

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
VALPARAISO - CHILE



**“PROPUESTA DE UN PLAN DE ESTUDIOS
PARA LA FORMACIÓN TÉCNICO -
PROFESIONAL EN INFORMÁTICA
ARTICULABLE CON LA EDUCACIÓN
SUPERIOR”**

ANDRÉS ARMANDO DELGADO ALVARADO

**MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN INFORMÁTICA**

Profesor Guía: Carlos Castro
Profesor Correferente: Cristian Lara

Octubre - 2023

DEDICATORIA

Dedico el trabajo realizado en esta memoria a mi familia y amigos, los cuales fueron mis principales apoyos económicos y emocionales en los altos y bajos que se presentaron durante el transcurso de esta travesía por la etapa universitaria.

AGRADECIMIENTOS

Primero quiero expresar mis agradecimientos a mi madre y abuelas. Por su apoyo y confianza que me brindaron siempre para estudiar y lograr ser un profesional universitario.

A mi tía Ariela, que, a pesar de tener sus nietos, siempre estuvo presente como una abuela más, hasta cuando me acompañó a matricularme a la USM junto con mi madre. También, agradecer a mi Tío Jano que siempre me brindo su apoyo como si fuera su hijo.

También a mi hermano Diego, que a pesar de siempre decir “no te extrañare”, sentí que, si me extrañaba y era su ejemplo a seguir, a pesar de que antes peleábamos mucho.

A mis amigos que siempre estuvieron y siguen estando ahí para lo que los necesite, sufriendo, riendo, disfrutando, etc. juntos el proceso universitario y con algunos viviendo juntos.

Finalmente quiero agradecer a mis profesores que tuve en el proceso Universitario de cada uno obtuve un aprendizaje, ya sea de conocimiento o actitudes, y a todas las personas que estuvieron involucradas ayudándome a sacarla adelante.

RESUMEN

Resumen— En un mundo donde la tecnología está siendo muy importante y creciendo de manera exponencial, es primordial ver como en Chile se aborda la enseñanza de la especialidad de Programación en la EMTP y ver el ejemplo de países que nos llevan la delantera en el tema, en esta memoria se menciona a ciertos países como EE. UU., Austria, Alemania y Japón, abordan la programación en la secundaria.

Por otro lado, utilizando una matriz de doble entrada se analizan las asignaturas de la especialidad de programación en la EMTP en Chile y el T.U. en informática de la USM, así de dicha manera obtener que asignaturas podrían ser articulables en la educación superior.

En base a este análisis se logra evidenciar que existen falencias en la actual especialidad de programación en la EMTP, ya que los temas que abarcan son demasiado densos para los dos años de especialización, esto debería ser más específico dando enfoque a un área de la programación. Los profesores no están capacitados adecuadamente y/o no desean enseñar programación, además las habilidades que se necesitan en el mercado no están siendo satisfechas en los programas educativos de la especialidad de programación en la EMTP, por esto existe un déficit de personas con habilidades computacionales técnicas.

Palabras Clave— EMTP, EMHC, Programación, IVET, TI.

ABSTRACT

Abstract— In a world where technology is becoming very important and growing exponentially, it is essential to see how Chile approaches the teaching of the specialty of programming in the EMTP and to see the example of countries that are ahead of us on the subject, in this memory it is mentioned certain countries such as USA, Austria, Germany and Japan, approach programming in high school.

On the other hand, using a double-entry matrix, we analyze the subjects of the programming specialty in the EMTP in Chile and the T.U. in computer science of the USM, thus obtaining which subjects could be articulated in higher education.

Based on this analysis it becomes evident that there are shortcomings in the current programming specialty in the EMTP, since the topics covered are too dense for the two years of specialization, this should be more specific giving focus to an area of programming. The teachers are not adequately trained and/or do not want to teach programming, in addition, the skills needed in the market are not being met in the educational programs of the programming specialty in TVET, so there is a deficit of people with technical computer skills.

Keywords— EMTP, EMHC, Programación, IVET, TI.

GLOSARIO

EMTP: Enseñanza media técnica profesional

EMHC: Enseñanza media humanista científico

IVET: Initial Vocational Education and Training.

TI: Tecnología de la información

TP: Técnico profesional.

HC: Humanista científico.

USM: Universidad Santa María.

UTFSM: Universidad Técnica Federico Santa María.

T.U.: Técnico Universitario.

Framework(s): conjunto de reglas y convenciones que se usan para desarrollar software de manera más eficiente y rápida.

INDICE DE CONTENIDOS

RESUMEN.....	4
ABSTRACT	4
INDICE DE FIGURAS	9
INDICE DE TABLAS.....	10
INTRODUCCIÓN	11
CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA	12
1.1 Conceptos básicos.....	12
1.1.1 Breve panorámica histórica de la EMTP en Chile.	12
1.1.2 Lineamientos del MINEDUC en Chile.	13
1.2 Objetivos.....	14
1.2.1 Objetivo general.....	14
1.2.2 Objetivos específicos.	14
CAPÍTULO 2: MARCO CONCEPTUAL	15
2.1 Establecimientos y matrícula asociada.	15
2.2 Malla curricular EMTP en programación (informática).	17
2.3 Formación recibida y práctica profesional.	18
2.4 Efectividad de la formación de enseñanza media técnico profesional.....	19
2.5 Caracterización de la trayectoria laboral.	19
2.6 Implementación de informática en educación secundaria en otros países.	22
2.1.1 EE. UU.	22
2.1.2 Austria	23
2.1.3 Alemania	25
2.1.4 Japón	27
CAPÍTULO 3: PROPUESTA DE SOLUCIÓN.....	29
3.1 Plan de estudios de la especialidad de programación	29

3.2 Módulos de especialidad	29
3.2.1 Programación y bases de datos.....	29
3.2.2 Instalación y configuración de equipos informáticos.	30
3.2.3 Soporte a usuarios y productividad.	31
3.2.4 Sistemas operativos.	32
3.2.5 Diseño de bases de datos relacionales.	32
3.2.6 Programación orientada a objetos.	33
3.2.7 Administración de bases de datos.....	34
3.2.8 Desarrollo de aplicaciones web.....	35
3.2.9 Emprendimiento y empleabilidad.	35
3.3 Asignaturas del área TI en el Técnico universitario en informática USM	36
3.3.1 Análisis de Sistemas de Información	36
3.3.2 Programación	39
3.3.3 Introducción a la Informática y Computación	42
3.3.4 Análisis y Diseño Orientado a Objeto	46
3.3.5 Diseño de Sistemas de Información	49
3.3.6 Estructura de Datos.....	54
3.3.7 Programación Orientada a Evento	57
3.3.8 Taller de Sistemas de Información I	59
3.3.9 Programación Orientada a Objeto	62
3.3.10 Diseño y Programación Orientada a la Web	64
3.3.11 Bases de Datos.....	69
3.3.12 Arquitectura y Organización de Computadores.....	72
3.3.13 Taller de Sistemas de Información II	75
3.3.14 Desarrollo de Aplicaciones Móviles	76
3.3.15 Taller de Desarrollo de Software	80

3.3.16 Sistemas Operativos.....	81
3.4 MATRIZ DE CONVERGENCIA DE LA ESPECIALIDAD DE PROGRAMACION EMTP - T.U. INFORMATICA USM.	87
3.5 ASIGNATURAS CONVALIDABLES.....	92
CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN	93
4.1 CONVERSACIÓN CON PROFESOR DEL LICEO NUEVO MUNDO DE MULCHÉN SOBRE LA ESPECIALIDAD DE PROGRAMACIÓN EN LA EMTP.....	93
4.2 CONVERSACIÓN CON PROFESOR DEL LICEO COMERCIAL DE TALCAHUANO SOBRE LA ESPECIALIDAD DE PROGRAMACIÓN EN LA EMTP.....	93
4.3 PROPUESTA DE CAMBIO EN LA MALLA CURRICULAR DE LA EMTP.....	94
4.4 MALLA CURRICULAR PARA PROGRAMACION EN EMTP CON CAMBIOS DE ACTUALIZACIÓN.....	96
4.5 COMPARACIÓN PLAN DE ESTUDIOS ACTUAL Y LA PROPUESTA DE PLAN DE ESTUDIOS PARA PROGRAMACIÓN EN EMTP.	97
CAPÍTULO 5: CONCLUSIONES	102
TRABAJOS FUTUROS.....	102
REFERENCIAS BIBLIOGRÁFICAS.....	103
REFERENCIAS BIBLIOGRÁFICAS EN NORMA APA	103

INDICE DE FIGURAS

Ilustración 1 Cantidad de egresados, renta promedio y tasa de empleabilidad, total cohorte y por modalidad de enseñanza, cohortes 2008-2018.	20
Ilustración 2 Cantidad de egresados, renta promedio y tasa de empleabilidad, total cohorte y por modalidad de enseñanza. Cohortes 2008-2018, desagregado por sector económico.	21
Ilustración 3 Sueldo promedio y tasa de empleabilidad. Cohortes 2016-2018, desagregado por sector económico. El sector de Tecnología y Telecomunicaciones se creó el año 2016.	21
Ilustración 4 Carreras del distrito educacional de Spring Hill, Kansas, EE. UU. Especialidades de computación: Apps de computador, Programación para empresas, Programación de computadores, Programación de computadores - lugar de trabajo, y Comunicaciones para el siglo 21. (SET Academy!, n.d.).....	23
Ilustración 5 Escolarización en el sistema educativo austriaco. (Austria G. , n.d.)	25
Ilustración 6 Modelo de educación Dual Alemania.....	27
Ilustración 7 Plan de Estudio de la especialidad PROGRAMACIÓN (Chile, 2015).	29

INDICE DE TABLAS

Tabla 1 Número de establecimientos TP, según dependencia.	15
Tabla 2 Distribución de la matrícula 3ero y 4to medio según modalidad de estudios	16

INTRODUCCIÓN

La enseñanza de la programación a tomada gran fuerza a nivel mundial y debido a esto varios países como EE. UU., Japón, Argentina, Austria, Alemania y varios países europeos, generaron planes de estudios donde la programación es obligatorio según el currículum escolar o programas de formación (educación técnica) para la programación. En todos estos países que ya implementaron especialidades relacionadas a la programación en la Educación Media Técnico Profesional (EMTP).

En Chile desde el 2016 se incorporó en planes de estudio de la Educación Media Técnico Profesional (EMTP) para abordar áreas de la computación, las cuales son: Programación, y Conectividad y Redes. En este documento nos enfocaremos en la especialidad de Programación en la EMTP en Chile y analizar las posibilidades de articular asignaturas de la EMTP con el T.U. informática de la UTFSM, proponer una nueva malla curricular para la especialidad de programación para una mejor articulación y continuidad de estudios superiores.

CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA

1.1 Conceptos básicos.

En esta memoria se dará a entender el contexto nacional e internacional de la enseñanza media técnica profesional (EMTP) en informática y/o programación, su ambiente formativo y profesional.

1.1.1 Breve panorámica histórica de la EMTP en Chile.

Si bien la educación técnica es sumamente antigua, en Chile se tienen registros desde el siglo XVIII en el año 1798 con la Academia San Luis creada por don Manuel de Salas Corbalán. Un segundo registro es del siglo XIX cuando en 1842 se crea la Escuela de Artes y Oficios, un tercer registro en 1886 con la creación del Consejo de la Educación Técnica que luego paso a ser el Consejo de Enseñanza Agrícola e Industrial bajo la administración del Ministerio de Industrias y Obras Públicas. [(mineduc, 2015)]

El 26 de agosto 1942, se crea la Dirección General de Enseñanza Profesional, en el gobierno de Juan Antonio Ríos, reconociendo la importancia de la Educación Profesional para el desarrollo productivo del país.

En 1947, se funda la Universidad Técnica del Estado, en el gobierno de Gabriel González Videla, basándose en los centros de enseñanza de la Escuela de Artes y Oficios.

El 19 de junio de 1953, se fusionan la Dirección General de Enseñanza Profesional del Ministerio de Educación Pública y el Departamento de Enseñanza Agrícola del Ministerio de Agricultura, creando la Dirección General de Educación.

Luego en 1965 se comenzó a hacer popular la EMTP, producto de una reforma educacional del presidente Eduardo Frei Montalva, se reestructura el sistema donde se establecieron 8 años de educación primaria y 4 años de educación secundaria divididos en científico humanista y técnico profesional. La EMTP brindaba las herramientas para incorporarse al mundo del trabajo, pero también la posibilidad de continuar con la educación superior.

En 1980 y 1981, las instituciones educativas técnicas públicas fueron en su mayoría entregadas a los municipios. Los cuatro años de la EMTP se dividen en dos ciclos de dos años, el primero es de educación humanística y el segundo de tecnología profesional. En el contexto de la liberalización de la época, a cada escuela secundaria se le permitió crear sus propias carreras, programas y planes de estudio.

En la década de 1990, se eliminó el primer ciclo de dos años en un intento de retrasar las decisiones profesionales y garantizar el acceso a lo que se consideraban oportunidades de aprendizaje esenciales. Siguiendo el ejemplo de países como Alemania y Holanda.

En 1998 se llevaron a cabo nuevas reformas educativas, estandarizando carreras, reduciendo el número de carreras de 403 a 46 y eliminando carreras afines a la informática.

El programa de la especialización en programación (Informática EMTP) corresponde al sector económico de la tecnología y la comunicación, y está dirigido a estudiantes de la enseñanza media técnica profesional. Debido que el uso de las TIC está creciendo y extendiéndose constantemente, es que se necesita más población con formación técnica que sean capaces de desarrollar competencias laborales en el área de la tecnología. En Chile, el índice de ventas promedio mensual de servicios informáticos y otras actividades conexas aumentó un 53% en 2011 respecto al promedio de 2006, considerándose las TIC un importante factor de crecimiento. Las organizaciones a menudo integran herramientas de *hardware* y *software* que les permiten desarrollar soluciones que mejoran significativamente el rendimiento. [(Especialidad Programacion , 2016)]

En 2013, la Junta Nacional de Educación aprobó Mediante Decreto N°452, el establecer la formación técnico-profesional diferenciada para 34 especialidades y 17 de la educación media técnico profesional, las cuales son obligatorias para los establecimientos de EMTP (Especialidad Programación, 2016). Además, se realizó la publicación de un decreto de ley con bases curriculares, debido a esto se inició la nueva reforma, donde se reducían a 34 las especialidades dictadas y se introdujeron las nuevas especialidades de Programación, y Conectividad y Redes. Tras esto a fines del 2014, en el gobierno de Michelle Bachelet se pospuso la fecha de entrada en vigor del decreto de ley, originalmente programada para el 2015 a tener una nueva fecha en el 2016, sin realizar modificaciones a las bases curriculares, las cuales aún se mantienen hasta la fecha. [(EDUCACIÓN, 2015)]

Debido a todo lo mencionado se realizará una investigación y análisis relacionado a la programación e/o informática en la educación secundaria en otros países del mundo como Austria, Japón, Alemania, EEUU (estado de Kansas) y Chile, para generar una malla curricular articulable con la educación superior en Chile, así obtener una posible convalidación de asignaturas en la educación superior y el título de informático EMTP tenga una mayor validez al momento de buscar oportunidades laborales para dichos alumnos que no puedan continuar con sus estudios en la educación superior.

1.1.2 Lineamientos del MINEDUC en Chile.

El decreto de ley del año 2013 que reforma la EMTP establece diez objetivos de aprendizaje para la especialidad de programación [(Lineamiento escolar 2022, 2021)]. Estos utilizan un enfoque funcional de la computación (informática), evitando en gran medida sus fundamentos teóricos, que se refieren principalmente a la instalación y el mantenimiento del software, así como al soporte al usuario. Los planes de investigación que se publicaron más tarde siguieron la misma línea, reduciendo de los objetivos de aprendizaje esperados igualmente. [(Lineamientos de evaluacion y promocion, 2021)]

1.2 Objetivos.

1.2.1 Objetivo general.

Analizar el plan de estudio de informática en enseñanza media técnico profesional (EMTP), para buscar la manera de articular la continuidad de estudios informáticos en educación superior.

1.2.2 Objetivos específicos.

- Proponer una modificación o generar una nueva malla para informática en EMTP.
- Desarrollar un esquema general de plan y programa para formación de profesores de EMTP en el área.
- Analizar un marco teórico que argumente la programación en la EMTP como disciplina científica y no sólo como herramienta técnica.

CAPÍTULO 2: MARCO CONCEPTUAL

2.1 Establecimientos y matrícula asociada.

En Chile la Enseñanza Media Técnica Profesional (EMTP) es un modelo educativo encargado de brindar a los jóvenes una formación integral en el ámbito de una especialidad que facilite su inserción laboral, con el fin de la continuación de sus estudios sistemáticos. Este tipo de formación, a diferencia de la Enseñanza Media Científico-Humanista (EMCH), tiene una clara orientación académica o de formación general una orientación profesional más centrada en al campo ocupacional específico. [(Educacion tecnica profesional en chile, 2012)]

Según datos oficiales del Ministerio de Educación en 2010, la EMTP se impartía en 946 instituciones educacionales de enseñanza media, de las cuales solo el 37% de las escuelas también imparten la EMCH en paralelo.

Tabla 1 Número de establecimientos TP, según dependencia.

Dependencia	Modalidades impartidas		Total	Porcentajes
	Solo TP	TP y CH		
Municipal	214	216	430	45.5%
Particular Subvencionado	315	130	445	47.0%
Particular Pagado	2	0	2	0.2%
Administración delegada	69	0	69	7.3%
Total	600	346	946	100.0%
Porcentaje	63.4%	36.6%	100.0%	

Fuente: Base Matrícula de Establecimientos 2010

Además, del total de establecimientos que imparten la modalidad de EMTP, el 45.5% corresponde a centros educacionales municipal y el 47.0% a establecimientos particular subvencionados, el 7.35% lo cual se traduce a 69 centros educacionales pertenecen a Corporaciones de Administración delegada y tan solo el 0.2% es del sector particular

pagado, lo cual nos hace notar que es prácticamente nula su participación en la EMTP. [(Educacion tecnica profesional en chile, 2012)]

Alrededor de 204.000 estudiantes se matriculan en establecimientos de EMTP a nivel nacional. Lo cual representa un 44.7% del total de la matricula en los dos últimos años de educación media versus un 55.3% de matrículas en establecimientos de EMCH. En la última década estas proporciones se han mantenido relativamente estables. Además, es importante tener en cuenta que la EMTP es mayoritariamente relevante en la III y IX región, ya que, un 54% de las matrículas de 3ero y 4to medio se adscribe a esta modalidad educativa, yendo al caso contrario en la IV, VI y XI región su participación es inferior al 38%.

Tabla 2 Distribución de la matricula 3ero y 4to medio según modalidad de estudios

Región	EMTP	EMCH
I	52.1%	47.9%
II	47.3%	52.7%
III	57.2%	42.8%
IV	37.8%	62.2%
V	38.1%	61.9%
VI	37.1%	62.9%
VII	52.3%	47.7%
VIII	42.3%	57.7%
IX	54.3%	45.7%
X	50.7%	49.3%
XI	36.9%	63.1%
XII	43.2%	56.8%
XIII	44.1%	55.9%

XIV	50.4%	49.6%
XV	44.9%	55.1%
Total	44.7%	55.3%

Fuente: Datos de matrícula 2010, MINEDUC

Debido a estos antecedentes se observa que la EMTP como modalidad concreta una importante proporción de las matrículas de enseñanza media, también se nota que principalmente es proveniente de hogares con menor nivel de educación e ingresos. [(Educación técnica profesional en Chile, 2012)]

2.2 Malla curricular EMTP en programación (informática).

La especialidad de Programación en la EMTP se estableció en el decreto de ley de Chile que modifica la EMTP, que fue promulgado y publicado en 2013 [(ESTABLECE BASES CURRICULARES PARA LA EDUCACIÓN MEDIA FORMACIÓN DIFERENCIADA TÉCNICO-PROFESIONAL, 2013)]. Esto demuestra contexto laboral de la especialidad de computación. El año 2015 se publicó el programa de estudios para la especialidad de programación de la EMTP. [(Especialidad Programación, 2016)]

La especialidad de Programación en la EMTP se comenzó a dictar en el 2016, por lo que se aprovecha la experiencia de los colegios que imparten la especialidad para el diseño y desarrollo de programas y proyectos. Estos cursos rara vez tienen en cuenta los desarrollos de la industria, por lo que muchas de las herramientas y habilidades que se enseñan son de poca utilidad práctica o incluso están desactualizadas. Algunos de los productos esperados de la especialidad, como soporte al usuario, mantenimiento y configuración de equipos, no coinciden con el perfil de programación. Muchos de estos servicios se subcontratan y las empresas que los brindan son las más indicadas para brindar la capacitación necesaria.

Finalmente estarán listados los objetivos de aprendizaje para la especialidad de computación [(Especialidad Programación, 2016)]:

1. Desarrollar programas y rutinas de baja y mediana complejidad que involucren estructuras y bases de datos de acuerdo con los requerimientos de la empresa.
2. Instalar sistemas de información manufacturados, para obtener la información necesaria de acuerdo con los requerimientos de la empresa.

3. Desarrollar aplicaciones y rutinas para el mantenimiento y la actualización de bases de datos de acuerdo con los requerimientos de la empresa.
4. Construir aplicaciones computacionales basadas en programación orientada a objetos, de manera de cumplir con las exigencias técnicas y de los usuarios.
5. Desarrollar aplicaciones web acordes a los requerimientos y utilizando los lenguajes de programación disponibles en el mercado, permitiendo la administración y publicación de información en internet.
6. Implementar planes de mantención preventiva y correctiva de software, de manera de fomentar su productividad.
7. Dar soporte técnico a usuarios de un sistema computacional en forma local o remota, cumpliendo con las exigencias técnicas y de los usuarios.
8. Instalar y actualizar elementos de hardware y software en una organización, con el fin de potenciar el rendimiento y/o corregir errores típicos.
9. Instalar y configurar sistemas operativos en computadores personales con el fin de incorporarlos a una red LAN, cumpliendo con los estándares de calidad y seguridad establecidos.
10. Instalar software de productividad y programas utilitarios en un equipo personal, de acuerdo con los requerimientos de los usuarios.

2.3 Formación recibida y práctica profesional.

La mayoría de los estudiantes de la EMTP, indican que la base que reciben en su formación es mínima, alcanza justo para cumplir con los requerimientos laborales en las respectivas especialidades que eligieron, esto queda a la vista en sus primeras experiencias de trabajo, por ejemplo, en su práctica profesional. Lamentablemente no existe un currículum, de la EMTP y los conocimientos adquiridos quedan al debe, por lo cual se nota que el currículum de la EMTP queda desactualizado en temas necesidades actuales de las empresas.

Adicionalmente, las personas egresadas, en su gran mayoría, también revelan que hay una ausencia de formación laboral focalizada. Una ex estudiante de un EMTP indica que recibió asignaturas que la prepararon exclusivamente para el desarrollo de habilidades blandas y herramientas laborales, como la presentación formal ante una entrevista de trabajo o la redacción de un *currículum vitae* (CV). [(Pavez, 2022)]

Finalmente, con esto evidenciamos que a la EMTP le hace falta focalizar más su malla curricular para preparar de mejor manera a los estudiantes, es por esto que desarrollaré una propuesta de mejora, y así el diploma de la especialidad sea de valor al momento de buscar un trabajo y/o tener la posibilidad de articular alguna asignatura en el caso de continuar los estudios en la educación superior.

2.4 Efectividad de la formación de enseñanza media técnico profesional.

La enseñanza media técnico profesional (EMTP) tiene como misión formar a sus alumnos para que una vez egresados se incorporen al mundo del trabajo en condiciones tales que cumplan una función económicamente productiva y, a la vez, satisfagan sus necesidades. Por lo que se evalúa constantemente las situaciones de los alumnos egresados de la EMTP, estudios técnicos e investigaciones empíricas realizadas en Chile dan cuenta que la situación no es adecuada, ya que, la calidad de la formación profesional otorgada en los establecimientos escolares de enseñanza media es insatisfactoria, quiere decir que prácticamente dicho diploma no es tomado en cuenta por las empresas. [(Poblete, 2017)]

Finalmente debido a que no existen mecanismos que permitan evaluar los resultados de aprendizaje, dificulta la retroalimentación para mejorar la malla curricular de las carreras en la EMTP.

2.5 Caracterización de la trayectoria laboral.

Se realiza el análisis entre las modalidades de enseñanza humanista-científico (HC) y técnico-profesional (TP), además de los sectores y especialidades de la modalidad de EMTP. Considerando que los alumnos egresados de la EMTP pertenecen principalmente de nivel socioeconómico bajo y medio-bajo, por otro lado, los alumnos egresados de la educación HC pertenecen al nivel socioeconómico medio, medio-alto y alto, es importante notar que debido a esto existe una gran influencia en la trayectoria laboral de los alumnos egresados de bajos recursos.

Para la caracterización de las trayectorias laborales se han considerado los datos de salario imponible otorgados por la Superintendencia de Pensiones. Estos datos contemplan a las personas afiliadas al sistema de pensiones y que sean cotizantes del mismo. [(Mineduc, 2020)]

Los datos que se presentan a continuación toman en cuenta los salarios hasta los egresados en el año 2018. Además, se presentan los datos de la tasa de empleabilidad, considerando que un trabajador está empleado si tiene al menos seis meses liquidación de sueldo en el último año.

En la Ilustración 1 se observan los datos comparativos de la desagregación entre la EMTP y HC, para egresados del año 2008 al 2018, cantidad de egresados por generación y modalidad, además, sueldo promedio del último año y empleabilidad.

Año egreso	Total cohorte			Total TP			Total HC		
	N	\$	%	N	\$	%	N	\$	%
2008	207.378	827.935	61,9	91.529	716.463	64,6	115.849	923.020	59,8
2009	203.684	767.308	59,7	90.233	672.153	63,7	113.451	852.640	56,5
2010	203.137	705.709	55,8	89.398	630.966	61,6	113.739	776.374	51,2
2011	195.959	636.667	50,5	86.078	590.122	59,8	109.881	687.136	43,2
2012	193.242	560.706	43,7	83.581	546.601	56,5	109.661	578.542	34,0
2013	190.998	495.496	36,3	79.208	503.868	52,5	111.790	482.989	24,9
2014	188.958	444.278	29,5	76.318	465.261	46,3	112.640	408.069	18,2
2015	190.884	407.978	24,3	75.765	431.291	39,9	115.119	364.360	14,0
2016	188.054	380.610	19,2	73.094	404.087	33,4	114.960	331.916	10,2
2017	189.466	356.780	15,4	71.897	379.806	27,1	117.569	310.269	8,2
2018	195.496	336.310	4,9	72.370	362.321	7,8	123.126	299.129	3,2

Fuente: Unidad de Asesoría Estratégica, Centro de Estudios MINEDUC, División de Planificación y Presupuesto, MINEDUC.

Ilustración 1 Cantidad de egresados, renta promedio y tasa de empleabilidad, total cohorte y por modalidad de enseñanza, cohortes 2008-2018.

Podemos apreciar en la ilustración 1, que los sueldos promedios para los egresados de la educación HC es mayor que en la TC, pero solo el año 2012, ya que, en ese año comienza a revertirse la tendencia y la educación técnica-profesional tiene un sueldo promedio superior. Por otro lado, llama la atención las tasas de empleabilidad (Para definir la tasa de empleabilidad, primero se calculó la densidad laboral promedio para todas las cohortes con las rentas desde septiembre 2018 a agosto 2019) y salarios promedios tienen una tendencia en baja año a año, se cree que esto es debido a factores como la experiencia laboral de los egresados y en la nula continuación con una educación superior.

En la Ilustración 2 podemos visualizar a los egresados con la especialidad de programación de los años (2016, 2017, 2018) en los cuales tenemos información en la EMTP, de las tres variables analizadas podemos notar que el número de egresados aumento de manera exponencial, mientras que en el salario promedio que obtienen comenzó a disminuir, al igual que el porcentaje de empleabilidad, probablemente esto se deba a que varios de esos

alumnos egresados no continuaron especializándose o no continuaron con estudios superiores para afianzar sus conocimientos. [(Mineduc, 2020)]

Año egreso	Total cohorte																	
	Minero			Gráfica			Químico			Confección			Tecnología y Telecomunicaciones			Alimentación		
	N	\$	%	N	\$	%	N	\$	%	N	\$	%	N	\$	%	N	\$	%
2008	622	1.022.912	69,1	1.028	668.237	60,9	1.083	762.272	66,3	918	545.440	49,5	-	-	-	7.674	576.675	57,9
2009	769	945.917	65,0	1.225	645.686	60,5	1.307	722.849	63,7	1.013	504.946	46,3	-	-	-	7.920	543.159	57,2
2010	820	875.252	64,5	1.238	632.986	58,2	1.265	674.510	59,4	831	481.382	49,2	-	-	-	8.178	\$517.399	55,4
2011	806	798.414	59,1	1.081	591.986	53,9	1.307	617.195	56,8	664	464.515	46,8	-	-	-	7.902	493.446	54,5
2012	1.124	670.818	55,2	1.098	513.507	51,8	1.254	563.597	51,1	592	441.558	47,6	-	-	-	7.497	471.416	53,6
2013	1.225	590.996	51,4	1.055	495.353	49,0	1.160	533.343	47,8	443	392.561	42,7	-	-	-	7.145	438.893	49,4
2014	1.693	530.988	43,1	1.041	458.103	43,6	1.172	482.221	41,4	424	388.517	36,8	-	-	-	6.728	417.480	43,9
2015	2.110	480.135	35,2	1.061	425.528	35,2	1.169	441.279	37,0	370	357.492	27,8	-	-	-	6.836	398.901	39,8
2016	2.242	452.279	30,0	1.033	401.854	28,4	1.086	442.279	31,2	315	332.591	24,8	2.609	421.330	33,4	6.601	374.902	32,5
2017	2.028	442.163	23,1	1.067	369.004	23,8	1.112	422.237	24,4	294	283.900	19,4	3.113	384.823	27,9	6.734	353.948	27,3
2018	1.847	391.651	5,5	1.147	349.222	6,1	1.094	391.165	7,3	279	268.804	3,6	3.795	378.385	8,0	6.716	337.603	8,3

Ilustración 2 Cantidad de egresados, renta promedio y tasa de empleabilidad, total cohorte y por modalidad de enseñanza. Cohortes 2008-2018, desagregado por sector económico.

Año egreso	Sector Tecnología y Telecomunicaciones								
	Conectividad y Redes			Programación			Telecomunicaciones		
	N	\$	%	N	\$	%	N	\$	%
2008	-	-	-	-	-	-	-	-	-
2009	-	-	-	-	-	-	-	-	-
2010	-	-	-	-	-	-	-	-	-
2011	-	-	-	-	-	-	-	-	-
2012	-	-	-	-	-	-	-	-	-
2013	-	-	-	-	-	-	-	-	-
2014	-	-	-	-	-	-	-	-	-
2015	-	-	-	-	-	-	-	-	-
2016	31	349.217	32,3	52	374.055	28,8	2.526	423.019	33,5
2017	487	361.384	27,3	377	321.692	21,8	2.249	397.525	29,0
2018	617	349.051	6,5	553	322.624	4,5	2.625	389.172	9,1

Ilustración 3 Sueldo promedio y tasa de empleabilidad. Cohortes 2016-2018, desagregado por sector económico. El sector de Tecnología y Telecomunicaciones se creó el año 2016.

El sector de Tecnología y Telecomunicaciones se creó el año 2016, por eso no presenta datos anteriores a dicho año, sin embargo, podemos notar la misma línea de tendencia que en la EMTP o HC de años anteriores.

2.6 Implementación de informática en educación secundaria en otros países.

2.1.1 EE. UU.

El sistema educacional de EE. UU. es descentralizado, esto quiere decir que cada estado y gobierno local son quienes toman directamente las decisiones educacionales. [(Organization of U.S. Education, 2008)]

Debido a esto la calidad, grado y su implementación depende netamente según el estado y distrito donde este ubicado.

A diferencia de Chile, en la mayoría de los estados y/o distritos de EE. UU. se puede encontrar más de una especialidad relacionada a la computación, debido a esto las especialidades varían demasiado entre los estados y distritos.

Ya que todos los estados en EE. UU. se rigen por sus propios programas de educación. Se toma de referencia la malla curricular de informática en la secundaria del estado de Kansas EE. UU., sin embargo, igual varían las asignaturas dictadas entre una escuela y otra, pero hay temas que son comunes como:

Los fundamentos de la tecnología, donde aprenden a manipular componentes básicos de la tecnología, lo más importante que aprenden es sobre redes y seguridad informática. Además, aprenden lenguajes de programación como Python, Java y C++, por otro lado, los lenguajes de marcado que aprenden son HTML, CSS y JavaScript poniéndolo a prueba con la creación de desarrollo web, donde lo aplicarán para diseño de interfaces usuarias y aprenderán a trabajar con bases de datos. En análisis de datos aprenderán a visualizar datos, mediante técnicas para analizar grandes conjuntos de datos y elaborar reportes para presentar los resultados del análisis.

Sin embargo, cada escuela puede tener un enfoque diferente y puede incluir otros temas y actividades en su plan de estudios.

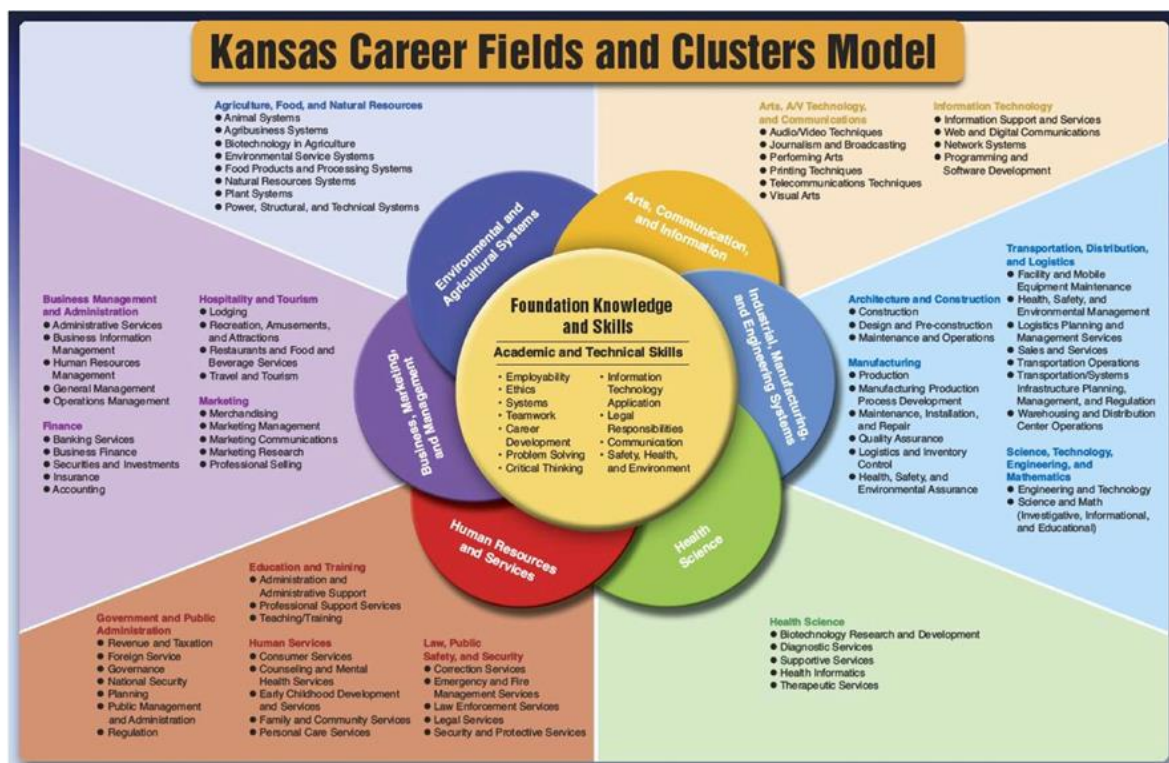


Ilustración 4 Carreras del distrito educacional de Spring Hill, Kansas, EE. UU.

Especialidades de computación: Apps de computador, Programación para empresas, Programación de computadores, Programación de computadores - lugar de trabajo, y Comunicaciones para el siglo 21. (SET Academy!, n.d.)

Por otro lado, los programas de CTE de las escuelas públicas de la ciudad de Chicago, EE.UU. Presentan especialidades de: Redes Cisco, Ingeniería en computación, Programación de computadores, Programación de juegos, Aplicaciones de TI, Programación de bases de datos, y Aplicaciones de software [(SCHOOLS, 2021)]. Entre los lenguajes de programación que más se repiten al momento de enseñan en las escuelas esta: Python, Java, C++, HTML, CSS, SQL, JavaScript. [(Aeseducation, 2019)]

2.1.2 Austria

En Austria implementaron la programación en el currículum nacional en la educación general durante toda la secundaria, por otro lado, para las especialidades de *Initial Vocational Education and Training* (IVET) solo es durante la educación secundaria superior.

En 2011, el Ministerio de Educación publicó nuevos estándares educacionales para la IVET. Tras ese suceso la educación técnica paso a durar cinco años, desde el decimotercer año de

estudio, además de certificar las cualificaciones profesionales, facilita el acceso a la Universidad. Esto se ve reflejado, ya que, la mitad de los estudiantes que obtienen el diploma optan por continuar con sus estudios en la Universidad, mientras que los otros son bien recibidos en las empresas. [(Schoolnet, Bruselas, 2015)]

Además, la educación secundaria se divide en secundaria inferior (*Hauptschule*) y en secundaria superior (*Gymnasium*).

En la *Hauptschule*, el principal enfoque de la informática es introducir a los estudiantes a conceptos de informática básica y alfabetización digital.

En el *Gymnasium*, se profundiza más en la informática añadiéndola en asignaturas como matemáticas, física y tecnología. También los estudiantes tienen la oportunidad de profundizar sus habilidades en la programación, la robótica, la inteligencia artificial, el análisis de datos y entre otros temas relacionados al área. La malla curricular depende de la escuela, pero deben cumplir con los objetivos de aprendizaje impuestos por el Ministerio de educación. [(Schoolnet, European Schoolnet, 2017)]

- Introducir a los estudiantes a la programación y la lógica de la computación.
- Fomentar el pensamiento crítico y la resolución de problemas.
- Desarrollar habilidades de colaboración y comunicación en el contexto de la tecnología.
- Enseñar a los estudiantes sobre la seguridad en línea y la privacidad de los datos.

Algunos de los lenguajes de programación que se enseñan en la secundaria son:

- Scratch: se enseña para los conceptos básicos de programación.
- Python: se enseña para la ciencia de datos, la inteligencia artificial y la informática en general.
- HTML, CSS y JavaScript: se enseñan para los fundamentos de la programación web.

De manera general, la computación en la IVET de Austria tiene buenos resultados, y se observa investigación en mallas curriculares y metodologías de enseñanza, por lo que Austria es también un buen candidato para caso de estudio.

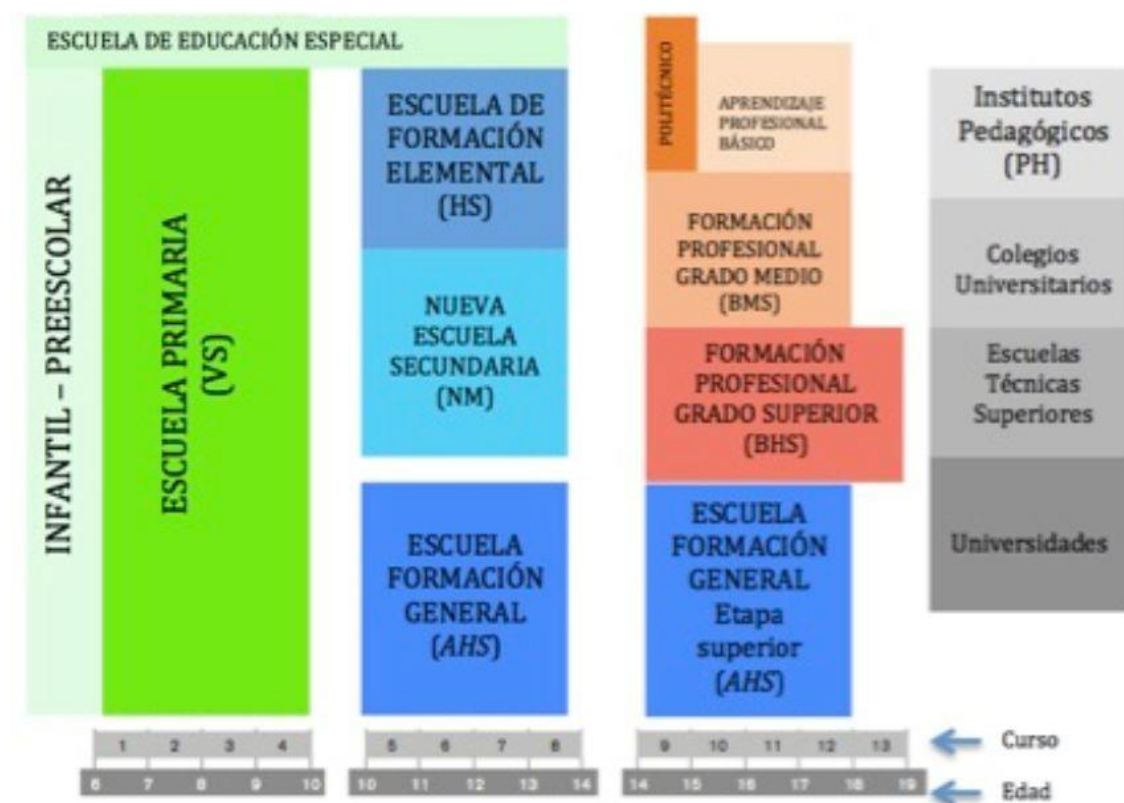


Ilustración 5 Escolarización en el sistema educativo austriaco. (Austria G. , n.d.)

2.1.3 Alemania

En Alemania, la computación se enseña en la educación general, no tan solo en los colegios técnicos, claro que, con orígenes y enfoques distintos, pero ambos orientados a actividades y contextualización. La educación técnica usa un modelo global, pero sin una descripción de niveles de competencia. En la educación general, las ciencias de la computación se enseñan entre el quinto y el décimo año de estudio, bajo el modelo educativo basado en el contexto, utilizado para enseñar las ciencias básicas tal como la física, química y biología. En 2004 se implanto la asignatura obligatoria de computación. [(Hubwieser, Noviembre, 2012)]

Sabemos que los estándares educacionales en Alemania se dividen en los contenidos, donde vemos la información y datos, algoritmos, lenguajes de autómatas, informática y sistemas informáticos, finalmente el hombre y la sociedad. Por otro lado, los procesos, donde vemos modelar e implementar, razonar y evaluar, estructurar e interrelacionar, comunicar y cooperar, finalmente representar e interpretar. [(Magenheim)]

La computación en educación media está muy desarrollada en Alemania, especialmente en *Initial Vocational Education and Training (IVET)*. Preliminarmente puede ser un muy buen referente en cuanto a mallas curriculares y metodologías de enseñanza.

En Alemania existen tres tipos de educación secundaria las cuales son:

1. Secundaria general, sin orientación técnica. Permite ir a la universidad después.
2. Secundaria que combina la formación general con la técnica. Permite ir a la universidad después.
3. Secundaria de exclusiva formación técnico profesional. No permite ir a la universidad después.

Esta última (3) es la conocida Educación dual con una duración de dos o tres años dependiendo el oficio, lo cual es equivalente a la EMTP. Esta como el nombre lo dice, esta dividida en dos grandes áreas una la educación teórica, en la cual debe asistir entre uno o dos días a la escuela, en la cual los contenidos de los cursos teóricos son orientados al oficio, además de enseñar alemán, matemáticas y ciencias sociales. Entre los lenguajes de programación más enseñados están Java, Python, C++, PHP, JavaScript y SQL, esto varía dependiendo del plan curricular que tenga la escuela. Esto es complementado con la educación práctica, esto se realiza directamente en las empresas y es entre tres a cuatro días por semana, donde el aprendiz (alumno) se integra como un trabajador de la empresa, donde un empleado certificado como maestro formador, lo orienta en cada una de las partes del oficio y comience a desenvolverse de manera óptima en el área laboral. Los conocimientos teóricos adquiridos en la escuela le ayudaran a complementar y resolver las dudas que surjan en su día a día al momento de colocar sus conocimientos en práctica.

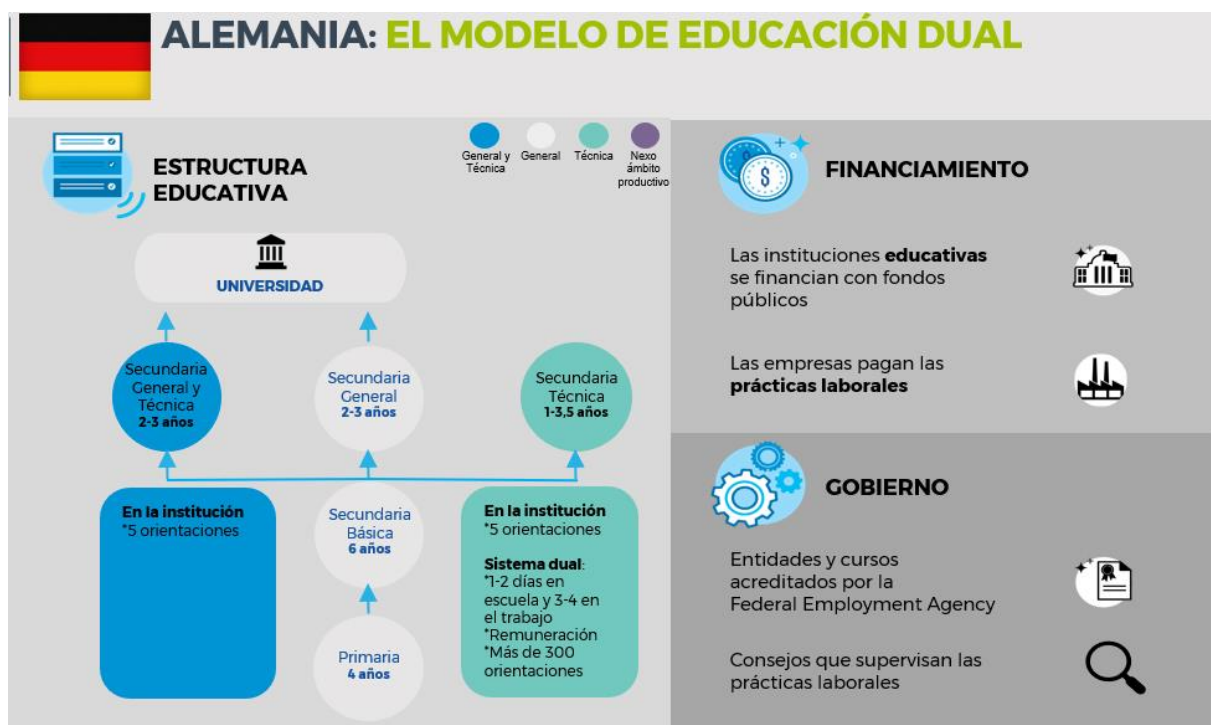


Ilustración 6 Modelo de educación Dual Alemania.

Las principales ventajas que tiene este método es que la educación técnica este articulada con el mercado laboral, respondiendo las necesidades económicas y suplir las vacantes laborales con los jóvenes graduados, así evitando que pasen años sin encontrar trabajo. Este sistema dual permite que en momentos de crisis no sean potencialmente perjudicados los recién graduados, y así disminuir el desempleo juvenil. Además, en Alemania la educación universitaria no garantiza un mayor salario: “El sueldo promedio de un técnico es de 42.000 euros al año, mientras que el de un periodista con dos años de experiencia es de 34.000 euros”. [(digital, 2016)]

2.1.4 Japón

En Japón a partir desde el año 2020 en la educación general del país, plantearon comenzar con un cambio paulatino para incrementar el conocimiento, donde enfocaran la computación y programación para la creación de software como asignaturas obligatorias. Además, en las ciudades de Takeo, Koha y la prefectura de Ibaraki un grupo de escuelas primarias y secundarias desde el 2020 imparten clases los fines de semana con la intención de que los alumnos creen programas y aprendan a trabajar con robots, impresoras 3D, entre otras tecnologías. [(PARLAMENTARIO, 2018)]

“Los medios japoneses comentaron que la conexión entre TI y la sociedad se está acercando cada vez más, y que la competencia internacional en industrias relacionadas se intensifica

constantemente. Sin embargo, Japón está rezagado en la educación de *TI*, con la esperanza de cultivar la escasez de talentos de *TI* al revisar los libros de texto.” [(Una nueva era de "educación en programación" Japón enumera la programación como una escuela primaria obligatoria, 2018)]

Por lo que en secundaria la programación ya es parte del plan curricular, además, que ya se enseña robótica básica en tecnología. Mientras que en la asignatura de programación realizaran aproximaciones a lenguajes básicos de programación, inducción con Arduino, ciencias de programación, entre otros temas. Los japoneses ven esto como un gran paso a que los jóvenes desarrollen el pensamiento lógico mediante experiencias de programación, los estudiantes aprenderán dos tipos de programación: medición-control y comunicación en redes en varias unidades. Dentro de los lenguajes de programación que aprenden los estudiantes incluye Java, C++, Python, Ruby y Scratch. [(Parlamentario, 2018)]

Finalmente, uno de los problemas más grandes que surge con este cambio educacional, es que, se necesita una cantidad considerable de profesores capacitados en programación (400 mil aprox.), el gobierno nipón establece la necesidad de incluir a la sociedad a la formación de competencias informáticas (codificación). El plan de acción para lograr este cometido es que las sociedades académicas y comunidades profesionales se involucren con esta transición educacional, de dicha manera ayudar en términos de contenidos e infraestructuras a los profesores.

CAPÍTULO 3: PROPUESTA DE SOLUCIÓN

3.1 PLAN DE ESTUDIOS DE LA ESPECIALIDAD DE PROGRAMACIÓN

NOMBRE DEL MÓDULO	TERCERO MEDIO	CUARTO MEDIO
	Duración (horas)	Duración (horas)
1. Programación y bases de datos	228	
2. Instalación y configuración de equipos informáticos	190	
3. Soporte a usuarios y productividad	228	
4. Sistemas operativos	190	
5. Diseño de bases de datos relacionales		152
6. Programación orientada a objetos		228
7. Administración de bases de datos		152
8. Desarrollo de aplicaciones web		228
9. Emprendimiento y empleabilidad		76
Total	836	836

Ilustración 7 Plan de Estudio de la especialidad PROGRAMACIÓN (Chile, 2015).

3.2 MÓDULOS DE ESPECIALIDAD

3.2.1 Programación y bases de datos.

Promueve el pensamiento algorítmico y sugiere establecer relaciones con el sector productivo para generar soluciones de software aplicadas en contextos laborales.

El objetivo de aprendizaje de la especialidad es poder desarrollar programas y rutinas de baja y mediana complejidad que involucren estructuras y bases de datos de acuerdo con los requerimientos de la empresa, donde los estudiantes representen situaciones y problemas mediante la programación de algoritmos y reconozcan patrones lógicos secuenciales,

utilizando NetBeans (principalmente Java) y MySQL. Por medio de la metodología de enseñanza de resolución de casos, centrada en el desarrollo de los Objetivos de Aprendizaje Genéricos y actividades que incorporan metodologías activas para potenciar el trabajo en equipo. (mineduc, 2015)

Los aprendizajes esperados en este módulo son:

1. Construye algoritmos y diagramas de flujo, en forma estructurada, empleando sentencias necesarias para solucionar requerimientos simples de programación, según estándares de la industria.
2. Desarrolla programas simples empleando la programación estructurada y/o modular, para resolver sistemas de información a requerimiento de la empresa, según estándares de programación.
3. Desarrolla programas de mediana complejidad, utilizando la orientación a objetos que involucre estructuras y bases de datos, aplicando “buenas prácticas” definidas por la industria.

3.2.2 Instalación y configuración de equipos informáticos.

Busca que los estudiantes utilicen los componentes de hardware en sistemas de tecnologías de la información, identifiquen cada componente y sus relaciones, desarrollen un criterio adecuado para generar soluciones de soporte. El objetivo principal es que aprendan a instalar y actualizar elementos de hardware y software en una organización, con el fin de potenciar el rendimiento y/o corregir errores típicos.

Los estudiantes también aprenderán a diferenciar entre software y hardware. Además, se espera que aprendan a ensamblar y diferenciar correctamente una computadora, identificar los dispositivos de entrada, salida y almacenamiento, y usar las herramientas de hardware y software.

El módulo consta de sesiones prácticas y teóricas, con presentación de casos, análisis de manuales técnicos y ejercicios en el laboratorio de computación. (mineduc, 2015)

Los aprendizajes esperados en este módulo son:

1. Instala componentes internos de hardware de una estación monousuario, según requerimientos de organización, manuales técnicos y normas de seguridad.
2. Realiza mantención de una estación monousuario para potenciar su rendimiento, de acuerdo con los requerimientos de usabilidad, manuales técnicos y normativa vigente.
3. Potencia el rendimiento de un modelo de red, de acuerdo con los requerimientos de los usuarios y en concordancia con estándares informáticos.

3.2.3 Soporte a usuarios y productividad.

En este módulo, se enfatiza en el aprendizaje práctico de las herramientas informáticas. También es importante que los y las estudiantes adquieran conocimientos sobre los diferentes ambientes operativos y las aplicaciones que pueden ser utilizadas en cada uno, y que desarrollen habilidades para identificar y solucionar problemas que puedan surgir durante la operación de las aplicaciones.

Es importante que las metodologías activas, centradas en los y las estudiantes, sean implementadas en las actividades de aprendizaje, de forma que los y las estudiantes puedan desarrollar habilidades prácticas y aplicables en su futuro desempeño laboral. Además, se recomienda que se fomente la investigación y actualización constante de la normativa vigente y los protocolos asociados, con el objetivo de brindar un soporte informático de calidad y acorde a los estándares del mercado. (mineduc, 2015)

Los objetivos principales esta aprender a:

- Instalar software de productividad y programas utilitarios en un equipo personal, de acuerdo con los requerimientos de los usuarios.
- Implementar planes de mantención preventiva y correctiva de software, de manera de fomentar su productividad.
- Dar soporte técnico a usuarios de un sistema computacional en forma local o remota, cumpliendo con las exigencias técnicas y de los usuarios.

Los aprendizajes esperados en este módulo son:

1. Instalar aplicaciones para optimizar tareas en el puesto de trabajo considerando, propósito específico, utilidades, especificaciones de trabajo y requerimientos del usuario.
2. Realiza mantenimiento preventivo de software a un equipo personal de computación, considerando procedimientos técnicos específicos y manuales de mantención.
3. Otorga asistencia técnica para corregir y resolver problemas de operación de sistemas (software) de acuerdo con los manuales y especificaciones técnicas.
4. Otorga soporte técnico local para resolver problemas de operación del equipo (hardware), que afectan la productividad en el puesto de trabajo, considerando normas ergonómicas y medioambientales.

5. Realiza instalación y configuración de aplicaciones de soporte centralizado, para asistencia a usuarios que requieren oportunidad en la atención, de acuerdo con el modelo mesa de ayuda (*HelpDesk*).

3.2.4 Sistemas operativos.

Este módulo se enfoca en el uso y manejo de sistemas operativos, tanto en computadores como en otros dispositivos tecnológicos que cuenten con procesadores. El objetivo principal es que los y las estudiantes desarrollen competencias para administrar y controlar el hardware.

Entre los temas que se abordan se encuentra el uso de programas propios del sistema operativo, la programación de tareas, la administración de recursos, la identificación de sistemas operativos adecuados según el tipo de usuario, la realización de mantenimientos y la solución de problemas frecuentes en el sistema. Por medio de sesiones prácticas y teóricas, se presentan casos y se entregan conocimientos para que los y las estudiantes realicen análisis que implementarán en los laboratorios. Al finalizar este módulo, los y las estudiantes deberían estar en condiciones de instalar y configurar sistemas operativos de estación cliente y servidores de red.

El objetivo principal es que aprendan a instalar y configurar sistemas operativos en computadores personales con el fin de incorporarlos a una red LAN, cumpliendo con los estándares de calidad y seguridad establecidos. (mineduc, 2015)

Los aprendizajes esperados en este módulo son:

1. Instala sistemas operativos en computadores personales a nivel local, de acuerdo procedimientos técnicos establecidos en manuales y a las necesidades y requerimientos de usuarios.
2. Instala sistemas operativos con servidores de red para multiusuarios considerando características tecnológicas y licenciamiento respectivo.

3.2.5 Diseño de bases de datos relacionales.

Este módulo parece centrarse en la representación de datos y su uso para la toma de decisiones empresariales. Los y las estudiantes aprenderán a desarrollar modelos de datos utilizando NetBeans y MySQL, así como a crear diagramas entidad-relación para representar situaciones y problemas del mundo real.

El uso de herramientas como *PowerDesigner* o *Visio* se sugiere para la diagramación, y se espera que los y las estudiantes conozcan el modelo entidad-relación y sus procesos, así como el lenguaje utilizado al respecto. La capacidad de construir modelos de datos que respondan a las necesidades transaccionales de un negocio e implementarlos en la base de datos es un objetivo importante de este módulo. Por otro lado, el objetivo principal es que aprendan a instalar sistemas de información manufacturados para obtener información necesaria de acuerdo a los requerimientos de la empresa.

Las actividades prácticas en el laboratorio de computación son una parte importante de este módulo, donde los y las estudiantes construyen modelos de datos y trabajan en la programación de algoritmos. (mineduc, 2015)

Los aprendizajes esperados en este módulo son:

1. Instala sistemas de manejo de datos y entornos de diagramación gráfica para representar relaciones entre datos, según especificaciones técnicas y manual de instalación.
2. Construye tablas de datos para representar la estructura de los datos requeridos para resolver un problema simple, en función de las principales entidades y relaciones entre ellas, de acuerdo con un modelo establecido.
3. Construye modelos de datos que respondan a las necesidades de información transaccional de negocios, aplicando técnicas relacionales, de acuerdo con protocolos definidos.
4. Implementa modelos de representación gráfica en una base de datos (BBDD); según protocolos y el lenguaje de programación correspondiente.

3.2.6 Programación orientada a objetos.

Este módulo está enfocado en introducir a los estudiantes a los conceptos fundamentales de la programación orientada a objetos (POO). Se espera que los estudiantes aprendan a resolver situaciones de negocio empresariales utilizando la programación Java y expandiendo su lógica resolutoria. Para ello, se utilizará el IDE NetBeans y MySQL como entorno de desarrollo. El objetivo principal es que aprendan a construir aplicaciones computacionales basadas en programación orientada a objetos, de manera de cumplir con las exigencias técnicas y de los usuarios.

El enfoque de las clases es práctico y teórico, y se espera que los estudiantes apliquen sus conocimientos y habilidades previas en diagramación. Además, es importante que las actividades incorporen metodologías activas y el cumplimiento de estándares. En general,

el objetivo es que los estudiantes puedan abordar cualquier lenguaje POO y metodología de análisis, y diseño POO de sistemas gestores de bases de datos. (mineduc, 2015)

Los aprendizajes esperados en este módulo son:

1. Construye unidades de prueba para verificar el correcto funcionamiento de la codificación realizada, de acuerdo con las exigencias técnicas de confiabilidad.
2. Detecta y corrige errores de codificación, analizando el comportamiento del código de programación, de acuerdo con las especificaciones y manuales de referencia.
3. Construye aplicaciones con clases predefinidas, utilizando herramientas del lenguaje de programación para resolver problemas complejos, de acuerdo con las especificaciones técnicas.
4. Utiliza componentes reutilizables en un lenguaje de uso estándar, para ser aplicados en programas de diversa índole, de acuerdo con los requerimientos y especificaciones técnicas.

3.2.7 Administración de bases de datos.

Este módulo se enfoca en el desarrollo de habilidades y competencias en el manejo de bases de datos y programación en PL/SQL, con el objetivo de que los y las estudiantes puedan resolver situaciones complejas de levantamiento de bases de datos y aplicar la lógica resolutoria de casos de negocios empresariales. El objetivo principal es desarrollar aplicaciones y rutinas para el mantenimiento y actualización de bases de datos de acuerdo con los requerimientos de la empresa.

Se utiliza SQL Developer y Oracle como herramientas principales de desarrollo. Además, se promoverá la realización de actividades prácticas y teóricas que incorporen metodologías activas.

Se espera que los y las estudiantes sean capaces de construir unidades de proyecto, planificar proyectos, crear elementos dinámicos de una web y programar unidades de código utilizando los recursos del lenguaje SQL y PL/SQL, considerando las sintaxis, restricciones, requisitos de la lógica de negocios y de información. (mineduc, 2015)

Los aprendizajes esperados en este módulo son:

1. Desarrollar unidades de programación complejas sobre una base de datos, según información de lenguajes estándar de la industria.

2. Programa aplicaciones para administrar una base de datos, realizando operaciones complejas que permitan su mantenimiento y actualización según, restricciones y requisitos de la lógica de negocios y de información.

3.2.8 Desarrollo de aplicaciones web.

Este módulo de programación orientada a objetos está diseñado para que los estudiantes adquieran habilidades prácticas en el desarrollo de aplicaciones web. El enfoque pedagógico en la simulación de un ambiente de trabajo y el uso de tecnologías abiertas y libres es muy positivo, ya que les permite a los estudiantes familiarizarse con herramientas que son ampliamente utilizadas en la industria. El objetivo principal es desarrollar aplicaciones web acordes a los requerimientos y utilizando los lenguajes de programación disponibles en el mercado, permitiendo la administración y/publicación de información en internet.

El uso del IDE NetBeans junto con MySQL y MySQL Connector es una buena elección para el entorno de desarrollo, ya que NetBeans es muy popular y MySQL es un gestor de bases de datos ampliamente utilizado en la industria. Esta asignatura tiene un enfoque práctico y centrado en el estudiante, lo que debería ayudar a los estudiantes a desarrollar habilidades valiosas en programación orientada a objetos y en el desarrollo de aplicaciones web. (mineduc, 2015)

Los aprendizajes esperados en este módulo son

1. Diseña aplicaciones web en lenguajes estándar, para requerimientos específicos, considerando estándares de la industria y manuales técnicos.
2. Desarrolla elementos dinámicos web, utilizando tecnologías abiertas según los requerimientos de la organización.
3. Evalúa el funcionamiento de una aplicación web según los requerimientos del negocio, utilizando herramientas propias del entorno de desarrollo en tecnologías abiertas.

3.2.9 Emprendimiento y empleabilidad.

Promueve el pensamiento algorítmico y sugiere establecer relaciones con el sector productivo. (mineduc, 2015)

3.3 ASIGNATURAS DEL ÁREA TI EN EL TÉCNICO UNIVERSITARIO EN INFORMÁTICA USM

3.3.1 Análisis de Sistemas de Información

La asignatura Análisis de Sistemas de Información entrega los conocimientos fundamentales sobre los Sistemas de Información y sus etapas de desarrollo, con énfasis en Análisis de Sistemas.

Los contenidos a tratar en la asignatura son:

1. Conceptos básicos de sistemas de información.
 - Teoría General de Sistemas.
 - Administración de las organizaciones y Sistemas de Información.
 - Conceptos y elementos de un Sistema de Información.
 - Sistemas típicos en una empresa. (Soluciones a medida, ERP, SCM, CRM, otros)
 - Ciclo de vida de un Sistema de Información.
 - Metodologías (Clásicas y Ágiles).
2. Identificación de requerimientos y estudio de factibilidad.
 - Generalidades de la preparación y evaluación de proyectos.
 - Diagnóstico y Análisis de requerimientos.
 - Metodologías de estimación de duración y costos.
 - Factibilidad Técnica, Operacional y Económica de un Sistema de Información.
3. Metodologías de análisis.
 - Fase de análisis en el desarrollo de un Sistema de Información.
 - Generalidades. Aspectos teóricos.
 - Aspectos prácticos: Desarrollo de casos de estudio.

Los resultados de aprendizaje que se esperan lograr son:

- I. Caracteriza los Sistemas de Información y sus diversas áreas de aplicación, bajo el enfoque de sistemas. (Laudon, 2012)

Con los resultados de aprendizaje lo que se espera lograr, es la caracterización de los Sistemas de Información y sus diversas áreas de aplicación bajo el enfoque de sistemas. Los Sistemas de Información (SI) se caracterizan por ser estructuras organizadas que recopilan, procesan, almacenan y distribuyen datos para apoyar la toma de decisiones y el funcionamiento de una organización.

Los Sistemas de Información tienen diversas áreas de aplicación, algunas de las cuales son:

- **Sistemas de Gestión Empresarial (ERP):** Es un software de gestión empresarial que permite a una organización aprovechar un conjunto de aplicaciones integradas, agilizando y automatizando los procesos y creando un funcionamiento más ágil, preciso y eficiente.
- **Sistemas de Gestión de la Cadena de Suministro (SCM):** Es la gestión del flujo de bienes, datos y finanzas relacionados con un producto o servicio, desde la adquisición de materias primas hasta la entrega del producto en su destino final.
- **Sistemas de Gestión de Relaciones con los Clientes (CRM):** Es un software empresarial que tiene como objetivo unificar y gestionar las relaciones de la empresa con sus clientes a través de todos los canales de comunicación para aumentar la satisfacción y fidelidad.

- II. Define las etapas del ciclo de vida de un Sistema de Información, con énfasis en el estudio de factibilidad y análisis. (Berzal, 2005-2006)

Un sistema de información tiene un ciclo de vida que comienza por la creación, implementación y mantenimiento. Sus principales etapas son:

- **Identificación de necesidades:** Se analizan e identifican los problemas y necesidades del sistema, al igual que los objetivos planteados.
- **Estudio de factibilidad:** Se analizan factores técnicos, operacionales y económicos, para verificar si es viable desarrollar el sistema.
- **Análisis de requerimientos:** Se realiza un análisis detallado de los procesos, datos y usuarios involucrados, para verificar las funciones, características y restricciones a cumplir.
- **Diseño del sistema:** Se elabora un diseño técnico que guiará la implementación, donde se analiza y define la arquitectura, estructura y componentes a utilizar para el sistema.
- **Desarrollo e implementación:** Se realiza la programación, configuración y pruebas del sistema. A continuación, la implementación, instalación y la ejecución en el entorno de producción.
- **Evaluación y mantenimiento:** Se realiza una evaluación del funcionamiento y modificaciones del sistema. Se realiza el mantenimiento, correcciones de errores, actualizaciones y mejoras del sistema.

- III. Aplica metodología para el análisis de Sistemas de Información, identificando y documentando requerimientos de negocio.

Los sistemas de información por medio de un análisis implican el uso de metodologías para identificar, comprender y documentar los requerimientos de negocio. Algunas etapas son: (pmoinformatica, 2018)

- Recopilación de información: Se recopila y mide la información de diversas fuentes, a fin de obtener un panorama completo, responder preguntas importantes, evaluar sus resultados y anticipar futuras tendencias.
- 2. Entrevistas y reuniones: Se llevan a cabo entrevistas con los stakeholders y usuarios para comprender sus necesidades y expectativas. Se realizan reuniones de trabajo para obtener información adicional y resolver dudas.
- 3. Modelado de procesos: Es el estudio de los procesos de negocio con el fin de desarrollar un modelo abstracto sobre el mismo que permita comprender el proceso y comunicarlo con otros, por ejemplo, utilizando diagramas de flujo, BPMN, etc.
- 4. Identificación de requerimientos: Se identifican los requerimientos funcionales y no funcionales del sistema, es decir, las funciones y características que debe tener, además las restricciones y criterios de calidad.
- 5. Documentación de requerimientos: es el lugar donde se da descripción a las características y requisitos de un software, producto, programa o conjunto de programas. Los requisitos se expresan en lenguaje natural, sin consideraciones ni términos técnicos.

IV. Define el rol del personal de informática dentro del contexto del desarrollo de Sistemas de Información.

Los trabajadores del área informática tienen gran presencia en el desarrollo de Sistemas de información. Principalmente en los cargos de:

- Analista de sistemas: Encargado del desarrollo de aplicaciones en lo que respecta a su diseño y obtención de los algoritmos, así como de analizar las posibles utilidades y modificaciones necesarias de los sistemas operativos para una mayor eficacia de un sistema informático.
- Programador o Desarrollador: Se encarga de la codificación y desarrollo del sistema de acuerdo con los requerimientos y el diseño establecido. Utiliza lenguajes de programación, frameworks y herramientas de desarrollo.
- Administrador de Bases de Datos: Administra las tecnologías de la información y la comunicación, siendo responsable de los aspectos técnicos, tecnológicos, científicos, inteligencia de negocios y las legalidades de las bases de datos, y de su calidad de datos.

- Ingeniero de Software: Se encarga de la creación de software confiable y de calidad, basándose en métodos y técnicas de ingeniería de software, y brindando soporte operacional y de mantenimiento. Aplica principios y metodologías de desarrollo de software.
 - Especialista en Seguridad de la Información: Se ocupa de garantizar la seguridad de los sistemas de información, identificando y mitigando posibles vulnerabilidades y riesgos de seguridad.
- V. Relaciona funciones básicas de la administración en el contexto de desarrollo de sistemas de información, determinando objetivos, metas y planes para alcanzar logros establecidos.

La administración es la parte esencial del desarrollo de sistemas de información. Entre sus funciones están:

- Planificación: Se establecen los objetivos y metas del proyecto de desarrollo de sistemas de información. Se elabora un plan que define las actividades, los recursos necesarios, los plazos y los responsables.
- Organización: Se asignan roles y responsabilidades a los miembros del equipo de desarrollo. Se establecen las estructuras de comunicación y coordinación necesarias para el proyecto.
- Dirección: Se lidera y se guía al equipo de desarrollo de sistemas de información. Se establecen las estrategias y se toman decisiones para asegurar el avance del proyecto y el cumplimiento de los objetivos.
- Control: Se monitorea el progreso del proyecto, se evalúa el cumplimiento de los plazos y los resultados obtenidos. Se toman acciones correctivas en caso de desviaciones o riesgos identificados.
- Organización de recursos: Se gestionan los recursos necesarios para el desarrollo del sistema de información, como personal, presupuesto, tecnología y tiempo. Se asegura la disponibilidad y adecuada asignación de los recursos.

3.3.2 Programación

Esta asignatura, permite desarrollar capacidades analíticas para resolución de un problema mediante un algoritmo computacional.

Además, permite conocer y aplicar nociones básicas de programación, usando para ello las estructuras básicas de un lenguaje de programación, creación de subprogramas y uso de estructura de datos arreglo y archivos de texto.

Los contenidos a tratar en la asignatura son:

1. Conceptos básicos de programación
 - Algoritmos y programas computacionales.
 - Lenguajes de programación: conceptos, categorías, traductores.
 - Elementos básicos de un programa computacional.
2. Enfoque algorítmico en la resolución de problemas y utilización de una herramienta de diseño de algoritmos.
 - Nociones sobre herramientas de diseño de algoritmos: pseudo lenguaje, diagramas de flujo.
 - Definición y aplicación de un pseudo lenguaje (tipos de datos, constantes variables, operadores, expresiones, sentencias, otros).
 - Aplicación de técnicas de programación (estructurada, modular, enfoque top-down).
 - Utilización estructura de datos arreglo.
3. Nociones del lenguaje de programación a utilizar
 - Estructura de un programa, tipos de datos, uso de expresiones.
 - Instrucciones de: asignación, decisión, iteración, lectura/escritura y otras.
 - Utilización de funciones y procedimientos. Uso de parámetros.
 - Aplicación de estructuras de control condicionales y repetitivas, para la creación de programas.
4. Aspectos misceláneos.
 - Uso de funciones de biblioteca y manejo de archivos de texto, para resolución de diversos problemas.

Los resultados de aprendizaje que se esperan lograr son:

- I. Identifica y relaciona conceptos fundamentales asociados a la programación de computadores (software, algoritmo, programa computacional, lenguaje de programación, etc.) en forma consistente.
 - Software: Se refiere a los programas y datos que se ejecutan en un computador. Incluye el sistema operativo, aplicaciones y otros componentes necesarios para el funcionamiento del computador.
 - Algoritmo: Es un conjunto de instrucciones definidas, ordenadas y acotadas para resolver un problema, realizar un cálculo o desarrollar una tarea. Es decir, un algoritmo es un procedimiento paso a paso para conseguir un fin. A partir de un estado e información iniciales, se siguen una serie de pasos ordenados para llegar a la solución de una situación. (Maluenda, 2021)

- Programa computacional: son un conjunto de instrucciones en forma secuencial, llamado código, que, a través de su interpretación por el sistema operativo o hardware, le permiten desarrollar una acción específica a una computadora portátil o de escritorio. (Euroinnova, n.d.)
 - Lenguaje de programación: Es el conjunto de instrucciones a través del cual los humanos interactúan con las computadoras. Un lenguaje de programación nos permite comunicarnos con las computadoras a través de algoritmos e instrucciones escritas en una sintaxis que la computadora entiende e interpreta en lenguaje de máquina. (Mendoza, 2020)
 - Compilador e intérprete: Son herramientas utilizadas para convertir el código fuente de un programa escrito en un lenguaje de programación a un código ejecutable por el computador. El compilador traduce el código completo de una vez, mientras que el intérprete lo traduce y ejecuta línea por línea.
- II. Diseña algoritmos computacionales, usando pseudocódigo o diagrama de flujo (una de ellas) para resolver un problema, el cual, por su naturaleza, puede ser resuelto a través del uso de computadores.
- Diseño de algoritmos computacionales implica la creación de un conjunto de pasos lógicos y secuenciales para resolver un problema utilizando un computador.
 - El pseudocódigo es una forma de representar código, como algoritmos, funciones y otros procesos, utilizando una combinación de lenguaje natural y elementos similares al lenguaje de programación. Se llama «pseudocódigo» porque no es realmente ejecutable. En cambio, es una forma de que los humanos comprendan y planifiquen la lógica de la programación. (kinsta, 2023)
 - Diagrama de flujo es un diagrama que describe un proceso, sistema o algoritmo informático. Se usan ampliamente en numerosos campos para documentar, estudiar, planificar, mejorar y comunicar procesos que suelen ser complejos en diagramas claros y fáciles de comprender. Utiliza símbolos gráficos para representar cada paso del algoritmo y las decisiones tomadas. (lucidchart, n.d.)
- III. Aplica estructura de datos arreglo, subprogramas y archivos de texto, según requerimientos en un programa computacional.
- Arreglo: es un dato estructurado que almacena un conjunto de datos homogéneo (todos del mismo tipo y relacionados). Cada uno de los elementos que componen un vector pueden ser Simples, Compuestos o estructurados. Almacenan un conjunto de elementos del mismo tipo de manera secuencial en

la memoria. Los arreglos permiten acceder y manipular los elementos utilizando un índice numérico. (Serrano, 2022)

- Subprogramas: Son secciones de código que realizan una tarea específica dentro de un programa. Pueden recibir parámetros, ejecutar un conjunto de instrucciones y retornar un resultado. Además, permiten modularizar y reutilizar el código, mejorando la estructura y legibilidad del programa.
- Archivos de texto: Es aquel que almacena información de texto plano, es decir que no contiene caracteres de control, sólo contiene caracteres imprimibles como letras, números y símbolos, por lo general del juego de caracteres ASCII. El estándar ASCII permite que los archivos de texto puedan ser intercambiados libremente y legible entre plataformas como Unix, Macintosh, Microsoft Windows, DOS. (cavsi, n.d.)

IV. Implementa programas de acuerdo a requerimientos de usuario utilizando una metodología algorítmica y técnicas de programación aplicando las estructuras de control y de datos básicos en un lenguaje de programación.

- Estructuras de control: son el conjunto de reglas que permiten controlar el flujo de ejecución de las instrucciones de un algoritmo o de un programa. (Durán, 2022)
- Estructuras de datos básicas: Es una colección de valores, la relación que existe entre estos valores y las operaciones que podemos hacer sobre ellos, en pocas palabras se refiere a cómo los datos están organizados y cómo se pueden administrar. (Sena, 2019)

3.3.3 Introducción a la Informática y Computación

La asignatura entrega los conocimientos fundamentales e introductorios a la especialidad, sobre el hardware de computadores y aplicaciones informáticas necesarias.

Los contenidos a tratar en la asignatura son:

1. Historia y evolución de la computación.
 - Antecedentes Históricos.
 - Evolución y Desarrollo de las Tecnologías de la Información.
2. Sistema computacional: hardware y software.
 - Características funcionales de los componentes de un computador.
 - Software y su clasificación.
3. Modos de procesamiento.

- Conceptos de procesamiento por lote y en línea.
 - Conceptos de multitarea, multiprocesamientos, multiprogramación y tiempo compartido.
4. Sistemas de representación de datos.
- Sistemas Numéricos, transformación entre bases numéricas y compuertas.
 - Introducción a sistemas combinacionales.
 - Representación de Datos en Memoria.
 - Representación de datos en Medios de almacenamiento secundarios.
5. Introducción a Arquitectura de computadores.
- Procesador, Memoria Principal, Dispositivos Periféricos, Integración de los distintos componentes.
6. Herramientas de Ofimática.

Los resultados de aprendizaje que se esperan lograr son:

- I. Identifica los aspectos fundamentales propios de la especialidad, definiendo conceptos del ámbito informático en el contexto de la historia, evolución y clasificación de la computación.

La computación es un área multidisciplinaria que se ha desarrollado por años y abarca diversos conceptos y tecnologías. Fundamentos y conceptos de área informática:

- Historia y evolución de la computación: Desde los primeros dispositivos mecánicos como la máquina de Turing hasta los ordenadores modernos, la historia de la computación ha experimentado avances significativos en términos de capacidad de procesamiento, tamaño y portabilidad. Se destacan hitos como la invención de los circuitos integrados, el desarrollo de la *World Wide Web* y la popularización de las computadoras personales. (Fielding, et al., n.d.)
- Clasificación de la computación: La computación se puede clasificar en diferentes áreas, como la computación de alto rendimiento (*HPC*), la computación en la nube (*cloud computing*), la inteligencia artificial (*AI*), la ciberseguridad, la computación móvil, entre otras. Cada área se enfoca en aspectos específicos de la computación y tiene sus propias características y aplicaciones.
- Conceptos clave: En el ámbito informático, se utilizan conceptos como algoritmos, lenguajes de programación, sistemas operativos, bases de datos,

redes de computadoras, seguridad informática, inteligencia artificial, realidad virtual, entre otros. Estos conceptos son fundamentales para comprender y trabajar en el campo de la informática.

- II. Describe los elementos principales de un computador, tanto de hardware como software, identificando sus características, funcionalidades e interrelaciones.

Un computador está compuesto por elementos de hardware y software que trabajan juntos para realizar tareas de procesamiento de información. Algunos elementos principales son:

- **Hardware:** Es la parte física de un ordenador o sistema informático. Está formado por los componentes eléctricos, electrónicos, electromecánicos y mecánicos, tales como circuitos de cables y luz, placas, memorias, discos duros, dispositivos periféricos y cualquier otro material en estado físico que sea necesario para hacer que el equipo funcione.
- **Software:** Es un conjunto de reglas o programas que dan instrucciones a un ordenador para que realice tareas específicas. También se conoce como aplicaciones de software, paquetes de software, herramientas de software y programas de software. El software puede utilizarse para gestionar datos, automatizar procesos y crear aplicaciones o productos informáticos. Su complejidad puede variar desde un simple programa de tratamiento de textos hasta complejos sistemas informáticos que controlan infraestructuras críticas en sectores como la sanidad y el transporte.

- III. Describe los modos de procesamiento, considerando sus aplicaciones, de acuerdo a diferentes contextos.

Hay una variedad de procesamiento los cuales se aplican a distintos contextos, dependiendo de las necesidades y requerimientos del procesamiento de la información. Tales como:

- **Procesamiento en lote:** Es el método que utilizan las computadoras para completar periódicamente trabajos de datos repetitivos y de gran volumen. Ciertas tareas de procesamiento de datos, como las copias de seguridad, el filtrado y la clasificación, pueden requerir un esfuerzo de computación intensivo y ser ineficientes para ejecutarse en transacciones de datos individuales. (AWS, n.d.)
- **Procesamiento interactivo:** Son procesos cuyo comportamiento está limitado por operaciones de Entrada/Salida (I/O-bounded, en inglés). Este tipo de

proceso realiza operaciones de Entrada/Salida con frecuencia. Por tanto, el proceso suele pasar a estado bloqueado al poco tiempo de estar asignado al procesador. Un ejemplo de este tipo de proceso es un editor de texto que solicita datos del teclado de manera continuada.

- Procesamiento en tiempo real: se define como el procesamiento de flujos no asociados de datos de entrada, con requisitos de latencia muy cortos para el procesamiento, que se miden en milisegundos o segundos. Estos datos de entrada normalmente llegan en un formato no estructurado o semiestructurado, como JSON, y tienen los mismos requisitos de procesamiento que el procesamiento por lotes, pero con tiempos de respuesta más cortos para admitir el consumo en tiempo real. (M, n.d.)

- IV. Aplica los sistemas numéricos y de representación de datos, de acuerdo a método definido.

En el ámbito informático, se utilizan diferentes sistemas numéricos y métodos de representación de datos. Tales como:

- Sistema binario: Es el sistema numérico fundamental en la computación. Utiliza dos dígitos, 0 y 1, para representar los valores numéricos y las señales electrónicas en los circuitos digitales. El sistema binario es utilizado por los componentes electrónicos de los computadores para el procesamiento de información.
- Sistema hexadecimal: Es un sistema numérico que utiliza 16 dígitos: los dígitos del 0 al 9 y las letras A, B, C, D, E, F. Se utiliza ampliamente en la programación y representación de datos debido a su conveniencia para expresar valores binarios de manera más compacta y legible.

- V. Utiliza herramientas de ofimática para solucionar problemas de mediana complejidad.

Las herramientas de ofimática son aplicaciones informáticas diseñadas para facilitar tareas de oficina y mejorar la productividad. Tales como:

- Procesadores de texto: Permite el tratamiento de textos (creación, edición, modificación y procesamiento de documentos de texto con formato tal como el tipo y tamaño de la tipografía, adición de gráficos, etc.) a diferencia de los editores de texto, que manejan solo texto simple. (Lasso, 2015)

- Hojas de cálculo: Son útiles para organizar, analizar y calcular datos numéricos. Se utilizan en tareas como presupuestos, planificación financiera, seguimiento de inventario, entre otros.
- Correo electrónico: Facilita la comunicación escrita a través de mensajes electrónicos. Se utiliza para enviar y recibir correos electrónicos, adjuntar archivos y gestionar la comunicación con colegas y clientes.
- Sistemas de gestión de bases de datos: Permiten administrar una base de datos. Esto significa que mediante este programa se puede utilizar, configurar y extraer información almacenada. Los usuarios pueden acceder a la información usando herramientas específicas de consulta y de generación de informes, o bien mediante aplicaciones al efecto. (Bachman, 2013)

3.3.4 Análisis y Diseño Orientado a Objeto

La asignatura entrega los conocimientos fundamentales del análisis y diseño orientado a objetos, utilizando el Lenguaje Unificado de Modelamiento bajo la perspectiva del Proceso Unificado.

Los contenidos a tratar en la asignatura son:

1. Introducción a Orientación a Objeto
 - Conceptos principales, Fundamentos del modelado Orientado a Objeto
2. Análisis Orientado a Objeto
 - Identificación de Requerimientos, Especificación de Requerimientos, Modelos de Análisis
3. Diseño Orientado a Objeto
 - Principios, Jerarquía, Atributos, Métodos, Encapsulamiento, Herencia y Polimorfismo
4. Introducción a UML (Lenguaje Unificado de Modelado): Lenguaje de modelado estándar usado en análisis y diseño orientado a objeto:
 - Conceptos generales, Historia de UML, Diagramas Básicos
5. Estudio del Lenguaje Unificado de Modelado:
 - Diagramas de Estructura, Diagramas de Comportamiento
6. Proceso de Desarrollo de Software basado en UML:
 - Proceso Unificado, Fases del Ciclo de Vida
7. Desarrollo de un proyecto utilizando UML

Los resultados de aprendizaje que se esperan lograr son:

- I. Describe los conceptos fundamentales relacionados al Análisis y Diseño Orientado a Objeto para el desarrollo de sistemas informáticos.
 - Análisis Orientado a Objeto (OOA): Consiste en desarrollar un modelo que describa el software de computadora necesario para satisfacer los requisitos definidos por el cliente. El modelo de análisis contiene el funcionamiento y el comportamiento de los elementos del modelo de objetos. (Pascual, 2013)
 - Diseño Orientado a Objeto (OOD): Es el proceso de usar una metodología orientada a objetos para diseñar un sistema informático o una aplicación. Esta técnica permite la implementación de una solución de software basada en los conceptos de objetos. (theastrology, 2022)
 - Principios SOLID: SOLID es un acrónimo que representa cinco principios utilizados por ingenieros de software y desarrolladores. Los cuales ayudan a crear sistemas más mantenibles, escalables y flexibles. Estos principios son los siguientes, Single Responsibility (Responsabilidad única), Open/Closed (Abierto/Cerrado), Liskov substitution (Sustitución de Liskov), Interface segregation (Segregación de interfaz) y Dependency Inversión (Inversión de dependencia). (FUSIONA, 2022)

- II. Describe los conceptos fundamentales y los distintos diagramas relacionados al Lenguaje Unificado de Modelado, bajo el enfoque de orientación a objetos.
 - Lenguaje de Modelado Unificado (UML): Es una herramienta que ayuda a capturar mediante un conjunto de símbolos y diagramas a comunicar la idea de un sistema (software orientado a objetos), a quien esté involucrado en su proceso de desarrollo sirviendo de apoyo en los procesos de análisis y diseño de un problema. (Monsivais, 2020)
 - Diagramas UML: Permiten graficar un proyecto complejo, para conocer sus componentes, relaciones y funcionamiento, según sean las necesidades del proyecto. Además, están estandarizados, y cuentan con un lenguaje semántico para que sea comprensible por diferentes actores e interesados en el proyecto. (Mancuzo, 2021)

- III. Desarrolla diagramas a nivel de análisis orientado a objetos, de acuerdo a requerimientos dados.

Durante el análisis orientado a objetos, se pueden desarrollar varios diagramas para comprender y especificar los requisitos del sistema. Algunos de los diagramas utilizados a nivel de análisis son:

- Diagrama de casos de uso: Representa las interacciones entre actores (usuarios) y el sistema. Muestra las funcionalidades y las interacciones de alto nivel.
- Diagrama de clases preliminar: Identifica las clases principales, sus atributos y las relaciones entre ellas. Ayuda a establecer una estructura inicial del sistema.

IV. Desarrolla diagramas a nivel de diseño orientado a objetos, de acuerdo a requerimientos.

A nivel de diseño orientado a objetos, se desarrollan diagramas más detallados para especificar las estructuras y comportamientos del sistema. Algunos de los diagramas utilizados son:

- Diagrama de secuencia: es un tipo de diagrama de interacción porque describe cómo un grupo de objetos funcionan en conjunto. (lucidchart, lucidchart, n.d.)
- Diagrama de clases detallado: Refina el diagrama de clases preliminar del análisis y especifica con mayor detalle las clases, los atributos, los métodos y las relaciones. En este diagrama se representa la estructura y el comportamiento de cada uno de los objetos del sistema y sus relaciones con los demás objetos, pero no muestra información temporal. (cillero, n.d.)
- Diagrama de componentes: representan las relaciones entre los componentes individuales del sistema mediante una vista de diseño estática. Pueden ilustrar aspectos de modelado lógico y físico. (IONOS, 2020)
- Diagrama de despliegue: se utilizan para visualizar los procesadores/nodos/dispositivos de hardware de un sistema, los enlaces de comunicación entre ellos y la colocación de los archivos de software en ese hardware. (creately, 2022)

V. Describe las etapas del proceso de desarrollo de software basado en el Lenguaje Unificado de Modelado, destacando sus principales características.

El proceso de desarrollo de software basado en UML, etapas: (UACH, n.d.)

- Requisitos: La captura de requisitos tiene dos objetivos: encontrar los verdaderos requisitos (funcionales y no funcionales) y representarlos de un modo adecuado para los usuarios, clientes y desarrolladores (modelo de casos de uso). Además, se recopilan y analizan los requisitos del sistema, identificando las necesidades y expectativas de los usuarios y estableciendo los objetivos del sistema.

- **Análisis:** El propósito fundamental del análisis es refinar los casos de uso con mayor profundidad, pero con la gran diferencia que puede utilizarse el lenguaje de los desarrolladores para describir los resultados. Se modelan los conceptos y estructuras del sistema utilizando diagramas UML a nivel de análisis. Se identifican los casos de uso, se desarrollan los diagramas de clases preliminares y se especifican los requisitos funcionales.
 - **Diseño:** Se realiza el diseño detallado del sistema utilizando diagramas UML a nivel de diseño. Se refinan los diagramas de clases, se desarrollan diagramas de secuencia, de componentes, de despliegue, entre otros.
 - **Implementación:** Consiste en implementar el sistema en términos de componentes, es decir, archivos de código fuente, scripts, archivos de código binario, ejecutables y similares. El propósito principal es desarrollar la arquitectura y el sistema como un todo.
 - **Pruebas:** Se verifica el resultado de la implementación probando cada construcción a través de casos de prueba que especifican una forma de probar el sistema. Se identifican y corrigen errores o problemas.
 - **Despliegue:** Se lleva a cabo la instalación y puesta en marcha del sistema en el entorno de producción.
- VI. Aplica metodología de análisis y diseño orientado a objeto, en el desarrollo de un proyecto informático propuesto.
- La metodología de análisis y diseño orientado a objeto se aplica siguiendo los pasos del proceso de desarrollo de software. Se realizan análisis de requerimientos, se desarrollan diagramas a nivel de análisis y diseño, se implementa el código del sistema y se llevan a cabo pruebas. Se utiliza el UML para representar visualmente los diferentes aspectos del sistema y se aplican los principios y patrones de diseño orientado a objeto para crear un sistema eficiente, mantenible y escalable.

3.3.5 Diseño de Sistemas de Información

La asignatura Diseño Sistemas de Información, entrega los conocimientos fundamentales sobre las etapas de diseño, construcción, prueba, implementación y mantención de Sistemas de Información, con énfasis en la etapa de Diseño.

Los contenidos a tratar en la asignatura son:

1. Fundamentos de la Ingeniería de Software para la fase del Diseño.
 - Conceptos fundamentales de la Ingeniería de Software relacionados al diseño.

2. Caracterización de la fase de Diseño de Software.
 - Diseño físico: objetivos, entradas, salidas y actividades.
 - Estructuración del diseño en procesos y programas (módulos).
 - Especificación de los procesos.
 - Especificación de los programas.
 - Especificación de tablas y archivos.
 - Especificación de las pantallas.
3. Diseño de datos.
 - Características del diseño de los datos.
 - Normalización.
4. Diseño de Programas.
 - Tipos de programas.
 - Diseño Arquitectónico.
 - Diseño Procedural.
5. Diseño de Interfaz de Usuario.
 - Características del diseño de la interfaz.
6. Construcción, prueba, implementación y mantenimiento de Sistemas.

Los resultados de aprendizaje que se esperan lograr son:

- I. Describe las actividades del diseño, construcción, prueba, implementación y mantenimiento de Sistemas de Información, según metodología utilizada.

Las metodologías utilizadas en el desarrollo de Sistemas de Información suelen incluir diversas etapas y actividades. A continuación, se describen algunas de las actividades comunes en cada una de las fases:

1. Diseño: (mdcloud, n.d.)
 - Diseño conceptual: En esta fase se representan una descripción a alto nivel del contenido de la base de datos, independientemente del sistema de gestión de base de datos que se utilizará a continuación. Se definen en un dibujo las entidades, sus atributos y las relaciones entre ellas.
 - Diseño lógico: En esta fase, se traduce el modelo conceptual obtenido con anterioridad a un esquema lógico, que describe la estructura de la base de datos. Se trata de la fase en la cual se diseñan las tablas propiamente dichas, con sus filas, columnas y relaciones. El modelo lógico depende del SGBD que se utilizará.

- **Diseño físico:** En esta fase se definen las estructuras de almacenamiento de la base de datos de forma física. Es cuando se escribe el código (por ejemplo, SQL) para concretar el diseño en el motor de base de datos que hemos escogido.
- **Construcción:**
 - **Programación:** Se escribe el código del sistema utilizando el lenguaje de programación seleccionado, siguiendo las especificaciones y el diseño establecido.
 - **Pruebas unitarias:** Se realizan pruebas para verificar que cada componente del sistema funcione correctamente de manera individual.
- **Prueba:**
 - **Pruebas de integración:** se realizan en el ámbito del desarrollo de software una vez que se han aprobado las pruebas unitarias y lo que prueban es que todos los elementos unitarios que componen el software funcionan juntos correctamente probándolos en grupo.
 - **Pruebas de sistema:** Se comprueba cómo funcionan juntos los distintos componentes del software y si se integran sin problemas entre sí. Se llevan a cabo pruebas exhaustivas del sistema en su conjunto para garantizar que cumpla con los requisitos y funcione correctamente. (zaptest, n.d.)
- **Implementación:**
 - **Instalación:** Se implementa el sistema en el entorno de producción, lo que incluye la instalación del hardware, el software y la configuración necesaria.
 - **Migración de datos:** Se transfieren los datos existentes al nuevo sistema, asegurando su integridad y consistencia.
- **Mantenimiento:**
 - **Soporte y corrección de errores:** Es una importante práctica para el mantenimiento e integridad de los datos a través de diferentes procedimientos y dispositivos como medios de almacenamiento confiables. Se brinda soporte a los usuarios y se corrigen los errores o problemas identificados durante el uso del sistema. (Layton, 2014)
 - **Mejoras y actualizaciones:** Se realizan mejoras y actualizaciones del sistema para agregar nuevas funcionalidades, mejorar el rendimiento o adaptarse a cambios en los requerimientos.

- II. Describe la fase de diseño de Sistemas de Información, según metodología definida. (Martínez, n.d.)
- Diseño de los datos: Transforma el modelo de dominio de la información, creado durante el análisis, en las estructuras de datos necesarios para implementar el software.
 - Diseño Arquitectónico: Define la relación entre cada uno de los elementos estructurales del programa.
 - Diseño de la Interfaz: Describe como se comunica el software consigo mismo, con los sistemas que operan en conjunto, operadores y usuarios que los emplean.
 - Diseño de procedimientos: Transforma elementos estructurales de la arquitectura del programa. La importancia del diseño del software se puede definir en una sola palabra calidad, dentro del diseño es donde se fomenta la calidad del proyecto. El diseño es la única manera de materializar con precisión los requerimientos del cliente.
- III. Aplica los diferentes tipos de diseño de Software, de acuerdo a requerimientos definidos.

En el desarrollo de software, existen diferentes tipos de diseño que se utilizan según los requerimientos y las características del sistema. Algunos de los tipos de diseño comunes son

- Diseño estructurado: es el proceso de decidir qué componentes, y la interconexión entre los mismos, para solucionar un problema bien especificado. El diseño estructurado utiliza técnicas como el diagrama de flujo, el diagrama de estructura de programa y el pseudocódigo. (altp, 2018)
- Diseño orientado a objetos: Es una de las principales formas de diseñar software porque imita la vida real. En el software, un objeto puede tener varias propiedades y realizar tareas. Los diagramas UML, como el diagrama de clases y el diagrama de secuencia, se utilizan para representar el diseño orientado a objetos. (INVID, n.d.)
- Diseño basado en componentes: Es una arquitectura basada en componentes describe una aproximación de ingeniería de software al diseño y desarrollo de un sistema. Esta arquitectura se enfoca en la descomposición del diseño en componentes funcionales o lógicos que expongan interfaces de comunicación bien definidas. Los componentes son unidades de software independientes que

encapsulan funcionalidades específicas y se pueden combinar para construir sistemas más grandes. (Geeks, n.d.)

- Diseño de arquitectura de software: Utiliza los conocimientos de programación para planear el diseño general del software de modo que puedan agregarse detalles más adelante, lo cual permite a los equipos de software delimitar el panorama general y comenzar a elaborar un prototipo. Se concentra en la estructura y organización del sistema en términos de componentes, módulos y subsistemas. Define las interfaces, las dependencias y la distribución física de los componentes. (lucidchart, lucidchart, n.d.)

IV. Describe las herramientas de la tecnología de Información relacionadas con el Diseño de Sistemas de Información.

- Herramientas de modelado: Garantiza un mínimo nivel de calidad en los modelos al comprobar que el modelo sea una instancia correcta de su metamodelo correspondiente. Estas herramientas permiten crear, visualizar modelos y diagramas utilizados en el diseño de sistemas de información. (Cabot, 2017)
- Herramientas CASE (Computer-Aided Software Engineering): Son herramientas que brindan soporte automatizado para el desarrollo de software. Estas herramientas incluyen funcionalidades para el modelado, el diseño, la generación de código, la documentación y la gestión de proyectos.
- Herramientas de gestión de bases de datos: Estas herramientas facilitan la creación, el diseño y la gestión de bases de datos, permitiendo la definición de tablas, la creación de consultas y la administración de la información. (Humano, n.d.)
- Herramientas de diseño de interfaz de usuario: Se enfoca en maximizar la usabilidad y la experiencia de usuario. El objetivo final del diseño de la interfaz de usuario es hacer que la interacción entre el usuario y el sistema del que es interfaz sea tan simple y eficiente como sea posible, en términos de cumplimiento de los objetivos del usuario. Permiten crear diseños interactivos, definir la disposición de los elementos y simular la experiencia del usuario. (Norman, 2018)

V. Describe conceptos de auditoria computacional, caracterizando sus elementos y utilidad.

La auditoría computacional es un proceso que tiene como objetivo evaluar y asegurar la integridad, confidencialidad, disponibilidad y cumplimiento de los sistemas y datos de información de una organización. (Gómez Ramírez, 2014)

- Auditoría interna: Es aquella que se hace desde el ambiente interno de la empresa, es decir, sin empleados ajenos a la misma, ya sea por empleados que fueron directamente contratados o alguna subsidiaria a esta y la entidad tiene el poder de decisión sobre el proceso llevado a cabo para realizarla.
- Auditoría externa: Se realiza por personal ajeno a la empresa ya sea remunerado o no y la organización no tiene poder de decisión sobre el proceso llevado a cabo por el ente que audita. Auditar consiste principalmente en estudiar los mecanismos de control que están implantados en una empresa u organización, determinando si los mismos son adecuados y cumplen unos determinados objetivos o estrategias, estableciendo los cambios que se deberían realizar para la consecución de estos.

3.3.6 Estructura de Datos

La asignatura entrega conocimientos de estructuras de datos básicas y reconozca cuándo usarlas en diferentes escenarios de su quehacer. Esto incluye el diseño, análisis, evaluación y programación de estructuras de datos usando la metodología de tipos de datos abstractos, capacitando al estudiante para resolver problemas de mediana complejidad utilizando un lenguaje de alto nivel. Además, entrega conocimientos fundamentales aplicables en las formas de organizar los archivos, métodos de acceso y su manejo.

Los contenidos a tratar en la asignatura son:

1. Técnicas de programación.
 - Programación modular, programación estructurada, recursividad, otras.
2. Elementos de estructuras de datos.
 - Conceptos y definiciones básicas.
 - Datos simples, datos estructurados.
 - Estructuras de datos estáticas, estructuras de datos dinámicas: listas, árboles y grafos.
 - Utilización de punteros para estructuras de datos.
3. Técnicas para almacenamiento y recuperación de datos.
 - Almacenamiento y acceso secuencial, directo.
 - Métodos de acceso: Búsqueda lineal, directa binaria, búsqueda indexada, técnicas de Hash-Code, otros.

4. Técnicas de ordenamiento:

- Por intercambio, por selección, por inserción, por mezclado, otras.

Los resultados de aprendizaje que se esperan lograr son:

I. Aplica las técnicas de programación estructurada y programación modular.

La programación estructurada se basa en la organización lógica y estructurada del código utilizando estructuras de control como secuencia, selección (if-else) y repetición (loops). Las tres estructuras básicas:

- Secuencia: La estructura secuencial es la que se da de forma natural en el lenguaje, porque las sentencias se ejecutan en el orden en el que aparecen en el programa, es decir, una detrás de la otra.
- Selección o condicional: La estructura condicional se basa en que una sentencia se ejecuta según el valor que se le atribuye a una variable booleana. ¡Un pequeño inciso! Una variable booleana es aquella que tiene dos valores posibles. Por tanto, esta estructura se puede ejecutar de dos formas distintas, dependiendo del valor que tenga su variable. Como apunte para los verdaderos amantes de la programación: para las estructuras condicionales o de selección, Python dispone de la sentencia if, que puede combinarse con elif y/o else.
- Iteración (ciclo o bucle): La estructura de repetición ejecuta una o un conjunto de sentencias siempre que una variable booleana sea verdadera. Para los bucles o iteraciones, los lenguajes de programación usan las estructuras while y for.

II. Aplica estructuras de datos (listas, árboles, grafos) de acuerdo a los requerimientos definidos.

Las estructuras de datos son organizaciones y formas de almacenamiento de datos en memoria o en disco. Tales como:

- Listas: Es una estructura de datos residente en la memoria que se llena con un conjunto de nombres extraídos de una fuente externa, como por ejemplo un archivo sin formato. Una vez creada y llenada con nombres, una lista de datos está disponible para utilizarla en las solicitudes de búsqueda subsiguientes. Cada lista de datos debe denominarse de forma exclusiva y debe estar en un formato específico. Las listas de datos pueden contener desde uno hasta cientos de millones de nombres. Pueden ser listas enlazadas o listas basadas en arreglos, y

permiten la inserción, eliminación y acceso eficiente a los elementos. (IBM, 2021)

- Árboles: se puede definir de forma recursiva (localmente) como una colección de nodos (a partir de un nodo raíz), donde cada nodo es una estructura de datos con un valor, junto con una lista de referencias a los nodos (los hijos), con la condición de que ninguna referencia esté duplicada ni que ningún nodo apunte a la raíz. Los árboles permiten realizar búsquedas, inserciones y eliminaciones eficientes.
- Grafos: Es una estructura matemática que permite modelar problemas de la vida cotidiana, mediante, como hemos visto, una representación gráfica formada por nodos o vértices que muestra a los actores y aristas que sirven para representar los lazos o relaciones entre los actores. Los grafos se utilizan en algoritmos de búsqueda, rutas más cortas, recorridos, entre otros. (Rochina, 2017)

III. Aplica recursividad como método de solución de problemas, según requerimientos.

La recursividad es una técnica de programación en la cual una función se llama a sí misma para resolver un problema de manera más simple. Como:

- Cálculos matemáticos: Algoritmos como el cálculo factorial, la sucesión de Fibonacci, la potencia de un número o la suma de elementos en una lista pueden implementarse de manera recursiva.
- Recorrido de estructuras de datos: Algunos algoritmos de recorrido de estructuras como árboles, grafos o listas enlazadas pueden implementarse de manera recursiva.
- Problemas de subdivisión: En algunos problemas, se puede aplicar la técnica de subdividir el problema en subproblemas más pequeños que se resuelven recursivamente, como en la técnica de dividir y conquistar.

IV. Aplica diversos métodos de almacenamiento, recuperación y ordenamiento de datos, según problema planteado.

En el manejo de datos, existen diferentes métodos y técnicas para almacenar, recuperar y ordenar la información de manera eficiente. Tales como:

- Almacenamiento en memoria principal: Se utiliza para almacenar datos en la memoria principal del computador, lo que proporciona un acceso rápido a los datos. Los arreglos y las listas enlazadas son estructuras de datos utilizadas para almacenar y acceder a datos en memoria principal.

- Almacenamiento en disco: Se utiliza para almacenar grandes volúmenes de datos en dispositivos de almacenamiento secundario, como discos duros. Los archivos se utilizan para organizar y acceder a los datos en disco.
- Métodos de ordenamiento: Son algoritmos que realizan la operación de arreglar los registros de una tabla en algún orden secuencial de acuerdo a un criterio de ordenamiento. El ordenamiento se efectúa con base en el valor de algún campo en un grupo de datos. El ordenamiento puede estar dado de forma iterativa o recursiva según la naturaleza y forma de ejecución del mismo. Algunos métodos de ordenamiento comunes son el ordenamiento burbuja (*BubbleSort*), el ordenamiento por selección (*SelectionSort*), el ordenamiento por inserción (*InsertionSort*), el ordenamiento por mezcla (*MergeSort*) y el ordenamiento rápido (*QuickSort*). (gbaudino, 2020)

V. Aplica distintos tipos de organización y métodos de acceso a archivos según requerimientos planteados.

El acceso y la organización de los archivos son aspectos críticos en el manejo de datos en un sistema de información. Algunos tipos de organización y métodos de acceso comunes son

- Organización secuencial: Es una sucesión de registros almacenados consecutivamente sobre el soporte externo, de tal modo que para acceder a un registro “n” dado es obligatorio pasar por todos los “n-1” artículos que le preceden. Es eficiente para lecturas secuenciales, pero no es adecuado para búsquedas aleatorias o actualizaciones frecuentes.
- Organización directa: Es un archivo está organizado en modo directo cuando el orden físico no se corresponde con el orden lógico. Los datos se sitúan en el archivo y se accede a ellos directa aleatoriamente mediante suposición, es decir, el lugar relativo que ocupan. (Mtz, n.d.)
- Organización de archivos de dispersión (hashing): Los registros se distribuyen en posiciones de almacenamiento utilizando una función hash. Permite un acceso rápido a registros a través de la función hash, pero puede haber colisiones que requieren técnicas de resolución.

3.3.7 Programación Orientada a Evento

La asignatura entrega los conocimientos fundamentales para la creación de aplicaciones orientada a eventos.

Los contenidos a tratar en la asignatura son:

1. Fundamentos de programación orientada al evento.
 - Conceptos generales de orientación al evento.
 - Propiedades y métodos de un evento.
 - Diferencia entre programa tradicional y orientado al evento.
2. Estudio de uno más lenguajes de programación orientado al evento.
 - Entorno de desarrollo.
 - Elementos del lenguaje.
 - Diseño de interfaz de usuario.
 - Gestión de objetos aplicados a eventos (creación, mantención, manipulación).
 - Manejo de bases de datos.
 - Generación de informes.
3. Construcción de aplicaciones.
 - Desarrollo de aplicaciones.
4. Tópicos de Especialidad.

Los resultados de aprendizaje que se esperan lograr son:

- I. Aplica conceptos de programación orientada al evento, según requerimientos definidos.

La programación orientada a eventos se refiere a un modelo de la programación de computadoras, donde se utilizan los eventos que suceden para la determinación del flujo de control de un programa. No es un tipo de tecnología o lenguaje de programación, sino un enfoque que se implementa durante la etapa de desarrollo del producto. Básicamente, separa la lógica de procesamiento de eventos del resto del código de un programa. Algunos conceptos clave de la programación orientada al evento son: (Chima, 2018)

- Eventos: Es una condición que surge durante la ejecución de un programa y que requiere alguna acción por parte del sistema. Cada evento es diferente por naturaleza, algunos requieren que el programa recobre y muestre cierta información, y otros que se inicien algunos cálculos y cambios de estado.
- Controladores de eventos: Es una determinada unidad del programa que se activa para reaccionar ante un evento. Es decir, es un tipo de función o método que ejecuta una acción específica cuando se activa un evento determinado.
- Tiempo controlado: Es un código específico que se ejecuta en un momento determinado. Esto significa que es una tarea preestablecida por hacer. La actualización de Windows es un ejemplo de tiempo controlado, donde el usuario

puede establecer cuándo actualizar o cuándo verificar y descargar la actualización.

- II. Utiliza lenguaje de programación orientada al evento, en la resolución de problemas.

Existen varios lenguajes de programación que admiten la programación orientada al evento, y su elección depende de los requerimientos y las preferencias del proyecto. Algunos de ellos son:

- Java: Es un lenguaje de programación ampliamente utilizado en el desarrollo de aplicaciones empresariales y aplicaciones móviles. A través de bibliotecas como Swing y JavaFX, se puede implementar la programación orientada al evento para responder a eventos de interfaz de usuario.
- Python: Aunque Python no es estrictamente un lenguaje de programación orientado al evento, se pueden utilizar bibliotecas como PyQt o Tkinter para desarrollar aplicaciones con programación orientada al evento, como interfaces gráficas interactivas.
- JavaScript: Es un lenguaje de programación utilizado principalmente en el desarrollo web. Es ampliamente utilizado para la programación orientada al evento en el lado del cliente, donde se puede responder a eventos como clics de botones, envío de formularios y cambios en el contenido de la página.
- C#: Es un lenguaje de programación desarrollado por Microsoft y utilizado en el desarrollo de aplicaciones de Windows y aplicaciones web basadas en ASP.NET. Permite una programación orientada al evento mediante el uso de controladores de eventos y delegados.

3.3.8 Taller de Sistemas de Información I

La asignatura Taller de Sistemas de Información, permite poner en práctica los conocimientos adquiridos en el plan de estudios hasta el momento, desarrollando las etapas de análisis y diseño de un sistema de información.

Los contenidos a tratar en la asignatura son:

1. Etapas de Análisis para un sistema de información en particular.
2. Etapas de Diseño para un sistema de Información en particular.

Los resultados de aprendizaje que se esperan lograr son:

- I. Aplica metodología para definir requerimiento de un sistema de información según necesidades del usuario.

La definición de los requerimientos de un sistema de información es un paso crítico en su desarrollo, y se requiere una metodología estructurada para garantizar que los requerimientos se comprendan y documenten correctamente. Esta metodología permite proyectar las estrategias, procesos y flujos de datos de la empresa al igual que las interrelaciones entre procesos y datos, con el fin de desarrollar un plan de sistema de información capaz de guiar el desarrollo de un sistema que permita dar soporte al área en estudio en el cumplimiento de sus objetivos. Pasos claves para definir los requerimientos: (Londoño, n.d.)

- **Análisis:** Se deben identificar y entrevistar a todas las partes interesadas en el sistema de información. Esto ayuda a comprender las necesidades y expectativas de cada grupo. El objetivo de esta fase es desarrollar el diseño de arquitectura en los sistemas. En el diseño de arquitectura, se engloban dos componentes: los datos y los procesos, los cuales serán analizados y diseñados desde una perspectiva conceptual a una física. (Londoño, n.d.)
- **Definición de los requisitos funcionales:** Se establecen los requisitos específicos del sistema, es decir, las funciones y operaciones que debe realizar para cumplir con las necesidades del usuario. (DECSAI, 2019)
- **Definición de los requisitos no funcionales:** Se identifican y especifican los requisitos no relacionados con las funciones directas del sistema, como requisitos de rendimiento, seguridad, usabilidad, compatibilidad, mantenibilidad, entre otros. (DECSAI, 2019)

- II. Desarrolla actividades de análisis de un sistema de información, según requerimientos de usuarios.

El análisis de un sistema de información implica comprender en profundidad los requerimientos de los usuarios. Tales como:

- **Validación de requerimientos:** Es un proceso continuo en el proyecto de desarrollo de software con el fin de asegurar que los requerimientos licitados sean representaciones exactas de las necesidades y expectativas de los usuarios. Esta actividad contribuye a mejorar la calidad de los requerimientos, a reducir costos, tiempos y riesgos en el desarrollo de software. (Santana, 2020)
- **Descomposición funcional:** se refiere al proceso de identificar y resolver las relaciones funcionales en sus partes constituyentes, de tal forma que la función

global pueda ser reconstruida a partir de sus partes. Por lo general, la descomposición funcional se realiza para identificar y entender los componentes o partes que constituyen un todo (o función global). Consiste en tomar los requerimientos de software, dividirlos en partes y analizarlos individualmente. De ser necesario, se pueden descomponer en más partes hasta lograr un nivel adecuado de detalle. (pmoinformatica, pmoinformatica, 2016)

- Modelado de datos: Es el proceso de creación de una representación visual o esquema que define los sistemas de recopilación y administración de información de cualquier organización. Este esquema o modelo de datos ayuda a las diferentes partes interesadas, como analistas de datos, científicos e ingenieros, a crear una vista unificada de los datos de una organización. El modelo esboza los datos que recoge la empresa, la relación entre los distintos conjuntos de datos y los métodos que se usarán para almacenarlos y analizarlos. Esto puede incluir la creación de diagramas de entidad-relación (ER) o modelos conceptuales de datos. (modelado, n.d.)

III. Desarrolla actividades de diseño de un sistema de información, según requerimientos de usuarios.

El diseño de un sistema de información implica la creación de una solución técnica que cumpla con los requerimientos identificados durante la fase de análisis. Algunas fases son:

- Diseño de arquitectura: Utiliza los conocimientos de programación para planear el diseño general del software de modo que puedan agregarse detalles más adelante, lo cual permite a los equipos de software delimitar el panorama general y comenzar a elaborar un prototipo. Se define la estructura general del sistema, incluyendo los componentes, módulos y su interconexión. Luego se selecciona la arquitectura adecuada que permita cumplir con los requerimientos funcionales y no funcionales. (lucidchart, lucidchart, n.d.)
- Diseño de interfaz de usuario: Es el proceso que utilizan los diseñadores para crear interfaces en software o dispositivos computarizados, centrándose en la apariencia o el estilo. Los diseñadores tienen como objetivo crear interfaces que los usuarios encuentren fáciles de usar y agradables. El diseño de la interfaz de usuario se refiere a interfaces gráficas de usuario y otras formas, por ejemplo, interfaces controladas por voz. (Debernardi, 2021)
- Diseño de base de datos: Es un proceso fundamental a la hora de modelar nuestros conjuntos de datos y definir las operaciones que queremos realizar sobre ellos. Los datos son el activo más importante de nuestra organización y

una base de datos bien diseñada influye de forma directa en la eficiencia que obtendremos a la hora de almacenar, recuperar y analizar nuestros datos. Además, se utilizan herramientas de modelado de datos para crear diagramas de entidad-relación o modelos de bases de datos. (Carisio, n.d.)

3.3.9 Programación Orientada a Objeto

Esta asignatura entrega los fundamentos de la programación orientada a objetos para ser aplicados en la implementación de programas basados en requerimientos de usuario.

Los contenidos a tratar en la asignatura son:

1. Fundamentos de programación orientada a objetos
 - Conceptos relacionados al paradigma orientado a objetos.
 - Diferencia entre programa tradicional y orientado al objeto.
2. Estudio de uno o más lenguajes de programación orientado a objetos.
 - Entorno de desarrollo.
 - Aplicación de los conceptos de orientación a objeto según lenguaje en estudio.
 - Diseño de clases (abstracción y modularidad)
 - Definición campos y parámetros con tipos de datos simples y complejos (tipo clase).
 - Implementación de comportamiento (métodos de creación, modificación, acceso y gestión de colecciones)
 - Diseño de interfaz de usuario.
 - Generación de informes.
 - Manejo de archivos y conexión a bases de datos.
3. Proceso de diseño de programas orientados a objetos.
4. Tópicos de Especialidad.

Los resultados de aprendizaje que se esperan lograr son:

- I. Aplica conceptos de programación orientada a objetos según requerimientos planteados.

La Programación Orientada a Objetos (POO) es un paradigma de programación, es decir, un modelo o un estilo de programación que nos da unas guías sobre cómo trabajar con él. Se basa en el concepto de clases y objetos. Este tipo de programación se utiliza para estructurar un programa de software en piezas simples y reutilizables de planos de código (clases) para crear instancias individuales de objetos. Al aplicar la POO a los requerimientos planteados, se pueden seguir los siguientes conceptos: (Canelo, 2020)

- Clases: Una clase es una especie de "plantilla" en la que se definen los atributos y métodos predeterminados de un tipo de objeto. Esta plantilla se crea para poder crear objetos fácilmente. Al método de crear nuevos objetos mediante la lectura y recuperación de los atributos y métodos de una clase se le conoce como instanciación. (Lewis & Loftus, 2019)
 - Objetos: Instancia de una clase. Entidad provista de un conjunto de propiedades o atributos (datos) y de comportamiento o funcionalidad (métodos), los mismos que consecuentemente reaccionan a eventos. Se corresponden con los objetos reales del mundo que nos rodea, o con objetos internos del sistema (del programa). (Lewis & Loftus, 2019)
 - Encapsulamiento: Contiene toda la información importante de un objeto dentro del mismo y solo expone la información seleccionada al mundo exterior. Esta propiedad permite asegurar que la información de un objeto esté oculta para el mundo exterior, agrupando en una Clase las características o atributos que cuentan con un acceso privado, y los comportamientos o métodos que presentan un acceso público. Se deben definir los niveles de acceso adecuados para los atributos y métodos de las clases, asegurando que solo se pueda acceder y modificar lo necesario. (Canelo, 2020)
 - Herencia: Define relaciones jerárquicas entre clases, de forma que atributos y métodos comunes puedan ser reutilizados. Las clases principales extienden atributos y comportamientos a las clases secundarias. A través de la definición en una clase de los atributos y comportamientos básicos, se pueden crear clases secundarias, ampliando así la funcionalidad de la clase principal y agregando atributos y comportamientos adicionales. (Canelo, 2020)
 - Polimorfismo: Consiste en diseñar objetos para compartir comportamientos, lo que nos permite procesar objetos de diferentes maneras. Es la capacidad de presentar la misma interfaz para diferentes formas subyacentes o tipos de datos. Al utilizar la herencia, los objetos pueden anular los comportamientos principales compartidos, con comportamientos secundarios específicos. El polimorfismo permite que el mismo método ejecute diferentes comportamientos de dos formas: anulación de método y sobrecarga de método. (Canelo, 2020)
- II. Utiliza lenguaje de programación orientada a objetos en la resolución de problemas. (Velázquez, 2016)

Para resolver problemas utilizando un lenguaje de programación orientada a objetos, se deben seguir los siguientes pasos:

- Identificar los objetos relevantes: Analiza el problema y determina qué objetos son necesarios para representar las entidades y conceptos involucrados en el mismo. Estos objetos deben tener atributos y comportamientos adecuados para resolver el problema.
- Definir las clases: Crea las clases necesarias en el lenguaje de programación, definiendo sus atributos y métodos. Estas clases deben reflejar los objetos identificados en el paso anterior.
- Crear instancias de objetos: Crea instancias de las clases utilizando la sintaxis del lenguaje de programación. Asigna valores a los atributos de los objetos según los datos del problema.
- Aplicar métodos y operaciones: Utiliza los métodos definidos en las clases para realizar las operaciones necesarias para resolver el problema. Manipula los objetos y sus atributos de acuerdo con los requerimientos planteados.
- Realizar pruebas y depuración: Ejecuta el programa y realiza pruebas para verificar que la solución propuesta funciona correctamente. Identifica y corrige posibles errores o fallas en el código.

3.3.10 Diseño y Programación Orientada a la Web

Esta asignatura entrega los conocimientos fundamentales para planificar, maquetar y diseñar portales web aplicando técnicas de diseño, utilizando “ambientes de desarrollo integrado” (IDE) de vanguardia y complementariamente software de producción gráfica.

Los contenidos a tratar en la asignatura son:

1. Internet.
 - Antecedentes Históricos.
 - Evolución y Desarrollo.
 - Conceptos asociados a Internet.
 - Caracterización de internet y la web en base a beneficios, arquitectura y funcionalidades.
 - Aspectos fundamentales de análisis y diseños de aplicaciones en internet.
2. Ambiente Web/ Entorno Web.
 - Características y atributos de los sistemas y aplicaciones basados en web.
 - Estratos/Capas de la Ingeniería Web:
 - Procesos.
 - Métodos.
 - Herramientas.

- Tecnologías.
- 3. Estudio y utilización de las herramientas y lenguajes de programación web.
 - Conceptos básicos.
 - Estructuras de control.
 - Estructuras de datos.
 - Manejo de Base de Datos, Utilización de Framework.
 - Otros elementos y características propias de el(los) lenguaje(s) en estudio.
(Diseño responsivo)

Los resultados de aprendizaje que se esperan lograr son:

- I. Define entorno de desarrollo de programas en ambiente web, determinando los elementos necesarios para su implementación.

El entorno de desarrollo de programas en ambiente web se refiere al conjunto de herramientas y tecnologías utilizadas para crear aplicaciones web. Algunos elementos necesarios para su implementación son:

- Lenguajes de marcado: Para dar formato o estructura a una página, el lenguaje de marcas utiliza una serie de etiquetas (a menudo entre paréntesis angulares) cuyo símbolo es “< >” y que funcionan como instrucciones que, al estar contenidas entre “<>”, se diferencian sintácticamente del texto sin formato. Las etiquetas son leídas e interpretadas por parsers o analizadores que luego generan una salida, generalmente como una página o un documento que tiene el formato y el estilo correspondiente y manteniendo la marca real oculta a simple vista. (seobility, n.d.)
- Servidores web: Un servidor web (server) es un ordenador de gran potencia que se encarga de “prestar el servicio” de transmitir la información pedida por sus clientes (otros ordenadores, dispositivos móviles, impresoras, personas, etc.). Los servidores web (web server) son un componente de los servidores que tienen como principal función almacenar, en web hosting, todos los archivos propios de una página web (imágenes, textos, videos, etc.) y transmitirlos a los usuarios a través de los navegadores mediante el protocolo HTTP (Hypertext Transfer Protocol). Los servidores web, como Apache, Nginx o Microsoft IIS, son responsables de recibir las solicitudes de los clientes y enviar las respuestas correspondientes. (Souza, 2019)
- Bases de datos: Una base de datos de un sitio web es un sistema de almacenamiento de información que permite el acceso a través de un sitio web.

La base de datos almacena información, como nombres de usuario y contraseñas, para crear una mejor experiencia de usuario. Algunas opciones comunes son MySQL, PostgreSQL y MongoDB. Las bases de datos permiten gestionar la información de manera eficiente y segura.

- Frameworks: Es un conjunto de herramientas, bibliotecas y convenciones que proporcionan una estructura para facilitar el desarrollo de software. Ofrece una base sólida y reutilizable que permite a los desarrolladores crear aplicaciones más rápido al proporcionar estructuras, patrones y componentes comunes. Estas herramientas agilizan el proceso de desarrollo y ayudan a mantener un código organizado y eficiente. (Souza B. , 2019)
- Control de versiones: Es un software que ayuda a realizar un seguimiento de los cambios realizados en el código a lo largo del tiempo. A medida que un desarrollador edita el código, el sistema de control de versiones toma una instantánea de los archivos. Después, guarda esa instantánea de forma permanente para que se pueda recuperar más adelante si es necesario. (mijacobs, 2023)
- Herramientas de depuración y pruebas: La depuración y las pruebas son procesos complementarios que garantizan que los programas de software funcionen como deben. Después de escribir una sección completa o parte de un código, los programadores realizan pruebas para identificar fallos y errores. Una vez encontrados los errores, los codificadores pueden comenzar el proceso de depuración y trabajar para librar al software de cualquier error. (amazon, n.d.)

- II. Aplica patrón de diseño modelo- vista-controlador (MVC), en el desarrollo del sitio web. (Pantoja, 2004)

El patrón de diseño Modelo-Vista-Controlador (MVC) es ampliamente utilizado en el desarrollo de aplicaciones web. Se divide en tres componentes principales:

- Modelo (*Model*): es un conjunto de clases que representan la información del mundo real que el sistema debe procesar, el cual representa los datos y la lógica de negocio de la aplicación. Aquí se definen las estructuras de datos, las operaciones y la interacción con la base de datos.
- Vista (*View*): Es un conjunto de clases que se encargan de mostrar al usuario la información contenida en el modelo. Una vista está asociada a un modelo, pudiendo existir varias vistas asociadas al mismo modelo, es responsable de la presentación y visualización de los datos al usuario.

- Controlador (*Controller*): Es un objeto que se encarga de dirigir el flujo del control de la aplicación debido a mensajes externos, como datos introducidos por el usuario u opciones del menú seleccionadas por él. A partir de estos mensajes, el controlador se encarga de modificar el modelo o de abrir y cerrar vistas. El controlador tiene acceso al modelo y a las vistas, pero las vistas y el modelo no conocen de la existencia del controlador.

III. Implementa sistemas de información utilizando herramientas basadas en programación web.

Para implementar sistemas de información utilizando herramientas basadas en programación web, existen tres grandes modelos:

- Secuencial lineal:
 - Cascada: Este modelo contiene múltiples etapas que deben ser seguidas sucesivamente, sin embargo, entre una y otra se deberá tomar un descanso usado para corroborar la reacción del lado del cliente, para poder desarrollar una etapa debe concluirse la anterior a esta. Estas son sus etapas:
 - Especificación de requisitos.
 - Diseño del software.
 - Construcción o Implementación del software.
 - Integración.
 - Pruebas (o validación).
 - Despliegue (o instalación).
 - Mantenimiento.
 - RAD (Modelo de Desarrollo Rápido de Aplicación): Como su nombre lo dice se trata de un modelo que cuyos tiempos para su desarrollo son cortos, se divide en equipos y de esa forma cada uno dirigirá una parte del sistema por lo tanto se requiere de múltiples diseñadores, usabilidad, la rapidez y su uso son las características que este modelo desarrolla.
- Evolutivo:
 - Incremental: Este modelo lleva un proceso paso por paso sin embargo en algunas ocasiones algunas etapas pueden ser pospuestas (contrario al modelo en cascada).
 - Espiral: Se ordenan con cuidado las etapas del desarrollo de sistemas, se desarrolla en ciclos donde en cada ciclo se define el objetivo, se toman

en cuenta los posibles riesgos, se desarrollan posibles soluciones y se ponen a prueba, los resultados se revisan y de esa forma se planifica el ciclo que viene. Este modelo es ideal para aquellos productos que tengan un alto nivel de inestabilidad.

- Basado en reutilización: Se incrementa la fiabilidad, reduciendo así el riesgo de errores. Lo que busca es armar un nuevo proyecto a partir de componentes ya existentes. La desventaja que conlleva es que algunos de los requisitos que sean requeridos no podrán ser cubiertos si no adaptados de cierto modo para que se le acerque a lo deseado.
- Basado en transformaciones:
 - Basado en lógica y matemáticas a través de técnicas y herramientas, un modelo bastante formal, cuyas especificaciones son refinadas hasta que el programa es alcanzado. Las refinaciones formales son descripciones concisas de las propiedades de un sistema que es basado principalmente en las matemáticas. Este modelo tiene sus desventajas como el costo, que suele ser elevado, requiere de mucho tiempo y de expertos en el área. (Ceupe, 2021)

IV. Utiliza *Search Engine Optimization* para optimizar la búsqueda en los portales web.

Para optimizar la búsqueda en los portales web, se pueden aplicar las siguientes técnicas de SEO: (Mousinho, 2020)

- Investigación de palabras clave: Realiza una investigación exhaustiva de palabras clave relevantes para tu portal web.
- Optimización del contenido: Crea contenido de alta calidad y relevante, utilizando palabras clave de manera natural en títulos, encabezados, texto del cuerpo y meta descripciones.
- Estructura del sitio web: Diseña una estructura clara y fácil de navegar, con una arquitectura de enlace interno que permita a los motores de búsqueda indexar y rastrear todas las páginas importantes.
- Optimización técnica: Asegúrate de que tu portal web se carga rápidamente, optimiza tus imágenes y asegúrate de que el sitio sea compatible con dispositivos móviles.
- Construcción de enlaces: Obtén enlaces de calidad desde sitios relevantes y de confianza, a través de la creación de contenido valioso o la colaboración con otros sitios web.

- Optimización de etiquetas y meta descripciones: Utiliza etiquetas de título y meta descripciones descriptivas y persuasivas, que contengan palabras clave relevantes.
- Monitoreo y análisis: Utiliza herramientas de análisis web para monitorear el rendimiento de tu portal y realizar ajustes y optimizaciones continuas.

3.3.11 Bases de Datos

La asignatura entrega conocimientos fundamentales de los aspectos involucrados en el enfoque de bases de datos, modelamiento de datos, metodologías de diseño de bases de datos, lenguaje no procedural y herramientas de desarrollo, necesarios para crear y manipular bases de datos. A su vez, trata otros tópicos misceláneos de la especialidad.

Los contenidos a tratar en la asignatura son:

1. Conceptos y enfoque de Bases de Datos.
 - Introducción a las bases de datos.
 - Sistemas de bases de datos v/s sistemas de administración de archivos.
 - Organización general de un sistema administrador de bases de datos.
 - Distintos modelos de una base de datos. Esquemas y sub-esquemas.
2. Diseño de Bases de Datos.
 - Análisis de requerimientos.
 - El proceso de diseño de datos: modelo externo, conceptual, lógico y físico.
3. Modelamiento de datos.
 - Objetivos del modelamiento de datos.
 - Tipos de modelo de datos: modelos semánticos y modelos clásicos.
 - Modelo Entidad-Relación y Entidad-Relación-Extendido.
 - Modelo relacional: propiedades estructurales, lenguajes de manipulación (álgebra relacional, cálculo relacional y estándar SQL), formas normales y reglas de transformación del modelo conceptual al relacional.
4. Sistema administrador de bases de datos relacionales (SABDR).
 - Componentes de un SABDR.
 - Definición, manipulación y control de datos.
 - Reglas de integridad.
 - Lenguaje no Procedural (SQL).

Los resultados de aprendizaje que se esperan lograr son:

- I. Analiza el rol de las bases de datos (BD) y de los sistemas administradores de bases de datos (SABD) en las organizaciones, conceptualizando sus principales características.

Las bases de datos (BD) y los sistemas administradores de bases de datos (SABD) desempeñan un papel fundamental en las organizaciones en la gestión eficiente y segura de la información.

- Bases de datos (BD): (ayudaley, n.d.)
 - Los datos almacenados tienen independencia física y lógica.
 - Garantizan la integridad de los datos.
 - Son sistema de almacenamiento que contribuyen a reducir la redundancia al mínimo.
 - Es habitual que puedan acceder a ellas múltiples usuarios.
 - Permiten realizar consultas complejas de diferentes tipos de datos.
 - El acceso a los datos es seguro.
 - Se pueden hacer copias de seguridad y recuperar los datos.
 - Se accede a la información a través de un lenguaje concreto.
- Sistemas administradores de bases de datos (SABD): (Alexander-cidecmerida, 2010)
 - Ofrecen una interfaz para administrar y controlar las bases de datos.
 - Proporcionan herramientas para crear, modificar y eliminar estructuras de datos, como tablas, índices y restricciones.
 - Permiten la definición de consultas y la ejecución de operaciones para acceder y manipular los datos.
 - Garantizan la seguridad de los datos, mediante la gestión de usuarios y permisos.
 - Controlan la concurrencia y el acceso concurrente a los datos para evitar conflictos.
 - Realizan tareas de respaldo y recuperación para proteger los datos contra pérdidas o fallas del sistema.

- II. Analiza el rol de los modelos de datos (MD) y de las bases de datos en el análisis y diseño de sistemas de información (SI), reconociendo su aporte en los procesos organizacionales.

Los modelos de datos y las bases de datos juegan un papel fundamental en el análisis y diseño de sistemas de información, ya que permiten representar y estructurar la

información de manera efectiva. Un modelo de datos completo y bien pensado es clave para el desarrollo de una base de datos verdaderamente funcional, útil, segura y precisa. El modelo de datos lógico dirige el modelo físico que es específico para una base de datos, y es el documento de diseño detallado que guía la creación de la base de datos y el software de la aplicación.

Un buen modelado de datos y diseño de la base de datos son esenciales para desarrollar aplicaciones funcionales, confiables y seguras, y bases de datos que funcionan bien con almacenes de datos y herramientas analíticas. (SAP, n.d.)

- III. Identifica características esenciales de los modelos de datos conceptuales, definiendo notaciones estándar de representación (como ER, ERE, entre otras).

Los modelos de datos conceptuales representan las entidades, atributos y relaciones de alto nivel en un sistema de información, sin preocuparse por los detalles de implementación. Tales como:

- Enfoque conceptual: Se trata de un método que permite analizar y comprender el funcionamiento de los sistemas complejos. Este enfoque se centra en el estudio de los conceptos y de las relaciones entre ellos. Los modelos de datos conceptuales se centran en la representación abstracta y lógica de las entidades y sus relaciones, sin considerar cómo se implementarán físicamente. (nunezmarlyn, 2020)
- Notación ER (Entidad-Relación): El modelo Entidad-Relación es una notación estándar ampliamente utilizada para representar modelos de datos conceptuales. Muestra cómo interactúan las entidades (personas, objetos y conceptos). Estos modelos conceptuales de datos ayudan a desarrolladores y diseñadores a visualizar las relaciones entre elementos clave del software. (Miro, n.d.)
- Notación ERE (Entidad-Relación Extendido): Son diagramas de bases de datos avanzados muy similares a los diagramas ER normales. Los diagramas ER mejorados son modelos de alto nivel que representan los requerimientos y complejidades de bases de datos complejas. Agrega elementos como relaciones recursivas, atributos compuestos, herencia y especialización. (Lucidchart, n.d.)

- IV. Diseña modelos de datos relacionales, aplicando normalización o mapeando desde el modelo de entidad-relación.

El diseño de modelos de datos relacionales implica la creación de esquemas de base de datos que sigan los principios de normalización y mapeo desde el modelo entidad-relación. Tales como:

- Normalización: Es la transformación de las vistas de usuario complejas y del almacén de datos a un juego de estructuras de datos más pequeñas y estables. La normalización de bases de datos es un proceso que consiste en designar y aplicar una serie de reglas a las relaciones obtenidas tras el paso del modelo entidad-relación al modelo relacional. Con objeto de minimizar la redundancia de datos, facilitando su gestión posterior. Esto se logra aplicando las formas normales, como la primera forma normal (1NF), la segunda forma normal (2NF) y la tercera forma normal (3NF), entre otras. (bookdown, 2006)
- Mapeo desde el modelo entidad-relación: Es una técnica de programación que permite convertir datos del sistema de tipos utilizado por los lenguajes de programación orientado a objetos al utilizado por las bases de datos relacionales. Cada entidad se convierte en una tabla, los atributos se convierten en columnas y las relaciones se representan mediante claves foráneas que establecen vínculos entre las tablas. (Yañez, 2018)
- Definición de claves primarias y foráneas: Se identifican las claves primarias que identifican de forma única a cada registro en una tabla y se definen las claves foráneas para establecer relaciones entre tablas. (qgis, 2018)

V. Desarrolla bases de datos, utilizando sistemas administradores de BD relacionales (SABDR).

Desarrollar bases de datos utilizando sistemas administradores de bases de datos relacionales (SABDR) implica seguir un proceso que incluye la definición de requisitos, el diseño del esquema, la implementación de la lógica de negocio, la optimización del rendimiento, la seguridad y la gestión de usuarios, el respaldo y la recuperación, y el mantenimiento continuo. Al seguir estos pasos, se pueden crear bases de datos eficientes y funcionales. Es importante tener en cuenta las características específicas del SABDR utilizado. (stibosystems, 2020)

3.3.12 Arquitectura y Organización de Computadores

La asignatura entrega los conceptos fundamentales que permiten comprender la organización de los computadores modernos y acercarse a una disciplina que está en constante cambio y que es básica para crear sistemas de software eficientes. Además,

muestra la interdependencia entre los programas computacionales y la arquitectura de hardware que lo sustenta.

Los contenidos a tratar en la asignatura son:

- Sistemas combinacionales.
- Sistemas secuenciales.
- Introducción al diseño asistido por computador (*Computer-Aided Design, CAD*) y lenguajes de descripción de hardware.
- Componentes digitales estructurados.
- Arquitectura de un computador y lenguaje de máquina.
- Microarquitectura.

Los resultados de aprendizaje que se esperan lograr son:

- I. Diseña sistemas lógicos simples, utilizando componentes digitales estructurados.

El diseño lógico de un sistema se refiere a una representación abstracta del flujo de datos, entradas y salidas del sistema. Esto se lleva a cabo a menudo a través de la modelización, utilizando un modelo muy abstracto (y a veces gráfico) del sistema real. En el contexto de los sistemas, los diseños son incluidos. El diseño lógico incluye diagramas de entidad-relación. Además, se pueden crear circuitos digitales utilizando componentes estructurados como compuertas lógicas, multiplexores, decodificadores, registros y flip-flops. (Bentley, 2004)

- II. Explica el funcionamiento lógico de las unidades que componen un computador digital, describiendo sus salidas frente a diferentes entradas (causa-efecto).

Las unidades principales que componen un computador digital incluyen la unidad central de procesamiento (CPU), la unidad de control y la unidad aritmético-lógica (ALU). Aquí hay una descripción general del funcionamiento lógico y las salidas de estas unidades frente a diferentes entradas:

- Unidad de Control: Es la encargada de la administración de todos los recursos del computador. Esta unidad se encarga del traslado de los datos y las instrucciones entre las partes que los requieran; interpreta los comandos y produce las órdenes para efectuar cada una de las tareas. Vigila las secuencias y sincroniza la operación de cada una de las partes que intervienen en la ejecución

de un programa. Su salida incluye señales de control que activan diferentes componentes del sistema, como el registro de instrucciones, el contador de programa y las unidades de ejecución. (aulafacil, 2018)

- Unidad Aritmético-Lógica (ALU): Es un circuito digital que calcula operaciones aritméticas (como adición, substracción, etc.) y operaciones lógicas (como OR, NOT, XOR, etc.), entre dos números. Su salida depende de las entradas que recibe, como los operandos y las señales de control. (alumdaci, 2015)
- Unidad Central de Procesamiento (CPU): es el componente en un ordenador, que interpreta las instrucciones y procesa los datos contenidos en los programas de la computadora. Las CPU proporcionan la característica fundamental de la computadora digital (la programabilidad) y son uno de los componentes necesarios encontrados en las computadoras de cualquier tiempo, junto con el almacenamiento primario y los dispositivos de Entrada/salida. Se conoce como Microprocesador el CPU que es manufacturado con circuitos integrados.

III. Crea programas simples, utilizando lenguaje de máquina y lenguaje *Assembly*.

La creación de programas simples implica escribir instrucciones utilizando lenguaje de máquina o lenguaje *Assembly*. Algunos pasos a seguir son: (uoc, 2017)

- Entender la arquitectura y el conjunto de instrucciones de la CPU.
- Escribir instrucciones utilizando el lenguaje de máquina o el lenguaje *Assembly*.
- Utilizar las instrucciones de entrada/salida para interactuar con los dispositivos periféricos, de ser necesario.
- Realizar pruebas y depurar el programa para asegurarse de que funcione correctamente.

IV. Analiza el desempeño de una *CPU*, monitoreando los principales factores que inciden en distintas configuraciones.

El análisis del desempeño de una CPU implica el monitoreo de varios factores clave que pueden afectar su rendimiento en diferentes configuraciones. Algunos factores a considerar son: (*Assembly*, n.d.)

- Frecuencia de reloj: es una de las especificaciones más importantes de un procesador de computadora. Se refiere al número de ciclos por segundo que la CPU puede ejecutar. Cuanto mayor sea la velocidad del reloj, más rápido puede el procesador realizar tareas. Esta velocidad se mide en Gigahertz (GHz), y es un

factor crucial para determinar el rendimiento general de una computadora. (CHUWI, 2021)

- Arquitectura de la CPU: Es básicamente la estructura interna que tiene este elemento. En la arquitectura de la CPU no hablamos de la forma y tamaño, sino de cómo están ubicadas las distintas unidades lógicas y físicas que componen un procesador, hablamos de la ALU, registros, Unidad de Control, etc. En este sentido, actualmente existen dos tipos de arquitectura la CISC y la RISC. Una mayor cantidad de núcleos permite una mayor capacidad de procesamiento simultáneo, y una caché más grande puede reducir los tiempos de acceso a la memoria.

3.3.13 Taller de Sistemas de Información II

En esta asignatura se espera que el estudiante sea capaz de integrar todos los conocimientos adquiridos y habilidades desarrolladas durante su proceso formativo, desarrollando un Sistema de Información.

Los contenidos a tratar en la asignatura son:

1. Programación de Proyecto Informático.

Los resultados de aprendizaje que se esperan lograr son:

- I. Realiza pruebas de cumplimiento de requerimientos, resistencia, recuperación y seguridad para asegurar la funcionalidad del software.

Para asegurar la funcionalidad del software, es necesario realizar pruebas, tales como:

- Pruebas de cumplimiento de requerimientos: Consisten en recolectar evidencia con el propósito de probar el cumplimiento de una organización con procedimientos de control. Esto difiere de la prueba sustantiva, en la que la evidencia se recoge para evaluar la integridad de transacciones individuales, datos u otra información. (ClasicWebTools, 2020)
- Pruebas de resistencia: Las pruebas de resistencia se refieren a las pruebas que generalmente se realizan para averiguar si una aplicación puede soportar la carga de procesamiento que se espera que tenga que soportar durante un período prolongado. Durante las pruebas de resistencia, se observa el consumo de memoria para determinar posibles fallas. La calidad del rendimiento a veces

también se controla durante las pruebas de resistencia. (theastrology, theastrology, 2022)

- Pruebas de recuperación: La prueba de recuperación ante desastres es un simulacro de varios pasos del plan de recuperación ante desastres (DRP) de una organización diseñado para garantizar que los sistemas de tecnología de la información (TI) se restablecerán si ocurre un desastre real. (Ortiz, 2020)
- Pruebas de seguridad: El propósito principal de las pruebas de seguridad es detectar vulnerabilidades y luego repararlas. Ayuda a impulsar el sistema actual y asegurarse de que el sistema pueda funcionar durante un tiempo prolongado. Para notar lagunas que provocarán la pérdida de información vital. (joshi, 2020)

- II. Corrige programación, considerando resultados de pruebas que permitan asegurar funcionalidad. (unrein, 2018)(unrein, 2018)

Después de realizar pruebas y obtener resultados, es importante corregir cualquier problema o error identificado en el software. Una vez analizados los resultados de las pruebas y de registrar los problemas encontrados (errores, fallos de rendimiento o vulnerabilidades de seguridad identificadas). Se priorizan los problemas para identificar que correcciones serán necesarias y así realizar las modificaciones en código fuente o configuración del software. Luego se realizarán pruebas adicionales para verificar que se hayan aplicado de manera correcta las modificaciones. Finalmente, se debe documentar que correcciones se realizaron en el software.

- III. Implementa sistemas de información para su explotación final, asegurando funcionalidad.

La implementación de sistemas de información se realiza para poner en funcionamiento el software de manera adecuada. Los pasos para seguir una implantación de un nuevo sistema de información podríamos clasificarlos en: Planificación, Adquisición, Introducción, Utilización y Retirada. Finalmente, con esto logramos garantizar que el software funcione correctamente y cumpla con los requerimientos definidos antes de su puesta en producción, y así conseguir el adecuado funcionamiento para los usuarios. (Pérez, 2022)

3.3.14 Desarrollo de Aplicaciones Móviles

La asignatura entrega los conocimientos fundamentales para el diseño y programación de aplicaciones destinados a dispositivos móviles.

Los contenidos a tratar en la asignatura son:

1. Introducción a la Tecnología de Dispositivos Móviles.
 - Metodologías de Desarrollo.
2. Entornos de Tecnologías Móviles.
 - Android.
 - IOS.
 - WINDOWS MOBILE.
3. Desarrollo de Aplicaciones.
 - Entorno de trabajo.
 - IDE.
 - Diseño de Interfaz.
 - Gestión del Dispositivo.
 - Gestión de Datos.
 - Distribución de aplicaciones.
 - Elementos de desarrollo.
4. ACTIVIDADES PRÁCTICAS

Los resultados de aprendizaje que se esperan lograr son:

- I. Aplica conceptos de programación para dispositivos móviles, según requerimientos definidos.

La definición de desarrollo de aplicaciones móviles puede ser amplia, ya que abarca todos los procesos de creación y lanzamiento de una aplicación móvil, o específica, ya que se refiere a la codificación y la implementación de la aplicación. Este artículo cubrirá ambas definiciones de desarrollo de aplicaciones móviles y se centrará especialmente en la definición de desarrollo de aplicaciones móviles para las personas que toman decisiones o son responsables de cómo se crea el propio software. Las fases del proceso de desarrollo de aplicaciones móviles son:

- Estrategia: El proceso de desarrollo de aplicaciones móviles comienza por entender por qué una aplicación móvil puede ser una buena solución a un problema o un enfoque a una oportunidad, y luego se decide qué construir y cómo. En la fase de estrategia, las organizaciones establecerán los objetivos de la aplicación y las medidas de éxito, analizarán el mercado y la competencia, investigarán las necesidades de sus clientes y determinarán a qué usuarios y casos de uso servirá la aplicación.

- **Planificación:** Las organizaciones acuerdan cuatro detalles clave del proceso de desarrollo de aplicaciones móviles: los equipos, las herramientas, las tecnologías y los plazos. En este punto, ya tienen una idea del tipo de aplicación móvil que necesitan crear, quiénes la crearán y qué plataforma o plataformas utilizarán.
- **Diseño:** La apariencia, las características y la función de la aplicación móvil comienzan a tomar forma durante la fase de diseño. En esta fase del proceso de desarrollo de la aplicación móvil, los equipos trazarán la experiencia del usuario y crearán las interfaces de usuario y los estados de la aplicación en varias fidelidades. Pueden empezar con maquetas de baja fidelidad para proporcionar una prueba de concepto o recoger los comentarios de los clientes en una fase temprana. Los desarrolladores e ingenieros pueden ayudar a crear prototipos o consultar sobre las decisiones de experiencia de usuario y de interfaz de usuario en esta fase, manteniendo a los equipos al día de los requisitos y las especificaciones técnicas de las plataformas que la organización ha decidido utilizar.
- **Desarrollo:** En este contexto, la palabra desarrollo se refiere a veces al ciclo de vida completo del desarrollo de software que implica la creación y el lanzamiento de una aplicación móvil. Sin embargo, lo más habitual es que desarrollo se refiera a esta fase de desarrollo de software de la aplicación móvil. Además, en esta fase se describe mejor como el paso en el que los equipos establecen la arquitectura técnica y la pila tecnológica, incluyendo el front-end, el back-end y las API, fijan los hitos de desarrollo, determinan el ciclo de vida del desarrollo de la aplicación y codifican la aplicación.
- **Prueba:** Las pruebas son una fase crucial del proceso de desarrollo de aplicaciones móviles. Sin las pruebas, las organizaciones no pueden estar seguras de que su aplicación será segura, funcionará de la manera que los usuarios necesitan y esperan que lo haga, y funcionará donde la implementen. Durante esta fase, los equipos se encargan de garantizar la calidad, creando casos de prueba para evaluar la usabilidad, el rendimiento, la estabilidad y la seguridad de la aplicación en todos los dispositivos en los que está previsto que se ejecute.
- **Lanzamiento:** Una vez implementada la aplicación, la organización ofrece soporte a los clientes que la utilizan y un seguimiento y mantenimiento continuos para garantizar que la aplicación esté disponible y funcione como debe. Las campañas de marketing y las acciones para fomentar la adopción de la aplicación suelen coincidir con esta fase.

- II. Aplica conceptos de diseño responsivo de aplicaciones web para dispositivos móviles, según requerimientos definidos.

El diseño responsivo de aplicaciones web es esencial para brindar una experiencia de usuario óptima en dispositivos móviles. Algunos conceptos:

- Diseño adaptable: Se trata de redimensionar y colocar los elementos de la web de forma que se adapten al ancho de cada dispositivo permitiendo una correcta visualización y una mejor experiencia de usuario. Se caracteriza porque los layouts (contenidos) e imágenes son fluidos y se usa código media-queries de CSS3. El diseño responsive permite reducir el tiempo de desarrollo, evita los contenidos duplicados, y aumenta la viralidad de los contenidos ya que permite compartirlos de una forma mucho más rápida y natural. (socialmood, 2021)
- Uso de media queries: Es una parte fundamental de CSS3 que te permite renderizar el contenido para adaptarlo a diferentes factores como el tamaño de la pantalla o la resolución. (Duò, 2022)
- Optimización de imágenes y multimedia: Se optimizan las imágenes y archivos multimedia para que se carguen rápidamente en dispositivos móviles, teniendo en cuenta las limitaciones de ancho de banda y rendimiento. Existen tres formas de optimizar imágenes web de manera rápida: Cambia el tamaño y recorta imágenes con una herramienta de edición de fotos, usa una herramienta de compresión en línea, e Instala un plugin de WordPress para optimización de imagen. (Hernandez, 2022)

- III. Desarrolla aplicaciones orientadas a dispositivos móviles, utilizando lenguaje de programación, según requerimientos definidos.

En el desarrollo de aplicaciones móviles, se utiliza un lenguaje de programación específico y se siguen los requerimientos definidos. Para desarrollar aplicaciones móviles, se utiliza un lenguaje de programación específico, como Java, Kotlin, Swift u otros, según la plataforma objetivo. Se siguen los requerimientos definidos, desarrollando la interfaz de usuario, la lógica de la aplicación y la integración de funcionalidades específicas de los dispositivos móviles, como cámara, sensores, geolocalización y notificaciones push. Se realizan pruebas y depuración para asegurar que la aplicación funcione correctamente y cumpla con los requerimientos establecidos.

3.3.15 Taller de Desarrollo de Software

La asignatura, al tener naturaleza de taller, incorpora el conocimiento de nuevas herramientas de desarrollo de software y/o de aspectos más avanzados en herramientas conocidas, en el marco de metodologías atingentes.

Los contenidos a tratar en la asignatura son:

1. Lenguaje Programación.
2. Procedimientos y funciones almacenados.
3. Construcción de aplicaciones.
4. Tópicos de Especialidad.

Los resultados de aprendizaje que se esperan lograr son:

- I. Aplica conceptos metodologías y herramientas de desarrollo de software, según requerimiento definidos.
 - Metodologías de desarrollo de software: Se aplican metodologías de desarrollo de software para gestionar el proceso de desarrollo de software de manera eficiente y efectiva. Existen dos metodologías:
 - Metodología ágil: Es una de varias metodologías de desarrollo de software basadas en el desarrollo iterativo e incremental, en contraposición a las metodologías tradicionales de desarrollo de software lineal o cascada.
 - Metodologías tradicionales: Imponen una disciplina de trabajo sobre el proceso de desarrollo del software, con el fin de conseguir un software más eficiente. Para ello, se hace énfasis en la planificación total de todo el trabajo a realizar y una vez que esta todo detallado, comienza el ciclo de desarrollo del producto software.

Dentro de ellas existen enfoques de desarrollo de software como: Cascada (Waterfall), Ágil, DevOps, entre otros. (Markus Rerych, 2007)

- Herramientas de gestión de proyectos: Las herramientas de gestión de proyectos son ayudas para facilitar que una persona o un equipo organicen eficazmente el trabajo y gestionen proyectos y tareas. El término suele referirse al software de gestión de proyectos. Se utilizan herramientas de gestión de proyectos para planificar, organizar y monitorear el desarrollo del software. Algunas herramientas conocidas son: Jira, Trello, Asana, entre otras. (Wrike, 2020)

- Control de versiones: Es la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración de este. Una versión, revisión o edición de un producto, es el estado en el que se encuentra el mismo en un momento dado de su desarrollo o modificación. Aunque un sistema de control de versiones puede realizarse de forma manual, es muy aconsejable disponer de herramientas que faciliten esta gestión dando lugar a los llamados sistemas de control de versiones o VCS (Version Control System). Algunas herramientas conocidas son: Git, Subversion (SVN), entre otras. (Collins-Sussman & Fitzpatrick, 2004)

II. Aplica lenguaje de programación para el desarrollo de software, según requerimientos definidos.

En el desarrollo de software, se utiliza un lenguaje de programación específico para implementar la lógica y la funcionalidad del software. La elección del lenguaje de programación y las herramientas a utilizar dependerán de los requerimientos y las características específicas del proyecto de desarrollo de software. Debemos tener en cuenta: (Drake, 2008)

- Selección del lenguaje de programación: Se elige un lenguaje de programación adecuado según los requerimientos y las necesidades del proyecto. Ejemplos comunes incluyen Java, Python, C++, C#, JavaScript, entre otros.
- Desarrollo de algoritmos y estructuras de datos: Se implementan algoritmos y se utilizan estructuras de datos apropiadas para manejar y manipular la información requerida por el software.
- Implementación de la lógica del software: Se escribe el código fuente utilizando el lenguaje de programación elegido para implementar la lógica y la funcionalidad requerida por el software.
- Utilización de *frameworks* y bibliotecas: Se utilizan *frameworks* y bibliotecas disponibles en el lenguaje de programación para acelerar el desarrollo y utilizar funcionalidades predefinidas que cumplan con los requerimientos establecidos.

3.3.16 Sistemas Operativos

La asignatura entrega los conocimientos fundamentales sobre los componentes estructurales y funcionales de un sistema operativo

Los contenidos a tratar en la asignatura son:

1. Organización y estructura de un sistema operativo moderno.

2. Planificación de tareas, gestión de interrupciones y del tiempo.
3. Interfaces de programación y llamadas al sistema.
4. Gestión de memoria, memoria virtual y algoritmos de remplazo de páginas.
5. Procesos, concurrencia y mecanismos de sincronización.
6. Gestión de discos y sistemas de archivos.

Los resultados de aprendizaje que se esperan lograr son:

- I. Describe los objetivos, la estructura, y las funciones de los sistemas operativos actuales, relacionándolos con instancias concretas (por ejemplo, Linux).

El sistema operativo es el núcleo de un ordenador, sin este software complejo, el resto de los programas no pueden funcionar. Las tareas de las que se encarga son de lo más diversas. Algunas se ejecutan completamente en segundo plano, y muchas ocurren en paralelo. Los sistemas operativos tienen varios objetivos principales, incluyendo la gestión eficiente de recursos del sistema, proporcionar una interfaz de usuario para interactuar con el hardware y los programas, garantizar la seguridad y protección de los datos. Un ejemplo concreto de un sistema operativo es Linux.

Linux es el nombre que reciben una serie de sistemas operativos de tipo Unix bajo la licencia GNU GPL (*General Public License* o Licencia Pública General de GNU) que son su mayoría gratuitos y con todo lo necesario para hacer funcionar un PC, con la peculiaridad de que podemos instalar un sistema muy ligero e ir añadiendo todo lo necesario posteriormente o según lo vayamos necesitando. Linux es multiusuario, multitarea y multiplataforma, además puede funcionar en modo consola para un consumo mínimo de recursos, pero que también podemos hacer funcionar con entorno gráfico, instalando uno mediante comandos de terminal o adquiriendo un paquete en el que venga uno incluido. Algunos de los objetivos, estructura y funciones clave de Linux incluyen: (Soto, 2020)

- **Objetivos:** Los objetivos de Linux incluyen la estabilidad, rendimiento, seguridad y flexibilidad. También se enfoca en ser un sistema operativo de código abierto, lo que significa que su código fuente está disponible públicamente y puede ser modificado y distribuido libremente. (Soto, 2020)
- **Estructura:** Linux sigue una arquitectura modular. El núcleo del sistema operativo, conocido como *kernel* de Linux, se encarga de la administración de recursos y proporciona servicios fundamentales. Además del *kernel*, Linux incluye otros componentes en los que destacan el *bootloader*, el *shell*, el

software GNU, el servidor gráfico, el entorno de escritorio, entre otros. (Lucas, 2017)

- Funciones: Linux proporciona una variedad de funciones importantes, como la administración de memoria, la gestión de procesos, el control de dispositivos, el sistema de archivos, la red y la seguridad. Además, Linux es compatible con una amplia gama de hardware y ofrece una interfaz de línea de comandos. Además, es un sistema que posee una gran variedad de funciones. Sistema multitarea, multiusuario, *shells* programables y capacidad de independencia. (Alfred, 2019)

- II. Explica la arquitectura de un sistema operativo, que comprende su estructura, componentes y relaciones, integrando también los mecanismos y políticas asociados.

La arquitectura de un sistema operativo se refiere a la estructura y organización interna del sistema operativo, así como a los componentes y sus relaciones. También incluye los mecanismos y políticas asociados que permiten el funcionamiento del sistema. Algunos aspectos clave de la arquitectura de un sistema operativo son:

- Estructura: La estructura de un sistema operativo puede ser monolítica, en capas, modular o *microkernel*.
 - Arquitectura monolítica: El sistema está constituido por un único programa compuesto de múltiples rutinas o subprogramas que pueden ser llamadas unas a otras ya que cualquier procedimiento puede invocar a otro. Se utilizó en los primeros sistemas operativos. (Klaasvangend, n.d.)
 - Arquitectura en capas: está constituido por una serie de capas o anillos que se comunican entre sí atendiendo a las funciones que puede realizar. El sistema operativo consta de una estructura que parte de una capa núcleo que tiene relación con el hardware y se va completando en capas de modo que cada capa suministra servicio a la capa siguiente. (Klaasvangend, n.d.)
 - Arquitectura modular o *microkernel*: se constituye de un núcleo que brinde un manejo mínimo de procesos, memoria y, además, provea de una capa de comunicación entre procesos. La capa de comunicación es la funcionalidad principal del sistema. (Klaasvangend, n.d.)
- Componentes: Los componentes principales de un sistema operativo incluyen el núcleo (*kernel*), controladores de dispositivos, gestor de memoria, planificador de tareas, gestor de archivos, gestor de red, interfaz de usuario, entre otros.

Cada componente cumple una función específica en el sistema operativo. (Espino, 2015)

- Relaciones: Los componentes del sistema operativo se comunican y cooperan entre sí mediante interfaces y mecanismos de comunicación, como llamadas al sistema, interrupciones y eventos. Estas relaciones permiten el intercambio de información y la ejecución de tareas en el sistema operativo. (Cervantes, n.d.)
- Mecanismos y políticas: Los mecanismos son las estructuras y funciones internas que implementan las operaciones del sistema operativo, mientras que las políticas son las reglas y decisiones que determinan cómo se utilizan esos mecanismos.

III. Describe los principales problemas de control de concurrencia, aplicando las técnicas de solución adecuadas.

El control de concurrencia se refiere a la gestión de la ejecución simultánea de múltiples procesos o hilos en un sistema operativo. Los problemas de control de concurrencia más conocidos son:

- Sección crítica: se trata de una sección del código que no debe ser ejecutada por varios procesos o hilos simultáneamente. Un ejemplo de sección crítica es la sección de código en la que se accede a un recurso compartido. Para evitar el acceso simultáneo a la sección crítica se emplean mecanismos que garantizan la exclusión mutua. Las técnicas de solución incluyen el uso de exclusión mutua, como semáforos o *mutex*, para garantizar que solo un proceso pueda acceder a un recurso compartido a la vez.
- Interbloqueo (*deadlock*): es el bloqueo permanente de un conjunto de procesos o hilos de ejecución en un sistema concurrente que compiten por recursos del sistema o bien se comunican entre ellos. A diferencia de otros problemas de concurrencia de procesos, no existe una solución general para los interbloqueos. Todos los interbloqueos surgen de necesidades que no pueden ser satisfechas, por parte de dos o más procesos. Para evitar el interbloqueo, se utilizan algoritmos de detección y prevención de interbloqueo, como el algoritmo del banquero o el enfoque de evitar la espera circular. (Coulouris, 2012)
- Condición de carrera (*race condition*): Ocurre cuando el resultado de la ejecución depende del orden y el tiempo en que se producen eventos concurrentes. Cuando la salida o estado de un proceso es dependiente de una secuencia de eventos que se ejecutan en orden arbitrario y van a trabajar sobre un mismo recurso compartido, se puede producir un error cuando dichos eventos no llegan

(se ejecutan) en el orden que el programador esperaba. Para solucionar este problema, se pueden utilizar técnicas como el uso de exclusión mutua, sincronización de hilos mediante semáforos o monitores, y el uso de primitivas de sincronización como variables de condición. (Robert M. Fuhrer, n.d.)

- *Starvation*: es un problema relacionado con los sistemas multitarea, donde a un proceso o un hilo de ejecución se le deniega siempre el acceso a un recurso compartido. Sin este recurso, la tarea a ejecutar no puede ser nunca finalizada. Para evitar la inanición, se pueden aplicar políticas de planificación justa, como la prioridad basada en envejecimiento o la asignación justa de recursos. (Joseph, 2016)

IV. Analiza los diferentes métodos y políticas de planificación de tareas y de asignación del procesador, evaluando cómo influyen en el desempeño del sistema.

Planificación de tareas: Se refiere a cómo se asigna el tiempo de CPU a los procesos o hilos en el sistema:

- Planificación de prioridad: Esta planificación se caracteriza porque a cada proceso se le asigna una prioridad y se continúan con un criterio determinado. Los procesos serán atendidos de acuerdo con la prioridad determinada. (Campos, 2021)
 - Planificación por turnos (*round-robin*): Es un algoritmo donde se determina el mismo tiempo para la ejecución de todos los procesos. Si un proceso no puede ejecutarse por completo en el tiempo asignado su ejecución será después de la ejecución de todos los procesos que se ejecuten con el tiempo asignado. Este algoritmo se fundamenta en FCFS y ordena la cola de procesos circularmente cuando se hallan en estado de listos. (Campos, 2021)
 - Planificación basada en colas múltiples: Derivado de MQS (*Multilevel Queue Scheduling*). Es un algoritmo donde la cola de procesos en estado de listos se divide en varias colas más pequeñas. Los procesos se clasifican a partir de un criterio que determina en qué cola se ubicará el proceso cuando se encuentre en estado de listo. (Campos, 2021)
- Asignación del procesador: Se refiere a cómo se asignan los procesos a los núcleos o CPU disponibles en un sistema multiprocesador: (Cabalar, n.d.)
 - Asignación basada en prioridad: Asigna los procesos a los núcleos en función de su prioridad. Los procesos con mayor prioridad se ejecutan en núcleos más rápidos o dedicados.

- Asignación basada en carga: Asigna los procesos a los núcleos en función de la carga de trabajo de cada núcleo. Los procesos se asignan a los núcleos menos ocupados para equilibrar la carga del sistema.
- Asignación basada en afinidad: Asigna procesos a núcleos específicos para aprovechar la afinidad entre el proceso y el núcleo. Esto puede mejorar el rendimiento al minimizar el cambio de contexto y la caché de fallos.

V. Determina la organización del almacenamiento de datos en medios persistentes, analizando los métodos de acceso asociados.

El almacenamiento de datos en medios persistentes se refiere a cómo se organizan y se acceden a los datos en dispositivos de almacenamiento a largo plazo, como discos duros o unidades de estado sólido. Algunos aspectos clave de la organización del almacenamiento de datos y los métodos de acceso asociados son:

- Sistemas de archivos: Es el sistema de almacenamiento de un dispositivo de memoria, que estructura y organiza la escritura, búsqueda, lectura, almacenamiento, edición y eliminación de archivos de una manera concreta. El objetivo principal de esta organización es que el usuario pueda identificar los archivos sin lugar a error y acceder a ellos lo más rápido posible. (ionos, 2020)
- Métodos de acceso: Los métodos de acceso determinan cómo se accede a los datos dentro de los sistemas de archivos. Algunos métodos comunes incluyen:
- Acceso secuencial: Este método organiza los registros secuencialmente en el orden en que se ingresan. Un conjunto de datos que tiene esta organización es un conjunto de datos secuenciales. Las operaciones más comunes son lecturas y escrituras.
- Acceso indexado: es un método de acceso que permite buscar archivos sin ningún orden, y para ello utiliza listas con un campo clave o más datos del archivo. Se utiliza en búsquedas para encontrar cierta posición o cierto valor en un archivo. Aunque tiene grandes ventajas en búsquedas y facilita su procesamiento, no permite recorrer los archivos secuencialmente, solo guarda la dirección de un archivo y la almacena, no guarda una dirección del siguiente archivo por lo cual tiene que llegar a él por otro método. (Valero, 2021)

VI. Instala uno o más Sistemas Operativos configurándolo para su operatividad.

La instalación de un sistema operativo implica el proceso de configurar y preparar el sistema para ejecutar el sistema operativo seleccionado. (Fidalgo, 2018)

- Preparación del entorno de instalación: Se crea o se obtiene el medio de instalación adecuado para el sistema operativo seleccionado. Esto puede ser un DVD, una unidad USB o una imagen de disco descargada desde Internet.
- Arranque del entorno de instalación: Se inicia el sistema desde el medio de instalación. Esto puede requerir configurar el orden de arranque en la configuración del BIOS o seleccionar manualmente el medio de arranque durante el inicio.
- Instalación: Se siguen los pasos proporcionados por el asistente de instalación del sistema operativo. Esto puede incluir la selección del idioma, la aceptación de los términos de licencia, la partición del disco duro, la selección de los componentes y la configuración de la red.
- Configuración del sistema operativo: Una vez completada la instalación, se realiza la configuración inicial del sistema operativo. Esto puede incluir la configuración de la fecha y la hora, la configuración de la red, la creación de cuentas de usuario y la personalización de las preferencias del sistema.
- Actualizaciones y controladores: Se aplican las actualizaciones de seguridad y se instalan los controladores necesarios para que el sistema operativo funcione correctamente con el hardware específico del sistema.
- Configuración adicional: Se realizan configuraciones adicionales según los requisitos y las preferencias del sistema operativo. Esto puede incluir la instalación de software adicional, la configuración de servicios y la personalización de la interfaz de usuario.

La instalación y configuración de un sistema operativo puede variar según el sistema operativo específico y la plataforma de hardware utilizada. Es importante seguir las instrucciones proporcionadas por el proveedor del sistema operativo y tener en cuenta los requisitos del sistema y las configuraciones recomendadas.

3.4 MATRIZ DE CONVERGENCIA DE LA ESPECIALIDAD DE PROGRAMACION EMTP - T.U. INFORMÁTICA USM.

Para generar el modelo de articulación entre la EMTP - T.U. Informática USM, se propone una matriz de convergencia para la articulación entre un sector y una especialidad determinada. Esto en búsqueda de una articulación de asignaturas de la carrera en la USM y asignaturas EMTP. Para esto, la matriz de convergencia es la herramienta para exponer los desempeños esperados por nivel, en comparación con el desempeño del estudiante y los criterios de evaluación. La matriz de convergencia permite visualizar las posibles articulaciones en el T.U de informática.

Ilustración 8 Matriz Convergencia USM-EMTP [Fuente: Creación propia a partir de contenido curricular declarativo de la Especialidad y Sector Tecnologías y comunicaciones, y la carrera T.U. Informática.]. (Creación propia)

		Nivel 3ero Medio				
		OA	1	1	1	
		Módulos	M1. Programación y base de	M1. Programación y base de	M1. Programación y base de	
		Aprendizajes Esperados / Resultados de	AP1.Construye algoritmos y	AP2.Desarrolla programas	AP3.Desarrolla programas de	
Análisis de Sistemas de Información		RDA1. Caracteriza los Sistemas de	X			
Análisis de Sistemas de Información		RDA2. Define las etapas del ciclo de	X			
Análisis de Sistemas de Información		RDA3. Aplica metodología para el	X	X		
Análisis de Sistemas de Información		RDA4. Define el rol del personal de		X	X	
Análisis de Sistemas de Información		RDA5. Relaciona funciones básicas de la	X			
Programación		RDA1. Identifica y relaciona conceptos	X	X		X
Programación		RDA2. Diseña algoritmos	X	X		X
Programación		RDA3. Aplica estructura de datos	X	X		X
Programación		RDA4. Implementa programas de	X	X		X
Introducción a la Informática y Computación		RDA1. Identifica los aspectos	X			
Introducción a la Informática y Computación		RDA2. Describe los elementos		X		X
Introducción a la Informática y Computación		RDA3. Describe los modos de	X	X		X
Introducción a la Informática y Computación		RDA4. Aplica los sistemas numéricos y	X	X		
Introducción a la Informática y Computación		RDA5. Utiliza herramientas de ofimática		X		
Análisis y Diseño Orientado a Objeto		RDA1. Describe los conceptos				X
Análisis y Diseño Orientado a Objeto		RDA2. Describe los conceptos				X
Análisis y Diseño Orientado a Objeto		RDA3. Desarrolla diagramas a nivel de	X			X
Análisis y Diseño Orientado a Objeto		RDA4. Desarrolla diagramas a nivel de	X			
Análisis y Diseño Orientado a Objeto		RDA5. Describe las etapas del proceso				X
Análisis y Diseño Orientado a Objeto		RDA6. Aplica metodología de análisis y		X		
Diseño de Sistemas de Información		RDA1. Describe las actividades del				
Diseño de Sistemas de Información		RDA2. Describe la fase de diseño de				
Diseño de Sistemas de Información		RDA3. Aplica los diferentes tipos de				
Diseño de Sistemas de Información		RDA4. Describe las herramientas de la				
Diseño de Sistemas de Información		RDA5. Describe conceptos de auditoría				
Estructura de Datos		RDA1. Aplica las técnicas de	X	X		
Estructura de Datos		RDA2. Aplica estructuras de datos	X	X		X
Estructura de Datos		RDA3. Aplica recursividad como método	X			
Estructura de Datos		RDA4. Aplica diversos métodos de	X	X		
Estructura de Datos		RDA5. Aplica distintos tipos de	X	X		

PROPUESTA DE UN PLAN DE ESTUDIOS PARA LA FORMACIÓN TÉCNICO - PROFESIONAL EN INFORMÁTICA ARTICULABLE CON LA EDUCACIÓN SUPERIOR

4to. Medio					
9		2		2	
M4. Sistemas operativos	M4. Sistemas operativos	M5. Diseño de base de datos	M5. Diseño de base de datos	M5. Diseño de base de datos	M5. Diseño de base de datos
AP1. Instala sistemas	AP2. Instala sistemas	AP1. Instala sistemas de	AP2. Construye tablas de datos	AP3. Construye modelos de	AP4. Implementa modelos de
X		X	X		X
X			X		X
X		X	X	X	X
	X		X	X	X
	X	X	X	X	X
X	X	X	X	X	X
	X	X	X	X	X
X	X	X	X	X	X
X	X	X	X	X	X
			X	X	X
X		X	X	X	X
	X	X	X	X	X
X		X	X		X
		X	X	X	X
		X			
		X	X		
		X	X	X	
		X			X
	X	X			X
X	X				
	X		X		
			X		
		X			X
		X	X	X	
			X		
				X	

[illegible]

		Nivel OA	3ero Medio		
		Módulos	1	1	1
		Aprendizajes Esperados / Resultados de	M1. Programación y base de	M1. Programación y base de	M1. Programación y base de
			AP1.Construye algoritmos y	AP2.Desarrolla programas	AP3.Desarrolla programas de
Programación Orientada a Evento	RDA1. Aplica conceptos de		X		X
Programación Orientada a Evento	RDA2. Utiliza lenguaje de programación		X		X
Taller de Sistemas de Información I	RDA1. Aplica metodología para definir				
Taller de Sistemas de Información I	RDA2. Desarrolla actividades de análisis				
Taller de Sistemas de Información I	RDA3. Desarrolla actividades de diseño				
Programación Orientada a Objeto	RDA1. Aplica conceptos de		X		
Programación Orientada a Objeto	RDA2. Utiliza lenguaje de programación		X		X
Diseño y Programación Orientada a la Web	RDA1. Define entorno de desarrollo de		X		X
Diseño y Programación Orientada a la Web	RDA2. Aplica patrón de diseño modelo-		X		X
Diseño y Programación Orientada a la Web	RDA3. Implementa sistemas de			X	
Diseño y Programación Orientada a la Web	RDA4. Utiliza Search Engine				
Bases de Datos	RDA1. Analiza el rol de las bases de			X	
Bases de Datos	RDA2. Analiza el rol de los modelos de			X	
Bases de Datos	RDA3. Identifica características				
Bases de Datos	RDA4. Diseña modelos de datos		X		
Bases de Datos	RDA5. Desarrolla bases de datos,		X		
Arquitectura y Organización de Computadores	RDA1. Diseña sistemas lógicos simples,			X	
Arquitectura y Organización de Computadores	RDA2. Explica el funcionamiento lógico				
Arquitectura y Organización de Computadores	RDA3. Crea programas simples,				
Arquitectura y Organización de Computadores	RDA4. Analiza el desempeño de una				X
Taller de Sistemas de Información II	RDA1. Realiza pruebas de cumplimiento				
Taller de Sistemas de Información II	RDA2. Corrige programación,				
Taller de Sistemas de Información II	RDA3. Implementa sistemas de				
Desarrollo de Aplicaciones Móviles	RDA1. Aplica conceptos de		X	X	X
Desarrollo de Aplicaciones Móviles	RDA2. Aplica conceptos de diseño		X		
Desarrollo de Aplicaciones Móviles	RDA3. Desarrolla aplicaciones				X
Taller de Desarrollo de Software	RDA1. Aplica conceptos metodologías y		X		
Taller de Desarrollo de Software	RDA2. Aplica lenguaje de programación		X	X	
Sistemas Operativos	RDA1. Describe los objetivos, la				X
Sistemas Operativos	RDA2. Explica la arquitectura de un				
Sistemas Operativos	RDA3. Describe los principales				
Sistemas Operativos	RDA4. Analiza los diferentes métodos y				
Sistemas Operativos	RDA5. Determina la organización del				
Sistemas Operativos	RDA6. Instala uno o más Sistemas				

[illegible]

PROPUESTA DE UN PLAN DE ESTUDIOS PARA LA FORMACIÓN TÉCNICO - PROFESIONAL EN INFORMÁTICA ARTICULABLE CON LA EDUCACIÓN SUPERIOR

4to. Medio					
9	9	2	2	2	2
M4. Sistemas operativos	M4. Sistemas operativos	M5. Diseño de base de datos	M5. Diseño de base de datos	M5. Diseño de base de datos	M5. Diseño de base de datos
AP1. Instala sistemas	AP2. Instala sistemas	AP1. Instala sistemas de	AP2. Construye tablas de datos	AP3. Construye modelos de	AP4. Implementa modelos de
X			X		
			X	X	
					X
			X		
	X			X	
		X	X		X
	X		X		X
	X			X	X
				X	
	X		X		
			X		
		X	X		
					X
	X		X	X	
	X				
X	X	X			
	X				
		X			X
X	X		X	X	

4	4	4	4	3	3	5	5	5
M6. Programación orientada a	M6. Programación orientada a	M6. Programación orientada a	M6. Programación orientada a	M7. Administración de base	M7. Administración de base	M8. Desarrollo de aplicaciones	M8. Desarrollo de aplicaciones	M8. Desarrollo de aplicaciones
AP1. Construye unidades de	AP2. Detecta y corrige errores	AP3. Construye aplicaciones	AP4. Utiliza componentes	AP1. Desarrollar unidades de	AP2. Programa aplicaciones	AP1. Diseña aplicaciones web	AP2. Desarrolla elementos	AP3. Evalúa el funcionamiento
X	X	X	X					
X	X	X	X					
X	X	X	X		X		X	
		X	X	X	X	X	X	
					X	X		
				X				X
X	X		X	X		X	X	
X			X					X
X			X					
		X	X		X		X	
	X	X		X		X		
	X				X			X
					X	X		
X	X				X		X	X
	X				X			
X			X		X	X	X	
X								

A partir de los aprendizajes esperados y resultados de aprendizaje, se puede realizar el análisis curricular de la EMTP como en la educación superior, con el objetivo de lograr la articulación dentro de la especialidad de programación. Esto permite favorecer la transición entre la especialidad en la EMTP a la educación superior. Lo que nos ofrece verificar las posibles convalidaciones de las asignaturas y enfocarnos en comprobar que los alumnos tienen los conocimientos correspondientes según malla curricular.

3.5 ASIGNATURAS CONVALIDABLES.

Para poder convalidar una asignatura, se debe cumplir con un 75% o más de convergencia en aprendizajes y resultados esperados, para articularlo sin una prueba previa. Por otro lado, si dicho porcentaje de convergencia varía entre el 60% a 74.9%, se debe realizar una evaluación de conocimientos para verificar los conocimientos del alumno, en caso de que apruebe la evaluación, dicha asignatura será articulable con el plan de estudios del T.U. Informática.

- Convalidación directa:
 - Programación con un 80.8%.
- Convalidación por medio de prueba:
 - Análisis de Sistemas de Información con un 62.3%.
 - Introducción a la Informática y Computación con un 61.5%.

CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN

4.1 CONVERSACIÓN CON PROFESOR DEL LICEO NUEVO MUNDO DE MULCHÉN SOBRE LA ESPECIALIDAD DE PROGRAMACIÓN EN LA EMTP.

En conversaciones con el profesor Rodrigo Ruiz del Liceo Nuevo Mundo de Mulchén menciono que efectivamente se rigen por la malla curricular que presenta el Mineduc para programación en EMTP, por ende, se analizó la matriz de convergencia USM-EMTP (especialidad programación y T.U. Informática), se revisaron y conversaron los casos puntuales de las posibles asignaturas a convalidar en la USM, nos mencionó que las posibles convalidaciones arrojadas por la matriz de convergencia son correctas, pero nos recomendó que en todas las asignaturas que presentaran algún tipo de convalidación se debiese realizar un examen para medir los conocimientos que traen los alumnos y así poder entender si son aptos para la convalidación o no, ya que existen alumnos de la EMTP sin los conocimientos necesarios para encontrarse a un nivel de educación superior, esto debido a distintos factores, que el alumno en la EMTP aprobó dichas asignaturas que facilitan convalidar asignaturas en la educación superior, sin embargo no fue constante en la EMTP, debido a esto no logro demostrar un completo aprendizaje de los contenidos, de tal manera que sería perjudicial realizar una convalidación en asignaturas del T.U. Informática.

4.2 CONVERSACIÓN CON PROFESOR DEL LICEO COMERCIAL DE TALCAHUANO SOBRE LA ESPECIALIDAD DE PROGRAMACIÓN EN LA EMTP.

En conversaciones con el profesor Eric Silva del Liceo Comercial de Talcahuano, menciono que se rigen por la malla curricular que presenta el Mineduc, pero ellos como establecimiento cambian el orden en que dictan ciertos contenidos de la malla para facilitar el aprendizaje de los estudiantes, ya que la actual malla curricular presenta falencias en el orden en que abarcan los contenidos, por ejemplo, nos comentó que administración de bases de datos es como volver atrás con los estudiantes de 4to medio, debido a que en 3ero medio tienen programación y bases de datos donde ven contenidos más complejos y aplicables que en administración de bases de datos. Además, nos hace notar que el enfoque propuesto por el Mineduc con la malla curricular es que el alumno aprenda lo técnico por encima de lo teórico, con esto olvidándose de la base (tienden a mecanizar), sin embargo, como establecimiento tratan de enseñarles una base teórica sobre los contenidos, debido a que en la programación es necesario, ya sea que los alumnos decidan continuar estudios superiores o no, la base teórica la mantendrán. En Chile al igual que en Alemania existe el sistema de educación Dual, pero con la diferencia que aquí en Chile no existe un apoyo de

parte de las empresas, porque estas buscan gente que venga y realice el trabajo, sin darle importancia en acompañar a los estudiantes en un proceso de formación, debido a que son costos que no están dispuestos a solventar. Por otro lado, igual sucede que a nivel establecimiento y nacional en liceos de EMTP, existe un gran déficit de profesores capacitados y dispuestos a enseñar en la especialidad de programación.

4.3 PROPUESTA DE CAMBIO EN LA MALLA CURRICULAR DE LA EMTP.

Considerando que el campo de la informática es tan amplio, existen diferentes estilos de mejoras en el actual plan curricular de programación, tanto como modificaciones puntuales, a nivel macro o dividir programación en menciones.

Respecto a la investigación sobre la educación técnica en la secundaria de otros países (EE. UU., Austria, Alemania y Japón), y analizar como imparten la programación en su modelo educativo y aplicación de la *IVET* para la formación de sus estudiantes, proponen herramientas, contenidos y metodologías de aprendizaje. Con esto logran desarrollar competencias técnicas de programación, diseñar soluciones y adaptarse al trabajo en equipos multidisciplinarios. También, es una buena alternativa complementar los estudios con cursos y material de aprendizaje que se encuentre en internet, de esta manera se haría un buen uso de las tecnologías para que el alumno estudie los contenidos a su ritmo y pueda reforzar de la mejor manera los contenidos que se le dificulta aprender.

Actualmente la formación en programación en la EMTP consta con una preparación de 2 años, se propone como modificación macro que se enfoque en temas de ciencias de la computación, y en metodologías y habilidades de dirección de proyectos informáticos. Además, si se pudiera contar con los dos primeros años de educación media para la formación de la especialidad, se sugiere enfocarse en enseñar la base de la programación. En el transcurso de los primeros dos años (primero y segundo medio) los alumnos debieran aprender a programar de manera aplicada, metodologías de programación y reforzar las habilidades blandas, por medio de equipos de desarrollos.

Un plan curricular actualizado debe tener una base teórica con aplicaciones prácticas de manera que se asemeje al estilo de educación dual de Alemania.

Sera importante que los alumnos desarrollen habilidades técnicas de programación, entender y manipular datos e información, desarrollar pensamiento algoritmo para comprender problemas no triviales, adaptabilidad a nuevos problemas y diseñar soluciones. Se sugiere que el alumno comience a desarrollar un portafolio desde primero medio, para cuando egrese de EMTP demuestre sus habilidades técnicas, manejo de datos y

conocimiento teórico. El lenguaje de programación base que se sugiere enseñar es Python, ya que, entre sus características destaca que:

- Es simple de aprender.
- Es un lenguaje interpretado.
- Es de alto nivel.
- Es dinámico.
- Es multiparadigma.
- Tiene tipado dinámico.

También se sugiere el aprendizaje de algún *framework* web como Spring Framework, Una de sus características más destacables es la inyección de dependencias, la cual permite la unión, el mantenimiento y la reutilización de clases complejas, además, con una arquitectura de *MVC (Model, View, Controller)*. En *front end* será necesario aprender HTML, CSS y JS (JavaScript). Se recomienda que JS se aprenda con algún *framework* que permita sacarle un mayor provecho, por ejemplo, React o VueJS.

Se propone realizar cambios en el actual Programa de estudios para la especialidad de Programación en la EMTP:

- Eliminar la asignatura Soporte a usuarios y tareas de mantención de equipos.
- Generar un portafolio de programación desde primero medio, reforzara contenidos mediante el desarrollo de proyectos concretos.
- Se considera el desarrollo web desde el comienzo, para mejorar habilidades con los *frameworks*.
- La asignatura Programación y bases de datos, se divide en dos asignaturas por separado.
- Emprendimiento y empleabilidad, debería llamarse Emprendimiento y empleabilidad informática, en el cual se dará énfasis en la organización del portafolio del alumno para presentarlo en el mundo laboral.
- Se utilizan metodologías ágiles del desarrollo de software.

En conversaciones con los profesores de los liceos de EMTP, se conversa que en el nuevo plan curricular debería separarse en varias menciones, por ejemplo, desarrollador de videojuegos, machine learning, entre otros, o continuar con una sola mención, pero generar una propuesta de plan de estudios que sea focalizado en una área de la informática y no diversificada, debido a que el actual plan de estudios abarca un poco de todo de distintas áreas, sin formar especialistas en una rama específica de la informática.

Finalmente, existe un gran problema, ya que los profesores de matemáticas y computación a nivel nacional son pocos y esta carrera es la que habilita a los docentes a realizar clases en la EMTP, debido a que pocas universidades dictan Pedagogía en matemáticas y computación, se dificulta encontrar profesores para dictar las asignaturas en la EMTP, y para complejizar más dicho problema, sucede que los profesores titulados de dicha carrera normalmente tienen un enfoque de que ven la computación como una herramienta y no con un enfoque de ciencias de la computación y programación.

4.4 MALLA CURRICULAR PARA PROGRAMACION EN EMTP CON CAMBIOS DE ACTUALIZACIÓN.

A continuación, se presenta como se vería una propuesta de cómo se podría ver un programa de estudios en la EMTP con los cambios propuestos.

1. Primero medio

- a. Introducción a la programación I.
 - i. Programación con bloques.
 - ii. Pensamiento algorítmico.
 - iii. Fundamentos de los paradigmas de programación.
- b. Configuración de equipos informáticos.
 - i. Funcionamiento de los computadores.
 - ii. Fundamentos y lógica de busca de internet.

2. Segundo medio

- a. Introducción a la programación II.
 - i. Introducción estructuras de datos.
 - ii. Programación imperativa.
 - iii. Introducción a la programación orientada a objetos.
- b. Introducción a las bases de datos.
 - i. Fundamentos de las bases de datos.
 - ii. Modelos y gestores de bases de datos.
 - iii. Conceptos y arquitectura de las bases de datos.

3. Tercero medio

- a. Programación.
 - i. Estructuras de datos.
 - ii. Programación orientada a objetos.
- b. Bases de datos.

- i. Implementación de modelos de representación gráfica en bases de datos.
 - ii. Bases de datos en la nube.
- c. Introducción al desarrollo web.
 - i. Modelo vista controlador.
 - ii. HTML y CSS.
 - iii. Introducción a JavaScript.
 - iv. Frameworks.

4. Cuarto medio

- a. Ingeniería de software
 - i. Procesos y desarrollos de software.
 - ii. Implementación de metodologías ágiles en proyectos.
- b. Arquitectura y seguridad informática.
 - i. Fundamentos, desempeño de computadores y arquitectura de un computador.
 - ii. Introducción a DevOps.
 - iii. Introducción y fundamentos de la ciberseguridad.
- c. Emprendimiento y empleabilidad informática
 - i. Desarrollo de habilidades blandas.
 - ii. Actualización de un portafolio.

4.5 COMPARACIÓN PLAN DE ESTUDIOS ACTUAL Y LA PROPUESTA DE PLAN DE ESTUDIOS PARA PROGRAMACIÓN EN EMTP.

A continuación, veremos el actual plan de estudios y la propuesta de plan de estudios, con esto lograremos visualizar y explicar las modificaciones de mejora para formar especialistas en programación.

Plan de estudios actual del Mineduc	Propuesta de plan de estudios
	Primero medio 1. Introducción a la programación I. a. Programación con bloques. b. Pensamiento algorítmico. c. Fundamentos de los paradigmas de programación.

	2. Configuración de equipos informáticos. a. Funcionamiento de los computadores. b. Fundamentos y lógica de busca de internet.
	Segundo medio 3. Introducción a la programación a. Introducción estructuras de datos. b. Programación imperativa. c. Introducción a la programación orientada a objetos. 4. Introducción a las bases de datos. a. Fundamentos de las bases de datos. b. Modelos y gestores de bases de datos. c. Conceptos y arquitectura de las bases de datos.
Tercero medio 1. Programación y bases de datos a. Construye algoritmos y diagramas de flujo. b. Desarrolla programas simples empleando la programación estructurada y/o modular. c. Desarrolla programas de mediana complejidad. 2. Instalación y configuración de equipos informáticos a. Instala componentes internos de hardware de una estación monousuario. b. Realiza mantención de una estación monousuario para potenciar su rendimiento. c. Potencia el rendimiento de un modelo de red. 3. Soporte a usuarios y productividad a. Instala aplicaciones para optimizar tareas en el puesto de trabajo considerando.	Tercero medio 5. Programación. a. Estructuras de datos. b. Programación orientada a objetos. 6. Bases de datos. a. Implementación de modelos de representación gráfica en bases de datos. b. Bases de datos en la nube. 7. Introducción al desarrollo web. a. Modelo vista controlador. b. HTML y CSS. c. Introducción a JavaScript. d. Frameworks.

b. Realiza mantenimiento preventivo de software a un equipo personal de computación. c. Otorga asistencia técnica para corregir y resolver problemas de operación de sistemas. d. Otorga soporte técnico local para resolver problemas de operación del equipo. e. Realiza instalación y configuración de aplicaciones de soporte centralizado. 4. Sistemas operativos a. Instala sistemas operativos en computadores personales a nivel local. b. Instala sistemas operativos con servidores de red para multiusuarios.	
Cuarto medio 5. Diseño de bases de datos relacionales a. Instala sistemas de manejo de datos y entornos de diagramación gráfica para representar relaciones entre datos. b. Construye tablas de datos para representar la estructura de los datos requeridos para resolver un problema simple. c. Construye modelos de datos que respondan a las necesidades de información transaccional de negocios. d. Implementa modelos de representación gráfica en una base de datos. 6. Programación orientada a objetos	Cuarto medio 8. Ingeniería de software a. Procesos y desarrollos de software. b. Implementación de metodologías ágiles en proyectos. 9. Arquitectura y seguridad informática. a. Fundamentos, desempeño de computadores y arquitectura de un computador. b. Introducción a DevOps. c. Introducción y fundamentos de la ciberseguridad. 10. Emprendimiento y empleabilidad informática. a. Desarrollo de habilidades blandas.

- | | |
|---|--|
| <ul style="list-style-type: none"> a. Construye unidades de prueba para verificar el correcto funcionamiento de la codificación realizada. b. Detecta y corrige errores de codificación, analizando el comportamiento del código de programación. c. Construye aplicaciones con clases predefinidas. d. Utiliza componentes reutilizables en un lenguaje de uso estándar. <p>7. Administración de bases de datos</p> <ul style="list-style-type: none"> a. Desarrollar unidades de programación complejas sobre una base de datos. b. Programa aplicaciones para administrar una base de datos. <p>8. Desarrollo de aplicaciones web</p> <ul style="list-style-type: none"> a. Diseña aplicaciones web en lenguajes estándar. b. Desarrolla elementos dinámicos web. c. Evalúa el funcionamiento de una aplicación web según los requerimientos del negocio. <p>9. Emprendimiento y empleabilidad</p> <ul style="list-style-type: none"> a. Diseña y ejecuta un proyecto para concretar iniciativas de emprendimiento. b. Maneja la legislación laboral y previsional chilena. c. Prepara los elementos necesarios para participar de un proceso de incorporación al mundo del trabajo. d. Selecciona alternativas de capacitación y de educación superior para fortalecer sus competencias o desarrollar nuevas y adquirir certificaciones. | <ul style="list-style-type: none"> b. Actualización de un portafolio. |
|---|--|

El actual plan de estudios del Mineduc en EMTP programación se representará por (numero), mientras que la propuesta para un nuevo plan de estudios se representará como (numero*), para hacer referencia a los nombres de las asignaturas.

- La asignatura (1) se dividió en las asignaturas (1*), (3*) y (4*), para brindar una base sólida en dichas ramas de la informática, de dicha manera se logrará mejorar la curva de aprendizaje de los estudiantes en los nuevos desafíos.
- La asignatura (2) se realizaron modificaciones en los contenidos, ya que, habían contenidos que los alumnos aprenderán en el proceso de las distintas asignaturas para configurar el entorno de trabajo y/o estudio, se pueden ver en la asignatura (2*).
- La asignatura (3) se elimina del plan de estudios, debido a que es prioridad darle un enfoque más centralizado en un área de la informática a los alumnos y los contenidos de esta asignatura los aprenderán en el proceso de formación y lo integrarán de manera natural.
- La asignatura (4) se elimina, pero sus contenidos se actualizan y pasan a formar parte de las asignaturas (2*) y (9*).
- La asignatura (5) se dividió en las asignaturas (4*) y (6*), de dicha manera los contenidos de sobre bases de datos se pasan de tal manera que la curva de aprendizaje sea baja para el aprendizaje de los alumnos.
- La asignatura (6) se dividió en las asignaturas (1*), (3*) y (5*), ya que, los contenidos sobre lenguajes de programación y sus paradigmas se dictan para que el alumno tenga facilidad en aprender los contenidos.
- La asignatura (7) se dividió en las asignaturas (4*) y (6*), de dicha manera los contenidos de sobre bases de datos se pasan de tal manera que la curva de aprendizaje sea baja para el aprendizaje de los alumnos.
- La asignatura (8) se fortalecieron sus contenidos dividiéndola en las asignaturas (7*) y (8*), así tener mejores conocimientos en el área de desarrollo web.
- La asignatura (9) se modifican los contenidos y se le da un enfoque principal en que el alumno tenga un buen ingreso en el ámbito laboral, ya sea practica o trabajo como desarrollador.

CAPÍTULO 5: CONCLUSIONES

El propósito de esta investigación se presentaba en analizar la especialidad de Programación en la educación secundaria de otros países y como está siendo implementada en Chile desde el año 2016, en países europeos la educación se rige principalmente por la IVET.

Hablando de Chile, nos enfocamos en la especialidad de programación en la EMTP, la cual es de la disciplina científica. Principalmente nos enfocamos en la comparación por una matriz de convergencia de doble entrada EMTP-USM, se analizaron las posibles articulaciones de asignaturas de la EMTP con el T.U. informática de la UTFSM, con esto notamos que existe una asignatura que presenta una convalidación directa, mientras que otras dos asignaturas presentan posible convalidación por medio de una prueba de conocimientos.

Basándonos en el levantamiento del estado del arte de Chile y los países mencionados anteriormente (EE. UU., Austria, Alemania y Japón), se desarrolló una propuesta de plan curricular para la EMTP en Chile, además, sugiriendo herramientas tecnológicas. Es importante notar que bajo el análisis realizado es necesario reformar el programa de estudios la especialidad de programación en la EMTP, principalmente entregando una base sólida de los conocimientos informáticos para que con el paso del tiempo los alumnos puedan seguir desarrollándose.

TRABAJOS FUTUROS.

- Mantener actualizado las tecnologías que se dictan en el plan curricular, ya que el mundo de la informática avanza a un ritmo exponencial.
- Que los alumnos comiencen desde primero medio a aprender una base de programación.
- Se necesita que los profesores manejen una base sólida sobre programación y ciencias de la informática.
- Lograr el interés de profesionales con experiencia a realizar clases en la EMTP.

REFERENCIAS BIBLIOGRÁFICAS

REFERENCIAS BIBLIOGRÁFICAS EN NORMA APA

(s.f.). Obtenido de https://es.wikipedia.org/wiki/Método_de_acceso

Aeseducation. (2019). *78 Career and Technical Education Facts for 2019*. Obtenido de <https://www.aeseducation.com/career-technical-education-facts-that-prove-its-awesome>

Alexander-cidecmerida. (2010). *Alexander-cidecmerida*. Obtenido de Alexander-cidecmerida: <http://alexander-cidecmerida.blogspot.com/2010/02/sistema-de-administracion-de-base-de.html>

Alfred. (01 de 05 de 2019). *voidlinux*. Obtenido de voidlinux: <https://voidlinux.eu/que-es-y-para-que-sirve-el-sistema-operativo-linux/>

altp. (30 de 04 de 2018). *GSITIC*. Obtenido de GSITIC: <https://gsitic.wordpress.com/2018/04/30/biii9-diseno-de-programas-diseno-estructurado-analisis-de-transformacion-y-de-transaccion-cohesion-y-acoplamiento/>

alumdaci. (2015). *alumdaci*. Obtenido de alumdaci: [https://alumdaci.es.tl/Unidad-Aritmetico_Logica.htm#:~:text=La%20Unidad%20Aritmético%20Lógica%20\(UAL,.\)%2C%20entre%20dos%20números.](https://alumdaci.es.tl/Unidad-Aritmetico_Logica.htm#:~:text=La%20Unidad%20Aritmético%20Lógica%20(UAL,.)%2C%20entre%20dos%20números.)

amazon. (s.f.). *¿Qué es la depuración?* Obtenido de ¿Qué es la depuración?: <https://aws.amazon.com/es/what-is/debugging/#:~:text=La%20depuración%20y%20las%20pruebas%20son%20procesos%20complementarios%20que%20garantizan,para%20identificar%20fallos%20y%20errores.>

Assembly, I. P. (s.f.). *IBM Personal Computer Assembly*. Obtenido de IBM Personal Computer Assembly: https://www.zator.com/Hardware/H3_2.htm

aulafacil. (2018). *aulafacil*. Obtenido de aulafacil: <https://www.aulafacil.com/cursos/hardware/arquitectura-de-computadores/la-unidad-de-control-l33363#:~:text=La%20Unidad%20de%20Control%20es,cada%20una%20de%20las%20tare.>

Austria. (2013). *La educación emocional y social en Austria*. Obtenido de https://fundacionbotin.org/89dguuytdfr276ed_uploads/EDUCACION/ANALISIS%20INTERNACIONAL/2013/Austria_es_2013.pdf

Austria, G. (s.f.). Obtenido de <https://www.educacionyfp.gob.es/austria/estudiar/en-austria.html>

AWS. (s.f.). *¿Qué es el procesamiento por lotes?* Obtenido de ¿Qué es el procesamiento por lotes?: <https://aws.amazon.com/es/what-is/batch-processing/>

ayudaley. (s.f.). *ayudaley*. Obtenido de ayudaley: https://ayudaleyprotecciondatos.es/bases-de-datos/#Caracteristicas_de_las_bases_de_datos

- Bachman, C. W. (17 de 02 de 2013). *Sistema de gestión de bases de datos*. Obtenido de Sistema de gestión de bases de datos:
https://es.wikipedia.org/wiki/Sistema_de_gesti3n_de_bases_de_datos
- Bentley, L. D. (2004). *System analysis and design methods*. Obtenido de System analysis and design methods.:
https://es.wikipedia.org/wiki/Dise1o_de_sistemas#:~:text=El%20dise1o%20l3gico%20de%20un,sistemas%2C%20los%20dise1os%20son%20incluidos.
- Berzal, F. (2005-2006). *El ciclo de vida de un sistema*. Obtenido de El ciclo de vida de un sistema:
<http://flanagan.ugr.es/docencia/2005-2006/2/apuntes/ciclovida.pdf>
- bookdown. (2006). *bookdown*. Obtenido de bookdown:
<https://bookdown.org/paranadagarcia/database/normalizacion.html>
- Cabalar, P. (s.f.). *Depto. de Computaci3n Universidade da Coru1a*. Obtenido de Depto. de Computaci3n Universidade da Coru1a: <https://www.dc.fi.udc.es/~so-grado/SO-Procesos-planif.pdf>
- Cabot, J. (24 de 01 de 2017). *HERRAMIENTAS DE DIBUJO VS HERRAMIENTAS DE MODELADO*. Obtenido de HERRAMIENTAS DE DIBUJO VS HERRAMIENTAS DE MODELADO: <https://ingenieriadesoftware.es/herramientas-de-dibujo-vs-modelado/#:~:text=Una%20herramienta%20de%20modelado%20garantiza,es%20mucho%20mejor%20que%20nada.>
- Campos, M. S. (08 de 10 de 2021). *Revista Virtual Pro*. Obtenido de Revista Virtual Pro:
<https://www.virtualpro.co/noticias/que-es-la-planificacion-de-procesos-de-un-sistema-operativo>
- Canelo, M. M. (02 de 11 de 2020). *Profile*. Obtenido de Profile: <https://profile.es/blog/que-es-la-programacion-orientada-a-objetos/>
- Carisio, E. (s.f.). *mdcloud*. Obtenido de mdcloud: <https://blog.mdcloud.es/que-es-el-diseno-de-base-de-datos-y-como-planificarlo/>
- cavsi. (s.f.). *cavsi*. Obtenido de cavsi: <https://www.cavsi.com/espanol/blog/archivos-de-texto-extensiones/>
- Cervantes, H. (s.f.). *sg*. Obtenido de sg: <https://sg.com.mx/revista/45/las-interfaces-y-la-arquitectura>
- Ceupe, N. (2021). *Ceupe*. Obtenido de Ceupe: <https://www.ceupe.cl/blog/modelos-de-sistemas-web.html>
- Chile, M. d. (10 de 2015). *curriculumnacional*. Obtenido de curriculumnacional:
https://www.curriculumnacional.cl/614/articles-86705_plan.pdf
- Chima, R. (2018). Obtenido de <https://www.studocu.com/es-mx/document/universidad-tecnologica-de-mexico/programacion-orientada-a-eventos/programacion-orientada-a-eventos/21743529>

- CHUWI. (21 de 04 de 2021). *newsroom*. Obtenido de newsroom:
<https://www.chuwi.com/es/news/items/3444.html#:~:text=Por%20lo%20general%2C%20una%20buena,que%205.0%20GHz%20es%20excelente.>
- cillero, m. (s.f.). *Diagrama de Clases*. Obtenido de Diagrama de Clases:
<https://manuel.cillero.es/doc/metodologia/metrica-3/tecnicas/diagrama-de-clases/>
- ClasicWebTools. (2020). *ClasicWebTools*. Obtenido de ClasicWebTools:
<https://clasicwebtools.com/articulos/programacion/Auditoria%20y%20pruebas%20a%20los%20sistemas.html#:~:text=Las%20pruebas%20de%20cumplimiento%20consisten,individuales%2C%20datos%20u%20otra%20informaci3n.>
- Collins-Sussman, B., & Fitzpatrick, B. a. (2004). *Control de versiones*. Obtenido de Control de versiones: https://es.wikipedia.org/wiki/Control_de_versiones
- Coulouris, G. (2012). *Distributed Systems Concepts and Design*. Obtenido de Distributed Systems Concepts and Design:
https://es.wikipedia.org/wiki/Bloqueo_mutuo#:~:text=En%20sistemas%20operativos%2C%20el%20bloqueo,bien%20se%20comunican%20entre%20ellos.
- creately. (18 de 10 de 2022). *creately*. Obtenido de creatively:
<https://creately.com/blog/es/diagramas/tutorial-de-diagrama-de-despliegue/#:~:text=Los%20diagramas%20de%20despliegue%20se,de%20software%20en%20ese%20hardware>
- Debernardi, F. (25 de 07 de 2021). *LinkedIn*. Obtenido de LinkedIn:
<https://www.linkedin.com/pulse/qu3-es-el-dise3o-de-la-interfaz-usuario-dise3ador-ui-ux/?originalSubdomain=es>
- DECSAI. (2019). Obtenido de <https://elvex.ugr.es/idbis/db/docs/design/2-requirements.pdf>
- digital, R. (05 de 08 de 2016). *Ser técnico paga, pero en Alemania*. Obtenido de <https://www.semana.com/educacion/articulo/educacion-tecnica-en-alemania/484541/>
- Drake, J. M. (2008). *Computadores y tiempo real*. Obtenido de Computaderes y tiempo real:
https://www.ctr.unican.es/asignaturas/mc_oo/doc/oo_08_i2_proceso.pdf
- Duò, M. (13 de 07 de 2022). *kinsta*. Obtenido de kinsta: <https://kinsta.com/es/blog/disen3o-de-paginas-web-sensibles/#media-queries>
- Durán, M. (24 de 11 de 2022). *hubspot*. Obtenido de hubspot:
<https://blog.hubspot.es/website/estructuras-de-control>
- Educacion tecnica profesional en chile*. (2012). Obtenido de <https://bibliotecadigital.mineduc.cl/bitstream/handle/20.500.12365/18281/E12-0035.pdf?sequence=1>
- EDUCACIÓN, M. D. (17 de 01 de 2015). *MODIFICA EL DECRETO N° 452, DE 2013, QUE ESTABLECE LAS BASES CURRICULARES PARA LA EDUCACIÓN MEDIA FORMACIÓN DIFERENCIADA TÉCNICO-PROFESIONAL*. Obtenido de <https://www.bcn.cl/leychile/navegar?idNorma=1073922>

- Especialidad Programacion* . (2016). Obtenido de <https://www.curriculumnacional.cl/portal/Educacion-Tecnico-Profesional/Tecnologia-y-Comunicaciones/Especialidad-Programacion/>
- Espino, J. M. (2015). *Universidad de las palmas de gran canaria*. Obtenido de Universidad de las palmas de gran canaria: <http://sopa.dis.ulpgc.es/fso/teoria/pdf/FSO-01.3-estructura%20del%20sistema%20operativo.pdf>
- ESTABLECE BASES CURRICULARES PARA LA EDUCACIÓN MEDIA FORMACIÓN DIFERENCIADA TÉCNICO-PROFESIONAL*. (21 de 11 de 2013). Obtenido de <https://www.bcn.cl/leychile/navegar?idNorma=1056485>
- Euroinnova. (s.f.). *Euroinnova*. Obtenido de Euroinnova: <https://www.euroinnova.cl/blog/que-es-un-programa-de-computadora>
- Fidalgo, F. F. (2018). *Implantacion de sistemas operativos*. Obtenido de Implantacion de sistemas operativos: <https://www.sintesis.com/data/indices/9788413572604.pdf>
- Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., & Berners-Lee, T. (s.f.). *World Wide Web*. Obtenido de World Wide Web: https://es.wikipedia.org/wiki/World_Wide_Web
- FUSIONA, B. (2 de 03 de 2022). *Principios SOLID: por qué usarlos en tu desarrollo de software*. Obtenido de Principios SOLID: por qué usarlos en tu desarrollo de software: <https://fusionea.cl/blog/tecnologia/principios-solid-por-que-usarlos-en-tu-desarrollo-de-software>
- gbaudino. (01 de 03 de 2020). *MetodosDeOrdenamiento*. Obtenido de MetodosDeOrdenamiento: <https://github.com/gbaudino/MetodosDeOrdenamiento>
- Geeks. (s.f.). *Arquitectura basada en Componentes*. Obtenido de Arquitectura basada en Componentes: <https://geeks.ms/jkpelaiez/2009/04/18/arquitectura-basada-en-componentes/#:~:text=Una%20arquitectura%20basada%20en%20componentes,interfaces%20de%20comunicación%20bien%20definidas.>
- Gómez Ramírez, V. M. (2014). *Auditoría informática*. Obtenido de Auditoría informática: https://es.wikipedia.org/wiki/Auditoría_informática
- Hernandez, I. (12 de 07 de 2022). *dreamhost*. Obtenido de dreamhost: <https://www.dreamhost.com/blog/es/formas-optimizar-imagenes-para-web/#:~:text=La%20optimización%20de%20imágenes%20es,que%20se%20preserva%20su%20calidad>
- Hubwieser, P. (Noviembre, 2012). "Computer Science Education in Secondary Schools – The Introduction of a New Compulsory Subject". En P. Hubwieser.
- Humano, C. (s.f.). *watchandact*. Obtenido de watchandact: <https://watchandact.eu/cuales-son-las-herramientas-para-la-gestion-de-datos/>
- IBM. (08 de 03 de 2021). *IBM*. Obtenido de IBM: <https://www.ibm.com/docs/es/ignm/4.2.0?topic=networks-data-lists>

- INVID. (s.f.). *INVID*. Obtenido de INVID: <https://invidgroup.com/es/que-es-el-diseno-orientado-a-objetos-y-como-puede-crear-un-gran-software/>
- IONOS. (23 de 09 de 2020). *Diagrama de componentes: modelado eficiente de sistemas con módulos de software*. Obtenido de Diagrama de componentes: modelado eficiente de sistemas con módulos de software: <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/diagrama-de-componentes/>
- ionos. (11 de 09 de 2020). *ionos*. Obtenido de ionos: <https://www.ionos.es/digitalguide/servidores/know-how/sistemas-de-archivos/>
- Joseph, A. D. (18 de 02 de 2016). *CS 162 Operating Systems and Systems Programming*. Obtenido de CS 162 Operating Systems and Systems Programming: [https://es.wikipedia.org/wiki/Inanición_\(informática\)](https://es.wikipedia.org/wiki/Inanición_(informática))
- joshi, v. (06 de 08 de 2020). *cynoteck*. Obtenido de cynoteck: <https://cynoteck.com/es/blog-post/introduction-to-security-testing/>
- kinsta. (23 de 01 de 2023). *kinsta*. Obtenido de kinsta: <https://kinsta.com/es/base-de-conocimiento/que-es-pseudocodigo/#qu-es-el-pseudocodigo>
- Klaasvangend. (s.f.). *ikastaroak*. Obtenido de ikastaroak: https://ikastaroak.ulhi.net/edu/es/ASIR/ISO/ISO01/es_ASIR_ISO01_Contenidos/website_2_arquitectura_de_un_sistema_operativo.html
- Lasso, I. (3 de 03 de 2015). *Procesador de texto*. Obtenido de Procesador de texto: https://es.wikipedia.org/wiki/Procesador_de_texto
- Laudon, K. C. (2012). *Sistemas de información gerencial (12a.ed. --.)*. México D.F.: Pearson. En K. C. Laudon, *Sistemas de información gerencial (12a.ed. --.)*. México D.F.: Pearson.
- Layton, J. (2014). *Detección y corrección de errores*. Obtenido de Detección y corrección de errores: https://es.wikipedia.org/wiki/Detección_y_corrección_de_errores
- Lewis, J., & Loftus, W. (2019). *Pearson Education Inc*. Obtenido de Pearson Education Inc: https://es.wikipedia.org/wiki/Programación_orientada_a_objetos
- Lineamiento escolar 2022*. (10 de 10 de 2021). Obtenido de <https://www.mineduc.cl/wp-content/uploads/sites/19/2021/11/LineamientosEscolar2022.pdf>
- Lineamientos de evaluación y promoción*. (30 de 06 de 2021). Obtenido de <https://www.curriculumnacional.cl/portal/Comunicaciones/Noticias-2021/246231:Lineamientos-de-Evaluacion-y-Promocion>
- Londoño, T. (s.f.). *Metodología para el Desarrollo de Sistema de Información*. Obtenido de Metodología para el Desarrollo de Sistema de Información: <https://www.monografias.com/trabajos90/metodologia-desarrollo-sistema-informacion/metodologia-desarrollo-sistema-informacion>
- Lucas. (27 de 02 de 2017). *ayudalinux*. Obtenido de ayudalinux: <https://ayudalinux.com/estructura-sistema-linux/>

lucidchart. (s.f.). *lucidchart*. Obtenido de lucidchart: <https://www.lucidchart.com/pages/es/que-es-un-diagrama-de-flujo>

lucidchart. (s.f.). *lucidchart*. Obtenido de lucidchart: <https://www.lucidchart.com/pages/es/diagrama-de-secuencia>

lucidchart. (s.f.). *lucidchart*. Obtenido de lucidchart: <https://www.lucidchart.com/blog/es/como-disenar-una-arquitectura-de-software#:~:text=El%20diseño%20de%20una%20arquitectura,comenzar%20a%20elaborar%20un%20prototipo>

lucidchart. (s.f.). *lucidchart*. Obtenido de lucidchart: <https://www.lucidchart.com/blog/es/como-disenar-una-arquitectura-de-software#:~:text=El%20diseño%20de%20una%20arquitectura,comenzar%20a%20elaborar%20un%20prototipo>.

Lucidchart. (s.f.). *Lucidchart*. Obtenido de Lucidchart: <https://www.lucidchart.com/pages/es/diagrama-entidad-relacion-extendido#:~:text=Un%20diagrama%20entidad-relación%20extendido,la%20programación%20orientada%20a%20objetos>.

M, A. (s.f.). *Procesamiento en tiempo real*. Obtenido de Procesamiento en tiempo real: <https://learn.microsoft.com/es-es/azure/architecture/data-guide/big-data/real-time-processing>

Magenheim, J. (s.f.). "Competence Measurement and Informatics".

Maluenda, R. (21 de 01 de 2021). *Qué es un algoritmo informático: características, tipos y ejemplos*. Obtenido de Qué es un algoritmo informático: características, tipos y ejemplos: https://profile.es/blog/que-es-un-algoritmo-informatico/#¿Que_es_un_algoritmo_informatico

Mancuzo, G. (24 de 06 de 2021). *Diagramas de UML, qué significa esta metodología*. Obtenido de Diagramas de UML, qué significa esta metodología: <https://blog.comparasoftware.com/diagramas-de-uml-que-significa-esta-metodologia/>

Markus Rerych, I. f.-u.-W. (2007). *Wasserfallmodell > Entstehungskontext*. Obtenido de Wasserfallmodell > Entstehungskontext: https://es.wikipedia.org/wiki/Metodología_de_desarrollo_de_software

Martínez, G. M. (s.f.). *Análisis y Diseño de Sistemas*. Obtenido de Análisis y Diseño de Sistemas: <https://www.monografias.com/trabajos5/andi/andi>

mdcloud. (s.f.). *Qué es el diseño de base de datos y cómo planificarlo*. Obtenido de Qué es el diseño de base de datos y cómo planificarlo: <https://blog.mdcloud.es/que-es-el-diseno-de-base-de-datos-y-como-planificarlo/>

Mendoza, M. L. (16 de 07 de 2020). *openwebinars*. Obtenido de openwebinars: <https://openwebinars.net/blog/que-es-un-lenguaje-de-programacion/>

mijacobs. (2023). *microsoft*. Obtenido de microsoft: <https://learn.microsoft.com/es-es/devops/develop/git/what-is-version-control>

- mineduc. (4 de 12 de 2015). *Historia de la Educación Técnico-Profesional*. Obtenido de <https://www.tecnicoprofesional.mineduc.cl/secretaria-tecnico-profesional/historia/>
- Mineduc, C. -C. (Abril de 2020). Obtenido de Estudio sobre trayectorias educativas y laborales de estudiantes de educación media técnico-profesional: https://centroestudios.mineduc.cl/wp-content/uploads/sites/100/2020/04/EVIDENCIAS-46_2020_f02.pdf
- Miro. (s.f.). *Miro*. Obtenido de Miro: <https://miro.com/es/diagrama/que-es-diagrama-entidad-relacion/>
- modelado, a. (s.f.). *¿Qué es el modelado de datos?* Obtenido de *¿Qué es el modelado de datos?*: <https://aws.amazon.com/es/what-is/data-modeling/>
- Monsivais, I. E. (2020). *UML : Lenguaje de Modelado Unificado*. Obtenido de UML : Lenguaje de Modelado Unificado: https://www.itesrc.edu.mx/portal/articles.php?id_art=1#:~:text=UML%20es%20una%20herramienta%20que,y%20diseño%20de%20un%20problema.
- Mousinho, A. (03 de 06 de 2020). *rockcontent*. Obtenido de rockcontent: <https://rockcontent.com/es/blog/que-es-seo/>
- Mtz, V. C. (s.f.). *Organización Directa*. Obtenido de Organización Directa: <https://www.scribd.com/presentation/256477839/Organizacion-Directa>
- Norman, D. A. (20 de 05 de 2018). *Diseño de interfaz de usuario*. Obtenido de Diseño de interfaz de usuario: https://es.wikipedia.org/wiki/Diseño_de_interfaz_de_usuario
- nunezmarlyn. (26 de 09 de 2020). *brainly*. Obtenido de brainly: <https://brainly.lat/tarea/24497229>
- Organization of U.S. Education*. (2008). Obtenido de <https://www2.ed.gov/about/offices/list/ous/international/usnei/us/edlite-org-us.html>
- Ortiz, A. E. (04 de 05 de 2020). *hostdime*. Obtenido de hostdime: <https://blog.hostdime.com.co/pruebas-de-recuperacion-ante-desastres-garantizar-que-su-plan-de-recuperacion-ante-desastres-funcione/#:~:text=La%20prueba%20de%20recuperación%20ante,si%20ocurre%20un%20desastre%20real.>
- Pantoja, E. B. (2004). *Desarrollador de software*. Obtenido de Desarrollador de software: http://www.scielo.org.bo/scielo.php?script=sci_arttext&pid=S1683-07892004000100005
- Parlamentario, A. P. (11 de 09 de 2018). *La decisión de Japón de enseñar programación en todo su sistema educativo*. Obtenido de <https://www.bcn.cl/observatorio/asiapacifico/noticias/programacion-escuelas-japon-educacion-futuro>
- PARLAMENTARIO, P. P.-O. (11 de 09 de 2018). *La decisión de Japón de enseñar programación en todo su sistema educativo*. Obtenido de <https://www.bcn.cl/observatorio/asiapacifico/noticias/programacion-escuelas-japon-educacion-futuro>

- Pascual, J. P. (29 de 04 de 2013). *Análisis orientado a objetos (AOO)*. Obtenido de Análisis orientado a objetos (AOO).: <https://analisisyprogramacionoop.blogspot.com/2013/04/analisis-orientado-objetos-aoo.html>
- Pavez, J. D. (03 de 2022). *Educación Media Técnico-Profesional: una aproximación cualitativa desde sus egresados*. Obtenido de <https://revistas.uft.cl/index.php/rre/article/view/181/230>
- Pérez, F. (2022). *iedge*. Obtenido de iedge: <https://www.iedge.eu/francisco-perez-implantacion-de-sistemas-de-la-informacion>
- pmoinformatica. (10 de 08 de 2016). *pmoinformatica*. Obtenido de pmoinformatica: <http://www.pmoinformatica.com/2016/08/tecnicas-analisis-requerimientos.html#:~:text=1.-%20Descomposición%20funcional&text=Aplicado%20a%20la%20Ingeniería%20de,un%20nivel%20adecuado%20de%20detalle.>
- pmoinformatica. (18 de 04 de 2018). *Documento de requerimientos de software*. Obtenido de Documento de requerimientos de software: http://www.pmoinformatica.com/2018/04/documento-de-requerimientos-de-software_37.html
- Poblete, B. S. (2017). *ORIGEN, TRAYECTORÍA Y EFECTIVIDAD DE LA FORMACIÓN DE ENSEÑANZA MEDIA TÉCNICO PROFESIONAL EN CHILE*. Obtenido de <https://www.scielo.br/j/heduc/a/pcBv9zdxhcrsPhFsQB6D5YD/?lang=es>
- qgis. (2018). *qgis*. Obtenido de qgis: [https://docs.qgis.org/3.28/es/docs/training_manual/database_concepts/db_intro.html#:~:text=Una%20clave%20primaria%20es%20uno,primaria%20de%20esa%20otra%20tabla\).](https://docs.qgis.org/3.28/es/docs/training_manual/database_concepts/db_intro.html#:~:text=Una%20clave%20primaria%20es%20uno,primaria%20de%20esa%20otra%20tabla).)
- Robert M. Fuhrer, B. L. (s.f.). *Algorithms for the Optimal State Assignment of Asynchronous State Machines*. Obtenido de Algorithms for the Optimal State Assignment of Asynchronous State Machines: https://es.wikipedia.org/wiki/Condición_de_carrera
- Rochina, P. (27 de 03 de 2017). *INESEM*. Obtenido de INESEM: <https://www.inesem.es/revistadigital/informatica-y-tics/teoria-grafos/#:~:text=Un%20grafo%2C%20es%20una%20estructura,o%20relaciones%20entre%20los%20actores>
- Santana, S. J. (2020). *sedici*. Obtenido de sedici: <http://sedici.unlp.edu.ar/handle/10915/103994#:~:text=La%20validación%20de%20requerimientos%20es,y%20expectativas%20de%20los%20usuarios>
- SAP. (s.f.). *SAP Business Technology Platform*. Obtenido de SAP Business Technology Platform: <https://www.sap.com/latinamerica/products/technology-platform/datasphere/what-is-data-modeling.html#:~:text=Los%20modelos%20de%20datos%20son,aplicaciones%20compartan%20los%20mismos%20datos.>
- Schoolnet, E. (2017). *European Schoolnet*. Obtenido de <http://www.eun.org/documents/411753/839549/Country+Report+Austria+2017.pdf/a86bf21c-6f90-4753-a3c1-fd715fe49ce0>
- Schoolnet, E. (Bruselas, 2015). *"Computing our future"*.

- SCHOOLS, C. P. (2021). *CHICAGO PUBLIC SCHOOLS*. Obtenido de <https://www.cps.edu/academics/career-and-technical-education/>
- Sena, M. (31 de 01 de 2019). *medium*. Obtenido de medium: [https://medium.com/techwomenc/estructuras-de-datos-a29062de5483#:~:text=Una%20estructura%20de%20datos"%20es,y%20cómo%20se%20pueden%20administrar](https://medium.com/techwomenc/estructuras-de-datos-a29062de5483#:~:text=Una%20estructura%20de%20datos)
- seobility. (s.f.). *seobility*. Obtenido de seobility: https://www.seobility.net/es/wiki/Lenguaje_de_marcado
- Serrano, S. (07 de 05 de 2022). *crehana*. Obtenido de crehana: <https://www.crehana.com/blog/transformacion-digital/que-es-array/>)
- SET Academy!* (s.f.). Obtenido de <https://web.usd475.org/school/jchs/set/SitePages/Home.aspx>
- socialmood. (2021). *socialmood*. Obtenido de socialmood: <https://www.40defiebre.com/que-es/diseño-responsivo>
- Soto, J. A. (29 de 06 de 2020). *geeknetic*. Obtenido de geeknetic: <https://www.geeknetic.es/Linux/que-es-y-para-que-sirve>
- Souza, B. (19 de 06 de 2019). *aluracursos*. Obtenido de aluracursos: <https://www.aluracursos.com/blog/diferencias-entre-biblioteca-framework-y-api#:~:text=Un%20framework%20es%20un%20conjunto,estructuras%2C%20patrones%20y%20componentes%20comunes>.
- Souza, I. d. (14 de 06 de 2019). *rockcontent*. Obtenido de rockcontent: <https://rockcontent.com/es/blog/que-es-un-servidor/>
- stibosystems. (2020). *stibosystems*. Obtenido de stibosystems: https://explore.stibosystems.com/c/que-es-la-gestion-de-datos-maestros-y-por-que-la-necesita?x=wfYasY&utm_campaign=gl-search-latam-ch&utm_source=ppc&utm_medium=google&utm_content=what-is-mdm-and-why-you-need-it&utm_term=software%20para%20gestion%20de%20da
- theastrology. (2022). *¿Qué es el diseño orientado a objetos (ood)?* Obtenido de *¿Qué es el diseño orientado a objetos (ood)?*: <https://es.theastrologypage.com/object-oriented-design>
- theastrology. (2022). *theastrology*. Obtenido de theastrology: <https://es.theastrologypage.com/endurance-testing>
- UACH. (s.f.). *METODOLOGIA UTILIZADA: EL PROCESO UNIFICADO*. Obtenido de *METODOLOGIA UTILIZADA: EL PROCESO UNIFICADO*: <http://cybertesis.uach.cl/tesis/uach/2003/bmficis718d/xhtml/TH.4.xml>
- Una nueva era de "educación en programación" Japón enumera la programación como una escuela primaria obligatoria.* (2018). Obtenido de <https://programmerclick.com/article/4129806262/>

unrein, j. (06 de 12 de 2018). *dev.to*. Obtenido de dev.to: <https://dev.to/thejessleigh/different-types-of-testing-explained-1ljo>

uoc. (2017). *Lenguajes máquina*. Obtenido de Lenguajes máquina:
http://cv.uoc.edu/moduls/XW02_79049_00373/web/main/m4/v2_3.html

Valero, B. (2021). *wiki*. Obtenido de wiki: https://es.wikipedia.org/wiki/Método_de_acceso

Velázquez, J. V. (2016). *DOCPLAYER*. Obtenido de DOCPLAYER: [https://docplayer.es/19501403-
Unidad-ii-metodologia-para-resolver-problemas-aplicando-la-poo-parte-1.html](https://docplayer.es/19501403-Unidad-ii-metodologia-para-resolver-problemas-aplicando-la-poo-parte-1.html)

Wrike. (2020). *Wrike*. Obtenido de Wrike: [https://www.wrike.com/es/project-management-
guide/faq/que-son-las-herramientas-de-gestion-de-proyectos/](https://www.wrike.com/es/project-management-guide/faq/que-son-las-herramientas-de-gestion-de-proyectos/)

Yañez, C. (16 de 05 de 2018). *ceac*. Obtenido de ceac: [https://www.ceac.es/blog/herramientas-de-
mapeo-objeto-relacional-orm#:~:text=El%20mapeo%20objeto-
relacional%20es,las%20bases%20de%20datos%20relacionales.](https://www.ceac.es/blog/herramientas-de-mapeo-objeto-relacional-orm#:~:text=El%20mapeo%20objeto-relacional%20es,las%20bases%20de%20datos%20relacionales.)

zapttest. (s.f.). *zapttest*. Obtenido de zapttest: [https://www.zapttest.com/es/que-es-la-comprobacion-
de-sistemas-una-inmersion-en-profundidad-en-enfoques-tipos-herramientas-consejos-y-
trucos-y-mucho-mas#:~:text=2.-
,Integración,software%20y%20si%20funcionan%20correctamente](https://www.zapttest.com/es/que-es-la-comprobacion-de-sistemas-una-inmersion-en-profundidad-en-enfoques-tipos-herramientas-consejos-y-trucos-y-mucho-mas#:~:text=2.-,Integración,software%20y%20si%20funcionan%20correctamente)