


ASIGNATURA DE METODOLOGÍAS PARA EL DESARROLLO DE PROYECTOS

1. Competencias	Construir soluciones de software y sistemas inteligentes mediante la gestión de proyectos, integración de metodologías, modelos y herramientas de desarrollo bajo la normatividad aplicable para la optimización de proyectos de investigación, innovación, desarrollo tecnológico y de emprendimiento.
2. Cuatrimestre	Séptimo
3. Horas Teóricas	15
4. Horas Prácticas	30
5. Horas Totales	45
6. Horas Totales por Semana Cuatrimestre	3
7. Objetivo de aprendizaje	El alumno gestionará proyectos de TI mediante metodologías de desarrollo para optimizar los recursos.

Unidades de Aprendizaje	Horas		
	Teóricas	Prácticas	Totales
I. Tipos de proyectos y normativa.	2	3	5
II. Metodologías de desarrollo.	10	20	30
III. Técnicas y modelos.	3	7	10
Totales	15	30	45


ELABORÓ:	Comité de Directores de la Carrera de Ingeniería en Desarrollo y Gestión de Software	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2020	

METODOLOGÍAS PARA EL DESARROLLO DE PROYECTOS

UNIDADES DE APRENDIZAJE

1. Unidad de aprendizaje	I. Tipos de proyectos y normativa.
2. Horas Teóricas	2
3. Horas Prácticas	3
4. Horas Totales	5
5. Objetivo de la Unidad de Aprendizaje	El alumno justificará por medio del marco regulatorio las normas y estándares para aplicarlas al tipo de proyecto de TI.


Temas	Saber	Saber hacer	Ser
Tipos de proyectos y en qué consisten.	Identificar los conceptos básicos de proyectos de TI. Identificar los tipos de proyectos: Investigación, innovación, desarrollo tecnológico y emprendimiento.	Establecer el tipo de proyecto.	Sistemático Analítico Ético Liderazgo Responsabilidad Toma de decisión
Marco regulatorio en el desarrollo de software.	Identificar el marco regulatorio en el desarrollo de software.	Aplicar el marco regulatorio en el desarrollo de software.	Sistemático Analítico Ético Liderazgo Responsabilidad Toma de decisión
Normas y estándares de TI de acuerdo al proyecto	Identificar las normas y estándares aplicables a TI.	Justificar el uso de normas y estándares aplicables a TI.	Sistemático Analítico Ético Liderazgo Responsabilidad Toma de decisión

ELABORÓ:	Comité de Directores de la Carrera de Ingeniería en Desarrollo y Gestión de Software	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2020	

METODOLOGÍAS PARA EL DESARROLLO DE PROYECTOS

PROCESO DE EVALUACIÓN

Resultado de aprendizaje	Secuencia de aprendizaje	Instrumentos y tipos de reactivos
Entrega reporte con base en un caso de estudio que incluya: <ul style="list-style-type: none">- Justificación del tipo de proyecto.- Título del proyecto.- Objetivo general y específicos del proyecto.- Justificación de normas y estándares de acuerdo al marco regulatorio.	<ol style="list-style-type: none">1. Identificar los tipos de proyectos.2. Comprender el marco regulatorio que aplica a los proyectos de TI.3. Identificar las normas y estándares aplicables a proyectos de TI.	<ul style="list-style-type: none">- Estudio de Casos.- Rúbrica

ELABORÓ:	Comité de Directores de la Carrera de Ingeniería en Desarrollo y Gestión de Software	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2020	


METODOLOGÍAS PARA EL DESARROLLO DE PROYECTOS

PROCESO ENSEÑANZA APRENDIZAJE

Métodos y técnicas de enseñanza	Medios y materiales didácticos
<ul style="list-style-type: none">- Práctica demostrativa.- Análisis de casos.- Equipos colaborativos.	<ul style="list-style-type: none">- Pintarrón.- Plumones.- Equipo audiovisual.- Equipo de cómputo.- Internet.- Repositorios.

ESPACIO FORMATIVO

Aula	Laboratorio / Taller	Empresa
X		


ELABORÓ:	Comité de Directores de la Carrera de Ingeniería en Desarrollo y Gestión de Software	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2020	

METODOLOGÍAS PARA EL DESARROLLO DE PROYECTOS


UNIDADES DE APRENDIZAJE

1. Unidad de aprendizaje	II. Metodologías de desarrollo.
2. Horas Teóricas	10
3. Horas Prácticas	20
4. Horas Totales	30
5. Objetivo de la Unidad de Aprendizaje	El alumno implementará las metodologías de desarrollo para gestionar el proyecto de TI.

Temas	Saber	Saber hacer	Ser
Metodologías de investigación	<p>Conocer los tipos de metodologías de investigación.</p> <p>Identificar los métodos y técnicas de investigación cuantitativas, cualitativas y mixtas.</p>	<p>Determinar métodos, técnicas e instrumentos aplicables al proyecto.</p> <p>Elaborar protocolos e informes técnicos de la investigación.</p>	<p>Responsabilidad</p> <p>Trabajo en equipo</p> <p>Liderazgo</p> <p>Toma de decisión</p> <p>Sentido de la planificación</p> <p>Trabajo bajo presión</p> <p>Razonamiento inductivo</p> <p>Razonamiento deductivo</p> <p>Razonamiento hipotético</p> <p>Proactivo</p>
Metodologías tradicionales de desarrollo	<p>Identificar las características, ventajas y desventajas de las metodologías tradicionales de desarrollo: CMMI, TSP, PSP.</p>	<p>Determina la metodología tradicional de desarrollo a utilizar.</p> <p>Justificar el uso de la metodología tradicional de desarrollo a utilizar.</p>	<p>Responsabilidad</p> <p>Trabajo en equipo</p> <p>Liderazgo</p> <p>Toma de decisión</p> <p>Sentido de la planificación</p> <p>Trabajo bajo presión</p> <p>Razonamiento inductivo</p> <p>Razonamiento deductivo</p> <p>Razonamiento hipotético</p> <p>Proactivo</p>

ELABORÓ:	Comité de Directores de la Carrera de Ingeniería en Desarrollo y Gestión de Software	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2020	


Metodologías ágiles de desarrollo.	Identificar las características, ventajas y desventajas de las metodologías ágiles de desarrollo: Scrum, XP, Kanban.	Determina la metodología ágil de desarrollo a utilizar. Justificar el uso de la metodología ágil de desarrollo a utilizar.	Responsabilidad Trabajo en equipo Liderazgo Toma de decisión Sentido de la planificación Trabajo bajo presión Razonamiento inductivo Razonamiento deductivo Razonamiento hipotético Proactivo
Metodologías de desarrollo de proyectos.	Identificar las características, ventajas y desventajas de las metodologías de desarrollo de proyectos: PMBok, Lean Startup.	Determina la metodología de desarrollo de proyectos a utilizar. Justificar el uso de la metodología de desarrollo de proyectos a utilizar.	Responsabilidad Trabajo en equipo Liderazgo Toma de decisión Sentido de la planificación Trabajo bajo presión Razonamiento inductivo Razonamiento deductivo Razonamiento hipotético Proactivo
Metodologías de emprendimiento.	Identificar las características, ventajas y desventajas de las metodologías de emprendimiento de acuerdo al proyecto de TI.	Determina la metodología de emprendimiento a utilizar. Justificar el uso de la metodología de emprendimiento a utilizar.	Responsabilidad Trabajo en equipo Liderazgo Toma de decisión Sentido de la planificación Trabajo bajo presión Razonamiento inductivo Razonamiento deductivo Razonamiento hipotético Proactivo

ELABORÓ:	Comité de Directores de la Carrera de Ingeniería en Desarrollo y Gestión de Software	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2020	

METODOLOGÍAS PARA EL DESARROLLO DE PROYECTOS

PROCESO DE EVALUACIÓN

Resultado de aprendizaje	Secuencia de aprendizaje	Instrumentos y tipos de reactivos
<p>Entrega un reporte a partir de un caso de estudio que incluya:</p> <ul style="list-style-type: none"> - Justificación del uso de la metodología de acuerdo al tipo de proyecto a desarrollar. - Requerimientos del proyecto. - Cronograma de actividades. - Lista de herramientas para aplicar la metodología. - Recursos materiales y humanos que se utilizarán en el proyecto. - Matriz de roles y responsabilidades. - Matriz de riesgos. 	<ol style="list-style-type: none"> 1. Identificar las metodologías para el desarrollo de proyectos. 2. Determinar métodos, técnicas e instrumentos de la metodología a utilizar. 3. Justificar el uso de la metodología a utilizar de acuerdo al proyecto. 4. Definir las herramientas técnicas para el desarrollo del proyecto. 	<ul style="list-style-type: none"> - Estudio de Casos. - Rúbrica

ELABORÓ:	Comité de Directores de la Carrera de Ingeniería en Desarrollo y Gestión de Software	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2020	


METODOLOGÍAS PARA EL DESARROLLO DE PROYECTOS

PROCESO ENSEÑANZA APRENDIZAJE

Métodos y técnicas de enseñanza	Medios y materiales didácticos
<ul style="list-style-type: none">- Práctica demostrativa.- Análisis de casos.- Equipos colaborativos.	<ul style="list-style-type: none">- Pintarrón.- Plumones.- Equipo audiovisual.- Equipo de cómputo.- Internet.- Repositorios.- Plataformas virtuales.

ESPACIO FORMATIVO

Aula	Laboratorio / Taller	Empresa
X		


ELABORÓ:	Comité de Directores de la Carrera de Ingeniería en Desarrollo y Gestión de Software	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2020	

METODOLOGÍAS PARA EL DESARROLLO DE PROYECTOS


UNIDADES DE APRENDIZAJE

1. Unidad de aprendizaje	III. Técnicas y modelos.
2. Horas Teóricas	3
3. Horas Prácticas	7
4. Horas Totales	10
5. Objetivo de la Unidad de Aprendizaje	El alumno ejecutará el modelo de negocio CANVAS para el seguimiento de un proyecto de TI.

Temas	Saber	Saber hacer	Ser
Modelos de negocios.	Identificar las características del modelo CANVAS.	Aplicar el modelo CANVAS.	Responsabilidad Trabajo en equipo Liderazgo Toma de decisión Sentido de la planificación Trabajo bajo presión Razonamiento inductivo Razonamiento deductivo Razonamiento hipotético Proactivo
Técnicas de definición de requerimientos.	Identificar las técnicas de definición de requerimientos del proyecto.	Documentar los requerimientos del sistema.	Responsabilidad Trabajo en equipo Liderazgo Toma de decisión Sentido de la planificación Trabajo bajo presión Razonamiento inductivo Razonamiento deductivo Razonamiento hipotético Proactivo

ELABORÓ:	Comité de Directores de la Carrera de Ingeniería en Desarrollo y Gestión de Software	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2020	


Técnicas de documentación	Identificar las técnicas de documentación de proyectos de TI.	Realizar la documentación de proyectos de TI.	Responsabilidad Trabajo en equipo Liderazgo Toma de decisión Sentido de la planificación Trabajo bajo presión Razonamiento inductivo Razonamiento deductivo Razonamiento hipotético Proactivo
Técnicas de estimación de trabajo y costos.	Identificar las técnicas de estimación de trabajo y costos: Recursos humanos y materiales.	Diagnosticar de acuerdo al alcance del proyecto los recursos humanos y materiales a utilizar.	Responsabilidad Trabajo en equipo Liderazgo Toma de decisión Sentido de la planificación Trabajo bajo presión Razonamiento inductivo Razonamiento deductivo Razonamiento hipotético Proactivo

ELABORÓ:	Comité de Directores de la Carrera de Ingeniería en Desarrollo y Gestión de Software	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2020	

METODOLOGÍAS PARA EL DESARROLLO DE PROYECTOS

PROCESO DE EVALUACIÓN

Resultado de aprendizaje	Secuencia de aprendizaje	Instrumentos y tipos de reactivos
Entrega un reporte a partir de un caso de estudio que incluya: <ul style="list-style-type: none">- Modelo de negocio CANVAS.- Requerimientos del sistema.- Planificación del proyecto.- Tablero CANVAS.- Recursos humanos y materiales del proyecto.	<ol style="list-style-type: none">1. Conocer el modelo de negocio CANVAS.2. Analizar las etapas del modelo de negocio CANVAS.3. Establecer requerimientos del proyecto.4. Establecer técnicas de documentación y seguimiento del modelo de negocio CANVAS.5. Establecer los recursos humanos y materiales del proyecto.	<ul style="list-style-type: none">- Estudio de Casos.- Rúbrica

ELABORÓ:	Comité de Directores de la Carrera de Ingeniería en Desarrollo y Gestión de Software	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2020	


METODOLOGÍAS PARA EL DESARROLLO DE PROYECTOS

PROCESO ENSEÑANZA APRENDIZAJE

Métodos y técnicas de enseñanza	Medios y materiales didácticos
<ul style="list-style-type: none">- Práctica demostrativa.- Análisis de casos.- Equipos colaborativos.	<ul style="list-style-type: none">- Pintarrón.- Plumones.- Equipo audiovisual.- Equipo de cómputo.- Internet.- Repositorios.- Plataformas virtuales.- Tablero CANVAS

ESPACIO FORMATIVO


Aula	Laboratorio / Taller	Empresa
X		

ELABORÓ:	Comité de Directores de la Carrera de Ingeniería en Desarrollo y Gestión de Software	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2020	


METODOLOGÍAS PARA EL DESARROLLO DE PROYECTOS

CAPACIDADES DERIVADAS DE LAS COMPETENCIAS PROFESIONALES A LAS QUE CONTRIBUYE LA ASIGNATURA


Capacidad	Criterios de Desempeño
Determinar el tipo de proyecto de software mediante el análisis de un problema para determinar la forma de desarrollo.	Entrega un documento de definición del proyecto que incluya: <ul style="list-style-type: none"> - Descripción de la problemática. - Objetivos establecidos. - Metas y alcances. - Requerimientos. - Justificación del tipo de proyecto.
Establecer metodologías y herramientas de gestión con base en el tipo y características del proyecto identificando las normas, estándares, leyes y regulaciones aplicables para el cumplimiento de los requerimientos establecidos.	Entrega un documento que incluya: <ul style="list-style-type: none"> - Justificación de la metodología seleccionada. - Justificación de las herramientas de gestión. - Listado de las normas, estándares, leyes y regulaciones aplicables.
Implementar el plan de trabajo del proyecto mediante la metodología y herramientas establecidas incorporando normas y estándares aplicables para la optimización de recursos.	Entrega un plan de trabajo del proyecto que incluye: <ul style="list-style-type: none"> - Descripción de actividades. - Análisis de recursos. - Descripción de roles y responsabilidades. - Definición de etapas y tiempos. - Descripción de los riesgos identificados.
Controlar la ejecución del proyecto mediante el seguimiento y evaluación del plan de trabajo para asegurar el cumplimiento de los objetivos establecidos y de las normas, estándares, leyes y regulaciones aplicables.	Entrega un documento que incluya: <ul style="list-style-type: none"> - Bitácoras. - Reportes de seguimiento. - Plan de gestión del proyecto.
Formalizar el cierre del proyecto a través del cumplimiento de los criterios y protocolos establecidos en el plan de trabajo para el despliegue óptimo de los entregables definidos.	1. Entrega un documento que incluya: <ul style="list-style-type: none"> - Acta de entrega y liberación del proyecto. - Acta de cierre del proyecto. - Plan de seguimiento y mantenimiento al proyecto. 2. Entrega los productos generados del proyecto.
Determinar procesos y herramientas de machine learning, data mining y big data mediante el análisis del problema de acuerdo a las características, ubicación de los datos y normativa aplicable para establecer un plan de desarrollo e integración que cubra las necesidades de información.	Entrega un plan de desarrollo e integración que incluya: <ul style="list-style-type: none"> - Planteamiento del problema. - Definición de necesidades de información. - Descripción de la ubicación y características de los datos. - Justificación de la normativa aplicable. - Justificación de las herramientas y procesos a utilizar.

ELABORÓ:	Comité de Directores de la Carrera de Ingeniería en Desarrollo y Gestión de Software	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2020	


<p>Determinar metodologías y herramientas de colaboración mediante el análisis del problema de acuerdo a las normas y estándares aplicables para una administración eficiente del proceso de desarrollo de software bajo un esquema de integración continua.</p>	<p>Entrega un documento que incluya:</p> <ul style="list-style-type: none"> - Objetivos del desarrollo software. - Metas y alcances del desarrollo de software. - Requerimientos funcionales y no funcionales del desarrollo de software. - Justificación de la metodología seleccionada. - Justificación de las herramientas para el proceso de comunicación, desarrollo, integración continua y monitoreo del software. - Listado de las normas, estándares, leyes y regulaciones aplicables al proceso de desarrollo de software del proyecto a realizar.
<p>Implementar planes de desarrollo y pruebas mediante la metodología y herramientas definidas, incorporando normas y estándares aplicables para atender los requerimientos establecidos.</p>	<p>1. Entrega un plan de trabajo que incluya:</p> <ul style="list-style-type: none"> - Cronograma de actividades. - Plan de entregas continuas. - Roles y responsabilidades. - Requisitos del entorno de desarrollo y producción. - Definición de manuales a entregar. <p>2. Entrega el plan de pruebas, de acuerdo a la metodología seleccionada.</p> <ul style="list-style-type: none"> - Tipos de pruebas. - Justificación de las herramientas de pruebas. - Criterios de aceptación y liberación. - Cronograma de pruebas. - Instrumentos de pruebas. - Requisitos del entorno de pruebas.
<p>Gestionar el desarrollo de software mediante el seguimiento de la codificación y ejecución de pruebas para la integración continua del producto.</p>	<p>1. Entrega los archivos de configuración del entorno de producción.</p> <p>2. Entrega un documento que incluya:</p> <ul style="list-style-type: none"> - Bitácoras del desarrollo de software. - Reportes de seguimiento del desarrollo de software. - Evidencias de control de cambios y versiones del código. - Resultado de pruebas.
<p>Realizar la liberación de unidades estables de software mediante un plan de entregas continuas para contribuir al cumplimiento de los objetivos del proyecto.</p>	<p>1. Entrega los resultados de las pruebas de sistema en entorno de operación.</p> <p>2. Entrega un documento que incluya:</p> <ul style="list-style-type: none"> - Evidencias del seguimiento de la ejecución del plan de entregas continuas. - Evidencia de acuerdo de aceptación. <p>3. Entrega archivos que incluya:</p> <ul style="list-style-type: none"> - Unidades estables de software a producción.

ELABORÓ:	Comité de Directores de la Carrera de Ingeniería en Desarrollo y Gestión de Software	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2020	

<p>Establecer planes de mantenimiento mediante la comunicación constante con el cliente e identificación de puntos críticos y áreas de oportunidad para la mejora continua del sistema de información.</p>	<p>Entrega un documento que incluya:</p> <ul style="list-style-type: none"> - Métricas de desempeño del software en el ambiente de producción. - Actividades de monitoreo del software en el ambiente de producción. - Bitácora de mantenimiento del software.
<p>Determinar la arquitectura de software mediante el análisis de los requerimientos, la definición de los componentes y la relación entre ellos para satisfacer los atributos de calidad del software y servir como guía en el desarrollo</p>	<p>Entrega un documento de arquitectura de software que incluya:</p> <ul style="list-style-type: none"> - Requerimientos funcionales y no funcionales de desarrollo de software. - Justificación de la arquitectura seleccionada. - Diagrama de arquitectura de software. - Características y funciones de los componentes. - Diagramas necesarios de acuerdo a la arquitectura seleccionada.
<p>Diseñar interfaces de usuario mediante el uso de herramientas y principios de usabilidad para optimizar la experiencia del usuario</p>	<p>1. Entrega un documento que incluya:</p> <ul style="list-style-type: none"> - Justificación de las herramientas para el diseño de las interfaces. - Maquetado de interfaces. - Modelos de navegación. - Guía de estilo de interfaces. <p>2. Entrega archivos de las interfaces.</p>
<p>Implementar modelos de datos mediante herramientas y sistemas gestores para garantizar la disponibilidad e integridad de la información</p>	<p>1. Entrega un documento que incluya:</p> <ul style="list-style-type: none"> - Justificación de las herramientas a utilizar para los modelos de datos. - Modelos de datos. - Descripción de metadatos. <p>2. Entrega archivos, credenciales de registro y secuencia de configuración para la creación de los modelos de datos.</p>
<p>Desarrollar componentes mediante el uso de patrones de diseño, APIs y frameworks de desarrollo conforme a la arquitectura establecida para la integración continua de unidades del software</p>	<p>1. Entrega un documento que incluya:</p> <ul style="list-style-type: none"> - Justificación de las herramientas para el desarrollo de componentes. - Justificación de los patrones de diseño a utilizar. - Justificación de los lenguajes de programación a utilizar. <p>2. Entrega archivos de código fuente y configuración de los componentes desarrollados.</p> <p>3. Entrega evidencia de almacenamiento de archivos de código fuente y actualizaciones en la documentación del proyecto en la plataforma de versionamiento seleccionada.</p>

ELABORÓ:	Comité de Directores de la Carrera de Ingeniería en Desarrollo y Gestión de Software	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2020	


<p>Ejecutar pruebas a componentes de software a través del uso de técnicas y herramientas de testing para asegurar su correcto funcionamiento</p>	<p>1. Entrega un documento que incluya:</p> <ul style="list-style-type: none"> - Justificación de las técnicas y herramientas utilizadas. - Descripción de las pruebas. - Informe de resultados del plan de pruebas. <p>2. Entrega archivos de código fuente y configuración del componente.</p>
<p>Implementar soluciones de software mediante la integración de componentes de acuerdo a la arquitectura definida para su liberación en un ambiente de producción</p>	<p>1. Entrega manuales del software establecidos en el plan de desarrollo.</p> <p>2. Entrega la versión estable del software integrando los componentes de acuerdo al plan de entregas continuas.</p> <p>3. Entrega evidencia de almacenamiento de archivos de código fuente y actualizaciones de la documentación del proyecto en la plataforma de versionamiento establecida.</p>

ELABORÓ:	Comité de Directores de la Carrera de Ingeniería en Desarrollo y Gestión de Software	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2020	


METODOLOGÍAS PARA EL DESARROLLO DE PROYECTOS

FUENTES BIBLIOGRÁFICAS

Autor	Año	Título del Documento	Ciudad	País	Editorial
Hernandez Sampieri, Christian Paulina Mendoza Torres	2018 ISBN: 978-1-4562-6096-5	Metodología De La Investigación: Las rutas cuantitativa, cualitativa y mixta	Ciudad de México	México	Mc Graw Hill
Sutherland, Jeff	2016 ISBN: 978-607-735-559-5	Scrum: El arte de hacer el doble de trabajo en la mitad del tiempo		Estados Unidos	Oceano
Marbaise, Magali	2017 ISBN de origen: 2806291089	<i>El modelo CANVAS</i>		Francia	50Minutos.es
Project Managment Institute, Inc.	2017 ISBN: 978-1-62825-194-4	<i>Guía de los Fundamentos Para la Dirección de Proyectos: Guía del Pmbok</i>	Pennsylvania	Estados Unidos	Project Managment Institute, Inc.
Edge, James	2018 ISBN: 978-1-7296-4234-4	<i>Lean: The Ultimate Guide to Lean Startup, Lean Six Sigma, Lean Analytics, Lean Enterprise, Lean Manufacturing, Scrum, Agile Project Management and Kanban</i>		Estados Unidos	
Ake Knapp, John Zeratsky, Braden Kowitz	2017 ISBN-10: 6073155239	<i>Sprint: El método para resolver problemas y probar nuevas ideas en solo cinco días</i>		Estados Unidos	CONNECTA
González Torres Antonio	2016 ISBN: 978-9977-37-006-4	<i>Tecnologías de la Información y Gestión de Proyectos</i>	San José	Costa Rica	Universidad Latinoamericana de Ciencia y Tecnología

ELABORÓ:	Comité de Directores de la Carrera de Ingeniería en Desarrollo y Gestión de Software	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2020	

Gothelf, Jeff	2017 ISBN: 13:978- 1547226764	<i>Lean vs Agile vs Design Thinking</i>	New Jersey	Estados Unidos	Gothelf.co
---------------	--	---	------------	-------------------	------------

ELABORÓ:	Comité de Directores de la Carrera de Ingeniería en Desarrollo y Gestión de Software	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2020	