglo anterior hasta la fecha, ahora nos encontramos viviendo en una época digital en la que no saber manipular las nuevas tecnologías no es opción para muchas personas, y en muchas profesiones se pide cada vez más ser más diestro con el uso la tecnología actual y la demanda de profesionales con más capacidad de análisis y resolución de problemas a aumentado. Por lo que las instituciones educativas deben ser capaces de educar a estudiantes más capaces en muchos sentidos, actualmente es necesario que los estudiantes de las escuelas se gradúan teniendo conocimiento de las famosas TICS (Tecnologías de Información y Comunicación) para así poder desempeñarse mejor y tener más oportunidad en la sociedad al terminar sus estudios.

El poder educar jóvenes con una buena capacidad de análisis y capacidad de resolver problemas puede ser difícil si no se usa el método apropiado, por eso la programación de software brinda una solución a esta situación, el aprender a programar siendo joven puede aportar muchos beneficios a los estudiantes, porque la programación se trata precisamente de analizar y resolver problemas usando la lógica, además de que es un método en el que se está creando algo, y una de las mejores formas de aprender es mediante la creación.

Para la implementación de la programación como materia en escuelas, en los primeros niveles nos enfocaremos en el uso de herramientas didácticas como MakeCode de Microsoft, Swift Playgrounds de Apple. En cursos con estudiantes de más edad (13 años en adelante) se usaron ya lenguajes de programación reales como Python. Básicamente se usarán herramientas y métodos que permiten al estudiante aprender mientras va creando sus propios programas, lo que irá aumentando la creatividad, capacidad de análisis y pensamiento crítico del estudiante.

El Instituto Tecnológico Fabio Amable Mota, enseña programación solo a los estudiante de informática, pero proponemos enseñar esto en todos los niveles de educación media, y en todas las áreas técnicas, con el objetivo final de que todos los estudiante graduados de bachiller puedan desarrollar su habilidad para resolver problemas tengan un buen nivel de conocimiento de programación de software, que le permita hacer carrera en esta área si así lo desea.

III- Justificación

La programación como materia en escuelas a nivel medio es conveniente en muchos sentidos (social, económico, etc.), pero lo principal es en la parte educativa, con esta materia los estudiantes adquieren habilidades como pensamiento crítico, resolución de problemas, desarrolla el espíritu emprendedor al crear programas, entre otras.

Además de obtener beneficios, al terminar la escuela media o bachillerato tendrán un conocimiento por el que se paga muy bien en las empresas y es un área con mucho desarrollo en la actualidad así que esta materia en las escuelas capacita a los jóvenes de mejor manera para el mundo real.

En República Dominicana una gran parte de los estudiantes no tienen la formación necesaria en cuanto a contenido en el pensum escolar. Esto deja lagunas en el pensamiento lógico de ellos, con la entrada de la programación como parte integral de su formación, dichas lagunas se pueden mitigar.

Esta investigación ayudará a tener los puntos claros cuando nos referimos a la programación de software en escuelas de nivel medio. Al tener documentación como esta se puede entender por qué la incorporación de estas materias en escuelas de nivel medio puede ser de beneficio para las estudiantes, por ende también para la sociedad ya que esta materia ayuda bastante a crear más jóvenes capaces que aporten valor a la sociedad dominicana.

IV - Delimitación del Tema y Planteamiento del Problema

IV.I - Delimitación del Tema

El Instituto Tecnológico Fabio Amable Mota., ubicado en la Prolongación Avenida Venezuela #2, Santo Domingo Este, 11906, Rep. Dom. en el periodo mayo-agosto 2020.

IV.II - Planteamiento del problema

El Instituto Tecnológico Fabio Amable Mota no contempla actualmente con la programación de software como materia base para todos sus estudiantes, solamente está incluida en el bachillerato técnico en desarrollo y administración de aplicaciones informáticas. Bajo el pensum actual los estudiantes no cuentan con las herramientas necesarias para dominar los conceptos básicos de la programación en general

La programación es necesaria en todos los niveles, no solamente en el área técnica de informática. Las demás modalidades del recinto necesitan incorporar esta materia para ayudar a sus estudiantes a desarrollar su mente y la lógica detrás de cada proyecto que quieran emprender. Al incorporar esta materia podemos mejorar la calidad educativa del instituto por ende mejorará el nivel académico de sus estudiantes.

V - Objetivos

V.I - Objetivo General

Implementar la enseñanza de la de programación de software como materia base a nivel medio, en el INSTITUTO FABIO AMABLE MOTA, en la ciudad de Santo Domingo, República Dominicana, en el periodo septiembre-diciembre 2020.

V.II - Objetivos Específicos

- Mostrar quienes serían los beneficiados al integrar la programación en las escuelas.
- Determinar las herramientas y métodos que se usarán para enseñar programación de software en las escuelas.
- Evaluar y presentar la eficiencia de las herramientas y métodos que se usarán para enseñar programación en las escuelas.
- Evaluar y describir el impacto económico, social y educativo que tendría la enseñanza de programación de software en las escuelas en la sociedad dominicana.

VI - Marco Teórico

VI.I - Marco Teórico Referencial

Zabala, G., Morán, R. , Blanco, S., (2013), en su investigación “*Una propuesta de enseñanza de programación en escuela media mediante el desarrollo de videojuegos con Etoys*” se proponen estudiar: Este es un documento plantea la enseñanza de la programación en escuela media pero de un modo distinto a lo usual.

Se explica que es la programación y los desafíos que puede tener el aprender a programar por lo que los autores sugieren utilizar un método mucho más llamativo y fácil para aprender a programar que es enseñando a programar videojuegos con un framework llamado Etoys, en

donde el estudiante tiene la posibilidad de aprender lógica de programación y de crear cosas que puedan ser divertidas, lo que incentiva a muchos jóvenes a seguir en este camino.

Terres, O., (2015), en su investigación *“Aprender a programar en la escuela para la mejora competencial en educación primaria”* se propone estudiar: La incorporación de la enseñanza de programación de software en escuelas primarias. Aprender a programar puede aportar a los niños las competencias en el medio digital que necesitan para poder llegar a ser sujetos con sólidos conocimientos en ese medio ya que es parte fundamental de nuestra realidad.

La idea esencial de esta investigación es que los estudiantes pasen de ser simples usuarios a ser también creadores, y esto es por lo que puede aportar este tipo de aprendizajes en el desarrollo de otras capacidades cognitivas y también por la demanda creciente de profesionales en el sector tecnologías de la información.

Fábrega, R., Fábrega F., B Blair, A., (2016), en su investigación *“La enseñanza de Lenguajes de Programación en la Escuela: ¿Por qué hay que prestarle atención?”* se proponen estudiar: La importancia de los programadores en la sociedad actual, y también la creciente necesidad de estos profesionales debido a los crecientes medios digitales. También destaca los desafíos de la enseñanza de la programación y métodos para facilitar su aprendizaje. Por lo que plantean que la programación debe ser enseñada en escuelas para así tener más personas interesadas y preparadas para entrar en esta área.

Díaz Tejera, K., Fierro Martín, E., Muñoz, M., (2018), en su investigación *“La enseñanza de la programación. Una experiencia en la formación de profesores de Informática”* se proponen estudiar: La enseñanza de la programación en escuelas tiene como objetivo esencial la formación y desarrollo de habilidades por parte de los estudiantes, que posibiliten la resolución de problemas del ámbito escolar, profesional o de la vida práctica teniendo en cuenta los recursos que brindan los diferentes lenguajes de programación.

En la formación de profesores de Informática, este proceso posee características que lo distinguen. El profesor en formación debe desarrollar la habilidad y, a la vez apropiarse de los procedimientos necesarios para poder dirigir este proceso en la escuela.

Santoyo Pardo, J., (2016), en su investigación *“INNOVACIÓN DE VIDEOJUEGOS CON EL SOFTWARE SCRATCH PARA FORTALECER LAS HABILIDADES DE PENSAMIENTO CREATIVO EN ESTUDIANTES DE TECNOLOGÍA INFORMÁTICA DEL GRADO NOVENO*

DEL INSTITUTO AGRÍCOLA DE ALTO JORDÁN DE VÉLEZ SANTANDER COLOMBIA PARA EL AÑO 2016” se propone estudiar: La influencia de la Innovación de videojuegos con el software Scratch en las habilidades de pensamiento creativo de estudiantes de tecnología informática. Este documento está fundamentado en que la computación creativa favorece el desarrollo de conexiones que implican creatividad e imaginación.

Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado, (2017) en su investigación “*LA ENSEÑANZA DE PROGRAMACIÓN EN LOS CENTROS ESCOLARES DEL REINO UNIDO*” se propone estudiar: La educación como método proporciona a los jóvenes el conocimiento y la comprensión de cómo funciona el mundo y oportunidades para embarcarse en experiencias formativas estimulantes. En las Ciencias, las Matemáticas y la Programación existe una fuerte alineación entre las necesidades intelectuales y culturales del individuo y las necesidades económicas del país. Los empleadores prevén el aumento de la cantidad de carreras profesionales que dependen de las habilidades de Programación. Muchos jóvenes que comienzan actualmente la escuela podrían desempeñar trabajos que actualmente no existen.

El diseño de un buen currículo educativo es el primer paso necesario para garantizar que el alumnado salga de la escuela y la universidad bien formados para emprender carreras profesionales exitosas y convertirse en ciudadanos inteligentes y responsables.

Para ello, los gobiernos y los agentes educativos deben asegurarse de que el currículo abarque los tres aspectos claves de la Programación, incluida la Alfabetización Digital, las Tecnologías de la Información y las Ciencias de la Computación, y que los niños y niñas comiencen a estudiar Programación a la edad más temprana posible.

Queiruga, C., Fava, L., Gómez, S., Miyuki Kimura, I., Brown Bartneche, M., (2014), en su investigación “*El juego como estrategia didáctica para acercar la programación a la escuela secundaria*” se proponen: El objetivo central es crear escenarios educativos innovadores en la escuela secundaria en torno a la enseñanza de programación, atendiendo el rol central que ocupan los videojuegos en la vida cotidiana de los jóvenes y a la programación como habilidad requerida para poder crear con los medios digitales.

Diaz Sagástegui, G., Lozano Cubas, R., (2018), en su investigación “*USO DE LAS APLICACIONES CODE.ORG Y SCRATCH PARA EL APRENDIZAJE DE PROGRAMACIÓN EN LOS ESTUDIANTES DEL 5° Y 6° GRADO DE EBR DEL C.E. N° 82099 DE LA*

PROVINCIA DE SAN PABLO" se proponen estudiar: El uso de aplicaciones de Code.org y Scratch permiten el aprendizaje de programación en los estudiantes. Se pretende investigar la influencia que puede tener el uso de estas herramientas en los estudiantes, y a probar la eficiencia de estas.

Román , M., (2016), en su investigación "*CÓDIGOALFABETIZACIÓN Y PENSAMIENTO COMPUTACIONAL EN EDUCACIÓN PRIMARIA Y SECUNDARIA: VALIDACIÓN DE UN INSTRUMENTO Y EVALUACIÓN DE PROGRAMAS*" se propone estudiar: El autor busca responder una pregunta general con esta investigación, ¿es posible y deseable incorporar la códigoalfabetización en el sistema educativo?, en este documento se plantea el término códigoalfabetización para referirse a la enseñanza de la programación de software.

Como parte de la investigación se plantea ser capaz de manejar el lenguaje de las computadoras emerge como una habilidad indispensable, un nuevo alfabetismo, que nos permite participar de manera plena y efectiva en la realidad digital que nos rodea.

Wolovick, N. y Martínez, M., (2016), en su investigación "*Enseñar a Programar y Programar para Aprender*" se proponen estudiar: Los autores de este documento busca responder qué se debe enseñar en las escuelas como parte de las ciencias computacionales, se destaca a la programación como materia fundamental para enseñar en escuelas. La enseñanza de la programación representa un paradigma nuevo de introducción de la computadora en la escuela y nos interpela a pensar nuevas formas de introducir las tecnologías en las aulas, más allá del uso de las TIC.

VI.II - Marco Conceptual

Algoritmo: Es un conjunto de pasos secuenciales y ordenados que permiten lograr un objetivo. Que sean pasos secuenciales significa que deben ser ejecutados uno después de otro y que sean pasos ordenados quiere decir que deben llevar un orden cuasi-obligatorio. (Trejos Buritacá, O. (1999). La Esencia de la Lógica de Programación. Pág. 18)

Ejemplo:

Algoritmo Adquisicion_Libro

Inicio

1. *Saber cuál es el libro que se quiere adquirir*
2. *Desplazarnos hacia una librería*
3. *Preguntar si tienen el libro que necesitamos*
4. *Si lo tienen*
adquirirlo y Parar allí (dentro de este algoritmo)
Si no lo tienen
ir al paso 2

Fin

(Trejos, O. 1999 La esencia de lógica de la programación, Pág. 19)

Variable: Informalmente algo variable es algo que puede cambiar de un momento a otro. Técnicamente una variable es un campo de memoria al que se le puede cambiar su contenido cuantas veces sea necesario. (Trejos Buritacá, O. (1999). La Esencia de la Lógica de Programación. Pág. 27)

Operadores: Los operadores son signos que nos permiten expresar relaciones entre variables y/o constantes, relaciones de las cuales normalmente se desprende un resultado. (Trejos Buritacá, O. (1999). La Esencia de la Lógica de Programación. Pág. 35)

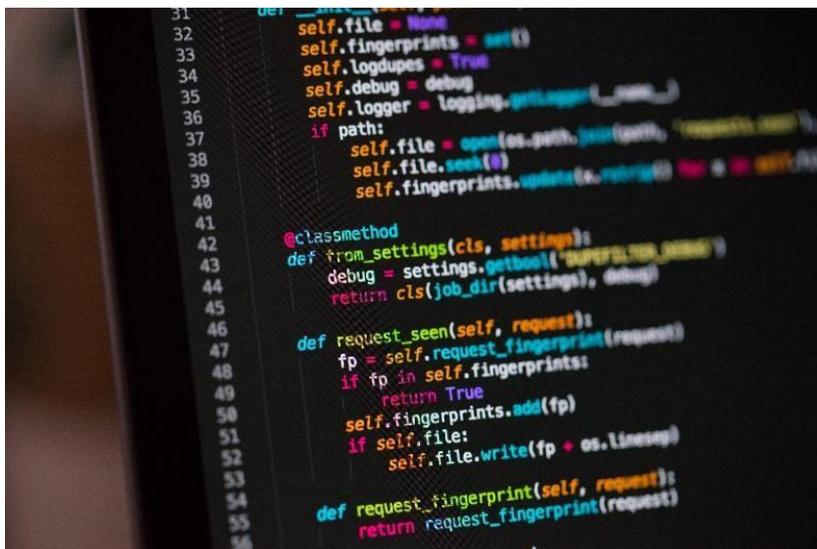
PseudoCódigo: Es la representación textual de un algoritmo de manera que dicho texto se encuentre enmarcado en algunas normas técnicas que faciliten su posterior transcripción a un lenguaje de Programación. (Trejos Buritacá, O. (1999). La Esencia de la Lógica de Programación. Pág. 66)



Cabanes, N. (2019). Introducción a la programación – Pseudocódigo. [Ilustración]. Obtenido de: <https://www.aprendeaprogramar.com/cursos/verApartado.php?id=2003>

Programa: Podemos definir un programa como un conjunto ordenado de instrucciones que guían a la computadora para que realice una tarea específica a través de sus operaciones. (Llopis Pascual, F., Pérez López, E., & Ortuño Ortín, F., (2000). Introducción a la programación: algoritmos y C/C++. Pág. 15)

Programación: Podemos llamar programación al proceso de tomar un algoritmo y modificarlo en una notación, un lenguaje de programación, de modo que pueda ser ejecutado por una computadora. Aunque existen muchos lenguajes de programación y muchos tipos diferentes de computadoras, el primer paso es la necesidad de tener una solución. Sin un algoritmo no puede haber un programa.



```
31
32 self.file = None
33 self.fingerprints = set()
34 self.logdupes = True
35 self.debug = debug
36 self.logger = logging.getLogger(__name__)
37 if path:
38     self.file = open(os.path.join(path, "requests.log"),
39                     "a")
40     self.fingerprints.update(os.listdir(path))
41
42 @classmethod
43 def from_settings(cls, settings):
44     debug = settings.getbool("DEBUG")
45     return cls(job_dir(settings), debug)
46
47 def request_seen(self, request):
48     fp = self.request_fingerprint(request)
49     if fp in self.fingerprints:
50         return True
51     self.fingerprints.add(fp)
52     if self.file:
53         self.file.write(fp + os.linesep)
54
55 def request_fingerprint(self, request):
56     return request_fingerprint(request)
```

Ried, C. (2019). [Illustration]. Beginning Python Programming — Part 1. Obtenido de: <https://medium.com/better-programming/beginning-python-programming-part-1-variables-constants-and-types-1199da1572c8>

Compilación: Es el proceso a través del cual el computador revisa que el programa que hemos digitado se ajuste a las reglas sintácticas de un determinado Lenguaje de Programación. (Trejos Buritacá, O. (1999). La Esencia de la Lógica de Programación. Pág. 22)

Software: Es un conjunto de instrucciones de programa detalladas que controlan y coordinan los componentes hardware de una computadora y controlan las operaciones de un sistema informático. (Joyanes Aguilar, L. (2000). Programación en C++: algoritmos, estructuras de datos y objetos. Pág. 31)

Lenguaje de programación: lenguaje de programación es el idioma que utiliza la persona para comunicarse con el ordenador y juega, por lo tanto, un papel fundamental en la programación. (Llorens Largo, F. (2002). Programación: formalización, análisis y reutilización de algoritmos matemáticos)



Belius, (2017). Elegir lenguaje de programación para desarrollar mi profesión. [Ilustración]. Obtenido de: <http://developeando.net/elegir-lenguaje-de-programacion/>

IDE: un IDE (Integrated Development Environment) es una aplicación que se utiliza para facilitar al programador el desarrollo de software. Este sistema de desarrollo de software combina diversas herramientas de desarrollo en una interfaz gráfica.



Guerrero, N. (2020). Los mejores IDE de Java. [Ilustración]. Obtenido de: <https://www.programaenlinea.net/los-mejores-ide-de-java/>

VII - Hipótesis

Aprender a programar puede ser de mucho beneficio para los jóvenes que están cursando el bachillerato, y más ahora en esta época digital, la programación los ayudará a ser más creativos, a asumir compromisos cuando se desarrolla un proyecto, a aprender sobre el trabajo colaborativo, además de que están aprendiendo una habilidad muy demandada en la actualidad, con la programación estarán mejor preparados para la sociedad una vez se gradúen.

VIII – Diseño metodológico: metodología y técnicas de investigación cuantitativa y/o cualitativa

VIII.I – Enfoque

Utilizaremos como enfoque para esta investigación los métodos de investigación cuantitativa y la investigación cualitativa.

VIII.II – Tipo de estudio

Descriptivo: El tipo de estudio será el descriptivo, en el que vamos a describir cómo se implementará la enseñanza de programación de software, que herramientas y métodos que utilizarán para este propósito, también se describirán los beneficios que adquirirán los estudiantes al aprender programación como parte del pensum escolar.

Exploratorio: también usaremos el tipo de estudio exploratorio, que se usará para recolectar información de libros, artículos de periódicos, artículos de revistas y por último trabajos de grados que tengan relación al nuestro.

VIII.III – Método de investigación

Método analítico: este método nos ayudará a identificar las mejores técnicas y herramientas a usar para incluir la programación de software en el pensum de las escuelas, en este caso el Instituto Tecnológico Fabio Amable Mota.

Método descriptivo: este método se utilizará para poder brindar una descripción bien detallada de todo lo relacionado con la implementación de la enseñanza de programación de software en el Instituto Tecnológico Fabio Amable Mota y sus beneficios.

VIII.IV – Fuente y técnicas para recolección de información

Recolección de información: esta se obtendrá de libros de programación, educación u otros temas relacionados a nuestro trabajo de grado, también de trabajos de grado relacionados al nuestro y de artículos (de periódicos, revistas y web) que podamos encontrar.

Entrevistas: se harán entrevistas a maestros del centro educativo, esto es, con el fin de poder obtener información fidedigna sobre el pensum escolar.

VIII.V – Análisis y Tratamiento de la Información

Para el análisis y el tratamiento de la información previamente recolectada, usaremos descripciones, definiciones, gráficos y diagramas, los dos últimos son para una mejor ilustración y presentación de esta, lo que a su vez permite una comprensión más rápida y ordenada.

IV - Fuente de Documentación

Martínez, M. (2018). Experiencias de programación en las escuelas. Obtenido de <https://eds-b-ebSCOhost-com.ezproxy.unapec.edu.do/eds/search/basic?vid=13&sid=c995bd88-8b7c-4195-813d-4b813d3ebaf8%40pdc-v-sessmgr02>

Jiménez, C. (2019). Aprendiendo a través de la enseñanza: programa enfocado a la enseñanza de la lógica de programación desde etapas tempranas en escuelas públicas apoyado por aprendizaje-servicio. Obtenido de <https://eds-b-ebSCOhost-com.ezproxy.unapec.edu.do/eds/search/basic?vid=13&sid=c995bd88-8b7c-4195-813d-4b813d3ebaf8%40pdc-v-sessmgr02>

Trejos Buritacá, O. (1999). La Esencia de la Lógica de Programación. Obtenido de http://biblioteca.inu.edu.sv/?wpfb_dl=263

González, C. (2019). Estado del arte en la enseñanza del pensamiento computacional y la programación en la etapa infantil. Obtenido de <https://eds-b-ebSCOhost-com.ezproxy.unapec.edu.do/eds/search/basic?vid=13&sid=c995bd88-8b7c-4195-813d-4b813d3ebaf8%40pdc-v-sessmgr02>

Aybar, V. (2012). Aplicaciones complementarias a Robocode que faciliten el aprendizaje de programación en escuelas secundarias. Obtenido de <http://sedici.unlp.edu.ar/handle/10915/47050>

Queiruga, C. (2017). Programar en la escuela: nuevos desafíos en las aulas. Obtenido de <http://sedici.unlp.edu.ar/handle/10915/62341#?>

Terres, O. (2015). Aprender a programar en la escuelas para mejora competencial en educación primaria. Obtenido de <https://reunir.unir.net/bitstream/handle/123456789/3257/TERRES%20PEREZ%2C%20OLGA.pdf?sequence=1&isAllowed=y>

Joyanes Aguilar, L. (2000). Programación en C++ : algoritmos, estructuras de datos y objetos. Obtenido de <https://eds-b-ebsohost-com.ezproxy.unapec.edu.do/eds/detail/detail?vid=7&sid=32d64fd9-55f6-4524-a6c8-018faa25e41b%40sessionmgr103&bdata=Jmxhbmc9ZXMmc2l0ZT1lZHMtbG12ZQ%3d%3d#db=cat07658a&AN=dbu.16285>

Queiruga C. (2014). El juego como estrategia didáctica para acercar la programación a la escuela secundaria. Obtenido de <http://catalogo.info.unlp.edu.ar/meran/getDocument.pl?id=1352>

Llopis Pascual, F., Pérez López, E., & Ortuño Ortín, F., (2000). Introducción a la programación: algoritmos y C/C++. Obtenido de <https://eds-b-ebsohost-com.ezproxy.unapec.edu.do/eds/detail/detail?vid=10&sid=32d64fd9-55f6-4524-a6c8-018faa25e41b%40sessionmgr103&bdata=Jmxhbmc9ZXMmc2l0ZT1lZHMtbG12ZQ%3d%3d#AN=edshlc.014355434.4&db=edshlc>

Llorens Largo, F. (2002). Programación: formalización, análisis y reutilización de algoritmos matemáticos. Obtenido de <https://eds-b-ebsohost-com.ezproxy.unapec.edu.do/eds/detail/detail?vid=13&sid=32d64fd9-55f6-4524-a6c8-018faa25e41b%40sessionmgr103&bdata=Jmxhbmc9ZXMmc2l0ZT1lZHMtbG12ZQ%3d%3d#AN=edshlc.014355463.8&db=edshlc>

X - Esquema Preliminar:

Agradecimiento	I
Dedicatorias	II
Resumen Ejecutivo	III
Introducción	IV
Aspecto Metodológicos	V

Capítulo I: ASPECTOS GENERALES DEL INSTITUTO TECNOLÓGICO FABIO AMABLE MOTA.

Contenido	Páginas
1.1 Introducción	000
1.2 Descripción del instituto	000
1.3 Reseña histórica	000
1.4 Filosofía	000
1.4.1 Misión	000
1.4.2 Visión	000
1.4.3 Valores	000
1.4 Objetivos del instituto	000
1.4.1 Objetivos generales	000
1.4.2 Objetivos específicos	000
1.5 Organigrama	000

Capítulo II: PROGRAMACIÓN DE SOFTWARE

Contenido	Páginas
2.1 Introducción	000
2.2 Conceptos de Programación	000
2.3 Definición de Software	000
2.4 Lenguajes de programación	000
2.5 Metodologías de programación	000
2.5.1 Programación Estructurada	000
2.5.2 Programación Modular	000
2.5.3 Programación Orientada a Objetos	000
2.5.4 Programación Concurrente	000
2.5.5 Programación Declarativa	000

2.5 Ventajas de aprender a programar	000
2.6 Importancia de la programación de software en la sociedad	000

Capítulo III: CONCEPTOS DE EDUCACIÓN

Contenido

3.1 Sistema escolar de la República Dominicana.	000
3.2 Tipos de aprendizaje	000
3.3 Métodos de enseñanza	000
3.4 Herramientas para enseñar a programar en escuelas	000
3.4.1 MakeCode	000
3.4.2 Swift Playgrounds	000
3.4.3 PSeint	000
3.4.4 Python	000

Capítulo IV: PROPUESTA DE IMPLEMENTACIÓN DE LA ENSEÑANZA DE PROGRAMACIÓN DE SOFTWARE COMO MATERIA BASE A NIVEL MEDIO EN EL INSTITUTO FABIO AMABLE MOTA

Contenido	Páginas
4.1 Introducción	000
4.2 Situación del pensum del INSTITUTO FABIO AMABLE MOTA	000
4.3 Propuesta de implementación de la enseñanza de la programación de software	000
4.3.1 Presupuesto de laboratorio	000
4.3.2 Fundamentos de la propuesta	000
4.3.3 Objetivos de la propuesta	000
4.3.4 Desarrollo de la propuesta	000
4.4 Beneficios del aprendizaje de la programación en educación media	000
4.4.1 Creatividad	000
4.4.2 Trabajo Colaborativo	000
4.4.3 Razonamiento Computacional	000
4.4.4 Constructivismo	000

Conclusión	000
Recomendaciones	000
Bibliografía	000
Glosario	000
Anexos	000