

Máster Título Propio

Ingeniería del Software Avanzada

```
back the deselected mirror modifier object
```

```
ects.active = modifier_ob  
str(modifier_ob)) # modifier ob is the active ob
```



Máster Título Propio Ingeniería del Software Avanzada

- » Modalidad: online
- » Duración: 12 meses
- » Titulación: TECH Global University
- » Acreditación: 60 ECTS
- » Horario: a tu ritmo
- » Exámenes: online

Acceso web: www.techtitute.com/informatica/master/master-ingenieria-software-avanzada

Índice

01

Presentación

pág. 4

02

Objetivos

pág. 8

03

Competencias

pág. 14

04

Estructura y contenido

pág. 18

05

Metodología

pág. 36

06

Titulación

pág. 44

01

Presentación

El desarrollo de programas informáticos no sería posible sin la existencia de la Ingeniería de Softwares. Gracias a los avances que se han realizado en materia tecnológica, hoy en día es posible encontrar sistemas y estructuras cada vez más complejas y específicas, diseñadas en base a las especificaciones de la programación con el objetivo de cumplir las necesidades y exigencias del mercado. Se trata de un proceso metódico y ordenado que requiere de un conocimiento especializado sobre las ciencias de la computación y sus herramientas, algo que el egresado podrá adquirir con este completísimo programa. A través de una experiencia multidisciplinar, se sumergirá en las claves de los procesos informáticos, así como en la integración de sistemas y adquirirá las habilidades necesarias para llevar a cabo proyectos del máximo nivel en base a los requerimientos de los *Stakeholders*. Todo ello, en tan solo 12 meses de capacitación 100% Online.





“

Si lo que buscas es un programa que te guíe para emprender tu propio proyecto de software desde 0, este Máster Título Propio es perfecto para ti. ¿A qué esperas para matricularte?”

Durante más de 6 décadas, la Ingeniería del Software ha protagonizado la revolución tecnológica a través del desarrollo de programas y aplicaciones cada vez más complejos y especializados. Se trata de un área que ha servido de apoyo a otras muchas para avanzar hacia el progreso y cuya aplicación se extrapola a prácticamente todas las especialidades que existen: la Medicina, la agricultura, la docencia, la administración, la industria, etc. Y es que se mire hacia donde se mire, hasta el proceso informático más sencillo como puede ser enviar un correo electrónico o hacer uso de la mensajería instantánea, algo francamente cotidiano en la actualidad, ha requerido de un diseño y de una programación exhaustiva para lograr su fin: satisfacer las necesidades del ser humano.

El amplio abanico de oportunidades que surge de esta ciencia y sus múltiples aplicaciones la sitúan como una de las más demandadas en el mercado laboral, no solo para crear nuevos proyectos, sino para realizar las labores de supervisión, mantenimiento y actualización de los ya existentes. Por esa razón, y siguiendo la máxima de TECH de ofrecer a todos sus egresados la posibilidad de especializarse en este campo, la universidad ha decidido lanzar este completísimo Máster Título Propio en Ingeniería del Software Avanzada.

Se trata de una experiencia académica que incluye 1.800 horas del mejor contenido teórico-práctico y adicional, abarcando la totalidad de este campo, desde el origen hasta el diseño, creación y gestión de sistemas de información innovadores y modernos. A lo largo de los 12 meses en los que se desarrolla el programa, el informático podrá sumergirse en los entresijos de esta especialidad: sus requisitos técnicos y estructurales, las claves para crear arquitecturas seguras, la integración de servicios basados en las TIC, la gestión de los *Stakeholders* y su alcance, el desarrollo de un proyecto desde la base hasta su lanzamiento ¡y muchas cosas más!

Todo ello de manera 100% Online, gracias a lo cual el egresado podrá acceder al curso de este Máster Título Propio cuando quiera y desde donde quiera, sin clases presenciales ni horarios encorsetados. Además, podrá entrar al campus virtual desde cualquier dispositivo con conexión a internet, ya sea PC, *Tablet* o móvil. Es, por lo tanto, una oportunidad única de especializarse en la Ingeniería del Software a través de una titulación adaptada a sus necesidades académicas y a los requisitos más exigentes de la industria Informática actual.

Este **Máster Título Propio en Ingeniería del Software Avanzada** contiene el programa educativo más completo y actualizado del mercado. Las características más destacadas son:

- ◆ El desarrollo de casos prácticos presentados por expertos en Ingeniería Informática
- ◆ Los contenidos gráficos, esquemáticos y eminentemente prácticos con los que está concebido recogen una información práctica sobre aquellas disciplinas indispensables para el ejercicio profesional
- ◆ Los ejercicios prácticos donde realizar el proceso de autoevaluación para mejorar el aprendizaje
- ◆ Su especial hincapié en metodologías innovadoras
- ◆ Las lecciones teóricas, preguntas al experto, foros de discusión de temas controvertidos y trabajos de reflexión individual
- ◆ La disponibilidad de acceso a los contenidos desde cualquier dispositivo fijo o portátil con conexión a internet



Una titulación que ahonda en las bases de la Ingeniería del Software: desde sus orígenes, hasta los procesos de la computación Informática que se llevan a cabo hoy en día”

“

¿Te gustaría adquirir las habilidades necesarias para diseñar arquitecturas complejas y alternativas a través del flujo de datos? Con este Máster Título Propio lo conseguirás en menos de 12 meses”

Incluye en su cuadro docente a un equipo de profesionales del sector que vierten en esta capacitación la experiencia de su trabajo, además de reconocidos especialistas de sociedades de referencia y universidades de prestigio.

Su contenido multimedia, elaborado con la última tecnología educativa, permitirá al profesional un aprendizaje situado y contextual, es decir, un entorno simulado que proporcionará una capacitación inmersiva programada para entrenarse ante situaciones reales.

El diseño de este programa se centra en el Aprendizaje Basado en Problemas, mediante el cual el profesional deberá tratar de resolver las distintas situaciones de práctica profesional que se le planteen a lo largo del curso académico. Para ello, contará con la ayuda de un novedoso sistema de vídeos interactivos realizados por reconocidos expertos.

Un programa diseñado para que puedas aplicar las estrategias más innovadoras en modelado de requisitos desde el primer momento.

Tendrás acceso a una guía práctica que recoge los principios de la Ingeniería del Software, desde el proceso inicial hasta la construcción y el despliegue.



02 Objetivos

La Ingeniería de Softwares se ha convertido en una de las principales herramientas en el desarrollo tecnológico actual. Y es que sin el trabajo que desempeñan los millones de profesionales de la Informática no hubiese sido posible alcanzar el altísimo nivel de innovación digital que se conoce hoy en día. Por esa razón, el objetivo de este Máster Título Propio es aportar a los egresados interesados en esta área toda la información que necesitan para conocerla al detalle y poder desarrollar sus propios programas y aplicaciones de manera exitosa, eficaz y en base a la máxima calidad y rigurosidad.





“

Si entre tus objetivos con el curso de este Máster Título Propio está el dominar Scrum y otras técnicas de metodologías ágiles, TECH pondrá a tu disposición todo lo que necesitas para superarlos de manera garantizada”



Objetivos generales

- ◆ Capacitar científica y tecnológicamente, así como prepararse para el ejercicio profesional de la Ingeniería del Software, todo ello con una experiencia académica transversal y versátil adaptada a las nuevas tecnologías e innovaciones en este campo
- ◆ Obtener amplios conocimientos en el campo de la Ingeniería del Software, pero también en el campo de la computación y la estructura de ordenadores, todo ello incluyendo la base matemática, estadística y física imprescindible en una Ingeniería

“

Trabajarás intensamente en las pruebas de softwares a través de TDD, ATDD y BDD, para que seas capaz de construir estructuras informáticas de la mayor calidad”





Objetivos específicos

Módulo 1. Ingeniería del Software

- ◆ Sentar las bases de la Ingeniería del Software y el modelado, aprendiendo los principales procesos y conceptos
- ◆ Entender el proceso del software y los distintos modelos para su desarrollo, incluyendo tecnologías ágiles
- ◆ Conocer las principales normas relativas a la calidad del Software y a la administración de proyectos

Módulo 2. Ingeniería del Software Avanzada

- ◆ Conocer en profundidad las distintas metodologías ágiles usadas en la Ingeniería del software
- ◆ Aprender a desarrollar usando las técnicas de *Scrum*, programación extrema y de desarrollo de software basado en reutilización
- ◆ Entender los conceptos y procesos del diseño de software, aprendiendo también sobre el diseño de la arquitectura y sobre el diseño a nivel de componentes y basado en patrones
- ◆ Introducir el concepto de DevOps y cuáles son sus principales prácticas
- ◆ Aprender a realizar pruebas al software, con metodologías como *Test Driven Development*, *Acceptance Test Driven Development*, *Behavior Driven Development*, BDD y *Cucumber*
- ◆ Comprender los distintos patrones de arquitecturas de sistemas y de diseño de software, así como la arquitectura de las aplicaciones en la nube

Módulo 3. Ingeniería de requisitos

- ◆ Comprender la Ingeniería de requisitos, su desarrollo, elaboración, negociación y validación
- ◆ Aprender el modelado de los requisitos y de los distintos elementos como escenarios, información, clases de análisis, flujo, comportamiento y patrones
- ◆ Comprender la importancia de la Ingeniería de requisitos en el proceso de desarrollo de software
- ◆ Aprender a realizar análisis de requisitos, así como a documentar adecuadamente los mismos
- ◆ Profundizar en las fuentes de requisitos y las técnicas de elicitación de requisitos, ya que son parte esencial del proceso
- ◆ Entender los procesos de validación y negociación de requisitos, así como el modelado y gestión de requisitos
- ◆ Adquirir los conocimientos necesarios para el manejo de sistemas críticos y la especificación formal de los requisitos

Módulo 4. Procesos de Ingeniería del Software

- ◆ Profundizar en la mejora del proceso de desarrollo de software y de calidad del software usando los estándares ISO/IEC
- ◆ Entender y aplicar la realización de prototipos como parte esencial del proceso de desarrollo
- ◆ Conocer el marco de referencia de la Ingeniería del Software y la norma ISO/IEC 12207
- ◆ Aprender las características del proceso unificado de desarrollo de software y la planificación en el contexto del desarrollo de software ágil
- ◆ Conocer los distintos estilos de diseño de software distribuido y arquitecturas software orientadas a servicios
- ◆ Aprender los conceptos esenciales en el diseño de interfaces gráficas de usuario
- ◆ Comprender las bases del desarrollo de aplicaciones web

Módulo 5. Calidad y auditoría de sistemas de información

- ◆ Profundizar en las estrategias y técnicas de pruebas software, factores de calidad del software y distintas métricas utilizadas
- ◆ Adquirir los conocimientos esenciales de los sistemas de gestión de seguridad de la Informática
- ◆ Introducir los conceptos de propiedad intelectual en los sistemas de gestión de la información
- ◆ Preparar al alumnado en la creación de planes de continuidad de negocio y de recuperación frente a desastres
- ◆ Aprender a planificar la gestión de la seguridad y a manejar los principales mecanismos para la protección de activos información
- ◆ Conocer los distintos tipos de auditorías y cuál es el proceso llevado a cabo durante la auditoría Informática

Módulo 6. Integración de sistemas

- ◆ Adquirir los conceptos esenciales relacionados con los sistemas de información en la empresa, así como identificar las oportunidades y necesidades de los sistemas de información en la empresa
- ◆ Conocer las bases del *Business Intelligence*, sus estrategias e implantación, así como el presente y futuro del BI
- ◆ Comprender el funcionamiento de los sistemas para la gestión integrada de recursos de la empresa
- ◆ Entender la transformación digital, desde el punto de vista de la innovación empresarial, la gestión financiera y de la producción, el Marketing y la gestión de Recursos Humanos

Módulo 7. Reutilización de software

- ◆ Conocer el panorama general en la estrategia de la reutilización de software
- ◆ Aprender los distintos patrones relacionados con la reutilización de software, tanto de diseño como de creación, estructurales y de comportamiento
- ◆ Introducir el concepto de *Framework*, así como conocer los principales tipos como los destinados al diseño de interfaces gráficas de usuario, al desarrollo de aplicaciones web y a la gestión de la persistencia de objetos en bases de datos
- ◆ Comprender el funcionamiento del patrón ampliamente usado actualmente de Modelo Vista Controlador (MVC)

Módulo 8. Servicios de tecnología de la información

- ◆ Capacitar para la toma de decisiones de inversión en TIC y la planificación de sistemas de información
- ◆ Conocer los Objetivos de Control para la Información y Tecnologías Relacionadas (COBIT)
- ◆ Aprender el funcionamiento de la Biblioteca de Infraestructura de Tecnologías de Información (ITIL), las estrategias, el diseño de servicios, transiciones y operaciones
- ◆ Profundizar en el sistema de gestión de servicios, conociendo los principios básicos de UNE-ISO/IEC 20000-1, la estructura de la serie de normas ISO/IEC 20000 y los requisitos del Sistema de Gestión del Servicio (SGS)
- ◆ Entender el funcionamiento de los sistemas y tecnologías de la información, sus componentes, clasificaciones, arquitecturas y formas de integración de sistemas
- ◆ Aprender el estándar ISO/IEC 12207, el análisis, diseño, implantación y aceptación de sistemas de información

Módulo 9. Seguridad en los sistemas de información

- ◆ Aprender el desarrollo del cronograma para la gestión del tiempo, el desarrollo del presupuesto y la respuesta ante los riesgos
- ◆ Analizar la naturaleza de los ataques en redes y los distintos tipos de arquitecturas de seguridad
- ◆ Comprender las distintas técnicas de protección de sistemas y de desarrollo de código seguro
- ◆ Conocer los componentes esenciales de *Botnets* y *Spam*, así como del *Malware* y del código malicioso
- ◆ Sentar las bases para el análisis forense en el mundo del software y de las auditorías informáticas
- ◆ Obtener una perspectiva global de la seguridad, la criptografía y los criptoanálisis clásicos
- ◆ Comprender los fundamentos de la criptografía simétrica y de la criptografía asimétrica, así como sus principales algoritmos

Módulo 10. Gestión de proyectos

- ◆ Comprender el funcionamiento de la gestión de la calidad en los proyectos, incluyendo la planificación, el aseguramiento, el control, los conceptos estadísticos y las herramientas disponibles
- ◆ Entender el funcionamiento de los procesos de aprovisionamiento, ejecución, monitorización, control y cierre de un proyecto
- ◆ Adquirir los conocimientos esenciales relacionados con la responsabilidad profesional derivada de la gestión de proyectos
- ◆ Conocer los conceptos fundamentales de la dirección de proyectos y el ciclo de vida de la gestión de proyecto
- ◆ Entender las distintas etapas de la gestión de proyectos como son el inicio, la planificación, la gestión de los *Stakeholders* y el alcance

03

Competencias

Entre las características más significativas de todas las titulaciones ofertadas por TECH está el permitir al egresado perfeccionar sus competencias a través del curso de las mismas. Esto es posible gracias a la adquisición de un conocimiento especializado sobre la materia, en este caso sobre la Ingeniería del Software. Sin embargo, la clave para conseguirlo de manera garantizada reside en la resolución de casos de uso basados en situaciones del contexto actual del panorama informático, gracias a los cuales el alumno puede poner en práctica sus habilidades, aplicar las estrategias señaladas en el temario y ampliar sus aptitudes en el diseño, análisis, gestión y lanzamiento de aplicaciones y programas.



“

El mejor programa del mercado académico actual para implementar a tu praxis Informática las técnicas de prototipado más innovadoras y eficaces del entorno del software”

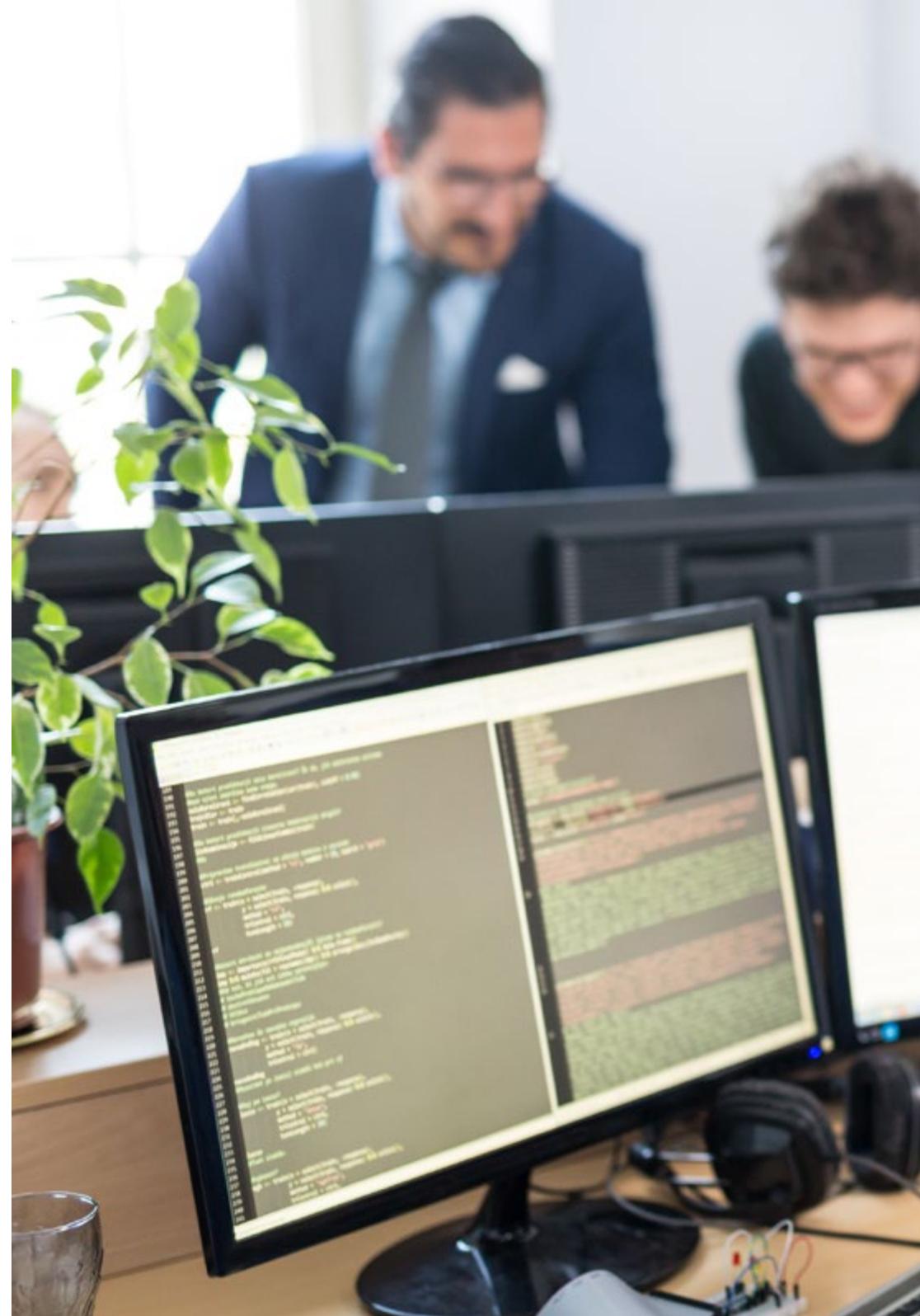


Competencias generales

- ◆ Responder a las necesidades actuales del área de Ingeniería del Software Avanzada
- ◆ Tener el dominio de los diferentes sistemas de trabajo en Ingeniería del Software Avanzada
- ◆ Describir y aprovechar el software libre y el conocimiento abierto existente en la red

“

Gracias al conocimiento de las estrategias y técnicas de pruebas de software podrás trabajar en la mejora de proyectos de otras personas para hacerlos más intuitivos y especializados”





Competencias específicas

- ◆ Conocer en profundidad todas las facetas de la interacción persona-ordenador y cómo involucran los desarrollos informáticos
- ◆ Ser solvente en el uso de bases de datos
- ◆ Desarrollar diferentes tipos de aplicaciones en red
- ◆ Trabajar como ingeniero de software
- ◆ Controlar el uso de las bases de datos avanzadas
- ◆ Realizar una programación avanzada
- ◆ Saber cómo se reutiliza el software
- ◆ Crear interfaces y aplicaciones en red

04

Estructura y contenido

Para este Máster Título Propio en Ingeniería del Software Avanzada TECH ha solicitado la ayuda de un equipo de expertos en Informática y Computación Digital. Gracias a ello, ha sido posible conformar un plan de estudios multidisciplinar y dinámico, pero, sobre todo, totalmente innovador, mediante la inclusión de la información más completa y actualizada sobre la gestión de proyectos relacionados con el diseño de programas y aplicaciones. Además, ha incluido cientos de horas de material adicional de la mejor calidad, para que el egresado pueda profundizar de manera personalizada en los distintos apartados del temario, así como satisfacer sus necesidades académicas de forma garantizada a través de una capacitación 100% Online.



“

Un programa con el que podrás trabajar en la toma de decisiones correctas con respecto a la inversión en TIC gracias a la planificación exhaustiva de los sistemas de información”

Módulo 1. Ingeniería del Software

- 1.1. Introducción a la Ingeniería del Software y al modelado
 - 1.1.1. La naturaleza del software
 - 1.1.2. La naturaleza única de las *Webapps*
 - 1.1.3. Ingeniería del Software
 - 1.1.4. El proceso del software
 - 1.1.5. La práctica de la Ingeniería del Software
 - 1.1.6. Mitos del software
 - 1.1.7. ¿Cómo comienza todo?
 - 1.1.8. Conceptos orientados a objetos
 - 1.1.9. Introducción a UML
- 1.2. El proceso del software
 - 1.2.1. Un modelo general de proceso
 - 1.2.2. Modelos de proceso prescriptivos
 - 1.2.3. Modelos de proceso especializado
 - 1.2.4. El proceso unificado
 - 1.2.5. Modelos del proceso personal y del equipo
 - 1.2.6. ¿Qué es la agilidad?
 - 1.2.7. ¿Qué es un proceso ágil?
 - 1.2.8. *Scrum*
 - 1.2.9. Conjunto de herramientas para el proceso ágil
- 1.3. Principios que guían la práctica de la Ingeniería del Software
 - 1.3.1. Principios que guían el proceso
 - 1.3.2. Principios que guían la práctica
 - 1.3.3. Principios de comunicación
 - 1.3.4. Principios de planificación
 - 1.3.5. Principios de modelado
 - 1.3.6. Principios de construcción
 - 1.3.7. Principios de despliegue
- 1.4. Comprensión de los requisitos
 - 1.4.1. Ingeniería de requisitos
 - 1.4.2. Establecer las bases
 - 1.4.3. Indagación de los requisitos
 - 1.4.4. Desarrollo de casos de uso
 - 1.4.5. Elaboración del modelo de los requisitos
 - 1.4.6. Negociación de los requisitos
 - 1.4.7. Validación de los requisitos
- 1.5. Modelado de los requisitos I: escenarios, información y clases de análisis
 - 1.5.1. Análisis de los requisitos
 - 1.5.2. Modelado basado en escenarios
 - 1.5.3. Modelos UML que proporcionan el caso de uso
 - 1.5.4. Conceptos de modelado de datos
 - 1.5.5. Modelado basado en clases
 - 1.5.6. Diagramas de clases
- 1.6. Modelado de los requisitos II: flujo, comportamiento y patrones
 - 1.6.1. Requisitos que modelan las estrategias
 - 1.6.2. Modelado orientado al flujo
 - 1.6.3. Diagramas de estado
 - 1.6.4. Creación de un modelo de comportamiento
 - 1.6.5. Diagramas de secuencia
 - 1.6.6. Diagramas de comunicación
 - 1.6.7. Patrones para el modelado de requisitos
- 1.7. Conceptos de diseño
 - 1.7.1. Diseño en el contexto de la Ingeniería del Software
 - 1.7.2. El proceso de diseño
 - 1.7.3. Conceptos de diseño
 - 1.7.4. Conceptos de diseño orientado a objetos
 - 1.7.5. El modelo del diseño

- 1.8. Diseño de la arquitectura
 - 1.8.1. Arquitectura del software
 - 1.8.2. Géneros arquitectónicos
 - 1.8.3. Estilos arquitectónicos
 - 1.8.4. Diseño arquitectónico
 - 1.8.5. Evolución de los diseños alternativos para la arquitectura
 - 1.8.6. Mapeo de la arquitectura con el uso del flujo de datos
- 1.9. Diseño en el nivel de componentes y basado en patrones
 - 1.9.1. ¿Qué es un componente?
 - 1.9.2. Diseño de componentes basados en clase
 - 1.9.3. Realización del diseño en el nivel de componentes
 - 1.9.4. Diseño de componentes tradicionales
 - 1.9.5. Desarrollo basado en componentes
 - 1.9.6. Patrones de diseño
 - 1.9.7. Diseño de software basado en patrones
 - 1.9.8. Patrones arquitectónicos
 - 1.9.9. Patrones de diseño en el nivel de componentes
 - 1.9.10. Patrones de diseño de la interfaz de usuario
- 1.10. Calidad del software y administración de proyectos
 - 1.10.1. Calidad
 - 1.10.2. Calidad del software
 - 1.10.3. El dilema de la calidad del software
 - 1.10.4. Lograr la calidad del software
 - 1.10.5. Aseguramiento de la calidad del software
 - 1.10.6. El espectro administrativo
 - 1.10.7. El personal
 - 1.10.8. El producto
 - 1.10.9. El proceso
 - 1.10.10. El proyecto
 - 1.10.11. Principios y prácticas

Módulo 2. Ingeniería del Software Avanzada

- 2.1. Introducción a las metodologías ágiles
 - 2.1.1. Modelos de proceso y metodologías
 - 2.1.2. Agilidad y procesos ágiles
 - 2.1.3. Manifiesto ágil
 - 2.1.4. Algunas metodologías ágiles
 - 2.1.5. Ágil vs. Tradicional
- 2.2. *Scrum*
 - 2.2.1. Orígenes y filosofía de *Scrum*
 - 2.2.2. Valores de *Scrum*
 - 2.2.3. Flujo del proceso *Scrum*
 - 2.2.4. Los roles de *Scrum*
 - 2.2.5. Los artefactos de *Scrum*
 - 2.2.6. Los eventos de *Scrum*
 - 2.2.7. Las historias de usuario
 - 2.2.8. Extensiones de *Scrum*
 - 2.2.9. Estimaciones ágiles
 - 2.2.10. Escalado de *Scrum*
- 2.3. Programación extrema
 - 2.3.1. Justificación y visión general de XP
 - 2.3.2. El ciclo de vida en XP
 - 2.3.3. Los cinco valores básicos
 - 2.3.4. Las doce prácticas básicas en XP
 - 2.3.5. Roles de los participantes
 - 2.3.6. XP Industrial
 - 2.3.7. Valoración crítica de XP
- 2.4. Desarrollo de software basado en reutilización
 - 2.4.1. La reutilización del software
 - 2.4.2. Niveles de reutilización de código
 - 2.4.3. Técnicas concretas de reutilización
 - 2.4.4. Desarrollo basado en componentes
 - 2.4.5. Beneficios y problemas de la reutilización
 - 2.4.6. Planificación de la reutilización

- 2.5. Patrones de arquitectura de sistemas y de diseño de software
 - 2.5.1. El diseño arquitectónico
 - 2.5.2. Patrones arquitectónicos generales
 - 2.5.3. Arquitecturas tolerantes a fallos
 - 2.5.4. Arquitecturas de sistemas distribuidos
 - 2.5.5. Los patrones de diseño
 - 2.5.6. Patrones de Gamma
 - 2.5.7. Patrones de diseño de interacción
- 2.6. Arquitectura de aplicaciones en la nube
 - 2.6.1. Fundamentos de *Cloud Computing*
 - 2.6.2. Calidad de las aplicaciones en la nube
 - 2.6.3. Estilos de arquitectura
 - 2.6.4. Patrones de diseño
- 2.7. Pruebas del software: TDD, ATDD y BDD
 - 2.7.1. Verificación y validación del software
 - 2.7.2. Las pruebas de software
 - 2.7.3. *Test Driven Development* (TDD)
 - 2.7.4. *Acceptance Test Driven Development* (ATDD)
 - 2.7.5. *Behavior Driven Development* (BDD)
 - 2.7.6. BDD y *Cucumber*
- 2.8. La mejora del proceso de software
 - 2.8.1. La mejora del proceso de software
 - 2.8.2. El proceso de mejora de procesos
 - 2.8.3. Modelos de madurez
 - 2.8.4. El modelo CMMI
 - 2.8.5. CMMI V2.0
 - 2.8.6. CMMI y Ágil





- 2.9. La calidad del producto software: Square
 - 2.9.1. La calidad del software
 - 2.9.2. Modelos de calidad del producto software
 - 2.9.3. Familia ISO/IEC 25000
 - 2.9.4. ISO/IEC 25010: modelo y características de calidad
 - 2.9.5. ISO/IEC 25012: la calidad de los datos
 - 2.9.6. ISO/IEC 25020: medición de la calidad del software
 - 2.9.7. ISO/IEC 25022, 25023 y 25024: métricas de calidad del software y de los datos
 - 2.9.8. ISO/IEC 25040: evaluación del software
 - 2.9.9. El proceso de certificación
- 2.10. Introducción a DevOps
 - 2.10.1. Concepto de DevOps
 - 2.10.2. Prácticas principales

Módulo 3. Ingeniería de requisitos

- 3.1. Introducción a la Ingeniería de Requisitos
 - 3.1.1. La importancia de los requisitos
 - 3.1.2. Concepto de requisito
 - 3.1.3. Dimensiones de los requisitos
 - 3.1.4. Niveles y tipos de requisitos
 - 3.1.5. Características de los requisitos
 - 3.1.6. La Ingeniería de requisitos
 - 3.1.7. El proceso de Ingeniería de Requisitos
 - 3.1.8. *Frameworks* para Ingeniería de Requisitos
 - 3.1.9. Buenas prácticas en Ingeniería de Requisitos
 - 3.1.10. El analista de negocio
- 3.2. Las fuentes de los requisitos
 - 3.2.1. La red de requisitos
 - 3.2.2. Los *Stakeholders*
 - 3.2.3. Los requisitos de negocio
 - 3.2.4. Documento de visión y alcance

- 3.3. Técnicas de elicitación de requisitos
 - 3.3.1. La elicitación de requisitos
 - 3.3.2. Problemas de la elicitación de requisitos
 - 3.3.3. Contextos de descubrimiento
 - 3.3.4. Entrevistas
 - 3.3.5. Observación y «aprendizaje»
 - 3.3.6. Etnografía
 - 3.3.7. *Workshops*
 - 3.3.8. *Focus Groups*
 - 3.3.9. Cuestionarios
 - 3.3.10. *Brainstorming* y técnicas creativas
 - 3.3.11. Medios grupales
 - 3.3.12. Análisis de interfaces del sistema
 - 3.3.13. Análisis de documentos y «arqueología»
 - 3.3.14. Casos de uso y escenarios
 - 3.3.15. Los prototipos
 - 3.3.16. La Ingeniería inversa
 - 3.3.17. Reutilización de requisitos
 - 3.3.18. Buenas prácticas de la elicitación
- 3.4. Requisitos de los usuarios
 - 3.4.1. Personas
 - 3.4.2. Casos de uso e historias de usuario
 - 3.4.3. Escenarios
 - 3.4.5. Tipos de escenarios
 - 3.4.6. ¿Cómo descubrir escenarios?
- 3.5. Técnicas de prototipado
 - 3.5.1. El prototipado
 - 3.5.2. Prototipos según su alcance
 - 3.5.3. Prototipos según su temporalidad
 - 3.5.4. La fidelidad de un prototipo
 - 3.5.5. Prototipos de interfaz de usuario
 - 3.5.6. Evaluación de prototipos
- 3.6. Análisis de requisitos
 - 3.6.1. El análisis de requisitos
 - 3.6.2. Buenas prácticas del análisis de requisitos
 - 3.6.3. El diccionario de datos
 - 3.6.4. Priorización de requisitos
- 3.7. Documentación de los requisitos
 - 3.7.1. El documento especificación de requisitos
 - 3.7.2. Estructura y contenidos de un SRS
 - 3.7.3. Documentación en lenguaje natural
 - 3.7.4. EARS: *Easy Approach to Requirements Syntax*
 - 3.7.5. Los requisitos no funcionales
 - 3.7.6. Atributos y plantillas en forma de tabla
 - 3.7.7. Buenas prácticas de especificación
- 3.8. Validación y negociación de requisitos
 - 3.8.1. Validación de requisitos
 - 3.8.2. Técnicas de validación de requisitos
 - 3.8.3. Negociación de requisitos
- 3.9. Modelado y gestión de requisitos
 - 3.9.1. El modelado de requisitos
 - 3.9.2. La perspectiva del usuario
 - 3.9.3. La perspectiva de los datos
 - 3.9.4. La perspectiva funcional u orientada al flujo
 - 3.9.5. La perspectiva del comportamiento
 - 3.9.6. La volatilidad de los requisitos
 - 3.9.7. Proceso de gestión de requisitos
 - 3.9.8. Herramientas para gestión de requisitos
 - 3.9.9. Buenas prácticas en la gestión de requisitos
- 3.10. Sistemas críticos y especificación formal
 - 3.10.1. Los sistemas críticos
 - 3.10.2. Especificación dirigida por riesgos
 - 3.10.3. Especificación formal

Módulo 4. Procesos de Ingeniería del Software

- 4.1. Marco de Ingeniería del Software
 - 4.1.1. Características del software
 - 4.1.2. Los procesos principales en Ingeniería del Software
 - 4.1.3. Modelos de proceso de desarrollo software
 - 4.1.4. Marco de referencia estándar para el proceso de desarrollo de software: la norma ISO/IEC 12207
- 4.2. Proceso unificado de desarrollo software
 - 4.2.1. Proceso unificado
 - 4.2.2. Dimensiones del proceso unificado
 - 4.2.3. Proceso de desarrollo dirigido por casos de uso
 - 4.2.4. Flujos de trabajo fundamentales de procesos unificados
- 4.3. Planificación en el contexto de desarrollo de software ágil
 - 4.3.1. Características del desarrollo software ágil
 - 4.3.2. Diferentes horizontes temporales de planificación en el desarrollo ágil
 - 4.3.3. Marco de desarrollo ágil *Scrum* y horizontes temporales de planificación
 - 4.3.4. Historias de usuario como unidad de planificación y estimación
 - 4.3.5. Técnicas comunes para derivar una estimación
 - 4.3.6. Escalas para interpretar las estimaciones
 - 4.3.7. *Planning Poker*
 - 4.3.8. Tipos de planificaciones comunes: planificación de entregas y planificación de iteración
- 4.4. Estilos de diseño de software distribuido y arquitecturas software orientadas a servicios
 - 4.4.1. Modelos de comunicación en sistemas software distribuidos
 - 4.4.2. Capa intermedia o *Middleware*
 - 4.4.3. Patrones de arquitectura para sistemas distribuidos
 - 4.4.4. Proceso general de diseño de servicios software
 - 4.4.5. Aspectos de diseño de servicios software
 - 4.4.6. Composición de servicios
 - 4.4.7. Arquitectura de servicios web
 - 4.4.8. Componentes de Infraestructura y SOA
- 4.5. Introducción al desarrollo software dirigido por modelos
 - 4.5.1. El concepto de modelo
 - 4.5.2. Desarrollo software dirigido por modelos
 - 4.5.3. Marco de referencia de desarrollo dirigido por modelos MDA
 - 4.5.4. Elementos de un modelo de transformación
- 4.6. Diseño de interfaces gráficas de usuario
 - 4.6.1. Principios de diseño de interfaces de usuario
 - 4.6.2. Patrones de diseño arquitectónico para sistemas interactivos: Modelo Vista Controlador (MVC)
 - 4.6.3. Experiencia de usuario (UX *User Experience*)
 - 4.6.4. Diseño centrado en el usuario
 - 4.6.5. Proceso de análisis y diseño de la interfaz gráfica de usuario
 - 4.6.6. Usabilidad de interfaces de usuario
 - 4.6.7. Accesibilidad en interfaces de usuario
- 4.7. Diseño de aplicaciones web
 - 4.7.1. Características de las aplicaciones web
 - 4.7.2. Interfaz de usuario de una aplicación web
 - 4.7.3. Diseño de navegación
 - 4.7.4. Protocolo de interacción base para aplicaciones web
 - 4.7.5. Estilos de arquitectura para aplicaciones web
- 4.8. Estrategias y técnicas de pruebas software y factores de calidad del software
 - 4.8.1. Estrategias de prueba
 - 4.8.2. Diseños de casos de prueba
 - 4.8.3. Relación coste-calidad
 - 4.8.4. Modelos de calidad
 - 4.8.5. Familia de normas ISO/IEC 25000 (Square)
 - 4.8.6. Modelo de calidad de producto (ISO 2501n)
 - 4.8.7. Modelos de calidad de datos (ISO 2501n)
 - 4.8.8. Gestión de la calidad del software

- 4.9. Introducción a las métricas en Ingeniería de Software
 - 4.9.1. Conceptos básicos: medidas, métricas e indicadores
 - 4.9.2. Tipos de métricas en Ingeniería de Software
 - 4.9.3. El proceso de medición
 - 4.9.4. ISO 25024. Métricas externas y de calidad en uso
 - 4.9.5. Métrica orientada a objetos
- 4.10. Mantenimiento y Reingeniería de software
 - 4.10.1. Proceso de mantenimiento
 - 4.10.2. Marco estándar de proceso de mantenimiento. ISO/EIEC 14764
 - 4.10.3. Modelo de proceso de reingeniería de software
 - 4.10.4. Ingeniería inversa

Módulo 5. Calidad y auditoría de Sistemas de Información

- 5.1. Introducción a los Sistemas de Gestión de Seguridad de la Información
 - 5.1.1. Principios fundamentales de los SGSI
 - 5.1.2. Reglas de oro de los SGSI
 - 5.1.3. Papel de la auditoría informática en los SGSI
- 5.2. Planificación en la gestión de la seguridad
 - 5.2.1. Conceptos relativos a la gestión de la seguridad
 - 5.2.2. Clasificación de la información: objetivos, conceptos y roles
 - 5.2.3. Implementación de las políticas de seguridad: políticas de seguridad, estándares y procedimientos
 - 5.2.4. Gestión del riesgo: principios y análisis del riesgo de los activos de información
- 5.3. Principales mecanismos para la protección de activos información I
 - 5.3.1. Resumen de las principales herramientas criptográficas para la protección de la triada CID
 - 5.3.2. Consideración de los requisitos de privacidad, anonimato y gestión adecuada de la trazabilidad de usuarios
- 5.4. Principales mecanismos para la protección de activos información II
 - 5.4.1. Seguridad de las comunicaciones: protocolos, dispositivos y arquitecturas de seguridad
 - 5.4.2. Seguridad de los sistemas operativos





- 5.5. Controles internos de los SGSI
 - 5.5.1. Taxonomía de los controles SGSI: controles administrativos, lógicos y físicos
 - 5.5.2. Clasificación de los controles en función del modo de abordar la amenaza: controles para la prevención, la detección y la corrección de amenazas
 - 5.5.3. Implantación de sistemas de control interno en los SGSI
- 5.6. Tipos de auditoría
 - 5.6.1. Diferencia entre auditoría y control interno
 - 5.6.2. Auditoría interna frente a auditoría externa
 - 5.6.3. Clasificación de la auditoría en función del objetivo y el tipo de análisis
- 5.7. Guionista y guion: sujeto y objeto protegido por la propiedad intelectual
 - 5.7.1. Introducción a los test de penetración y al análisis forense
 - 5.7.2. Definición y relevancia de los conceptos de *Fingerprinting* y *Footprinting*
- 5.8. Análisis de vulnerabilidades y monitorización de tráfico de red
 - 5.8.1. Herramientas para el análisis de vulnerabilidades en sistemas
 - 5.8.2. Principales vulnerabilidades en el contexto de las aplicaciones web
 - 5.8.3. Análisis de protocolos de comunicaciones
- 5.9. El proceso de la auditoría informática
 - 5.9.1. Concepto de ciclo de vida en el desarrollo de sistemas
 - 5.9.2. Monitorización de actividad y de procesos: recolección y tratamiento de evidencias
 - 5.9.3. Metodología de la auditoría informática
 - 5.9.4. Proceso de una auditoría informática
 - 5.9.5. Identificación de los principales delitos y faltas en el contexto de las tecnologías de la información
 - 5.9.6. Investigación de delitos informáticos: introducción al análisis forense y su relación con la auditoría informática
- 5.10. Planes de continuidad de negocio y de recuperación frente a desastres
 - 5.10.1. Definición de plan de continuidad de negocio y del concepto de interrupción del negocio
 - 5.10.2. Recomendación NIST sobre los planes de continuidad de negocio
 - 5.10.3. Plan de recuperación ante desastres
 - 5.10.4. Proceso de plan de recuperación ante desastres

Módulo 6. Integración de sistemas

- 6.1. Introducción a los Sistemas de Información en la empresa
 - 6.1.1. El papel de los Sistemas de Información
 - 6.1.2. ¿Qué es un Sistema de Información?
 - 6.1.3. Dimensiones de los Sistemas de Información
 - 6.1.4. Procesos de negocio y Sistemas de Información
 - 6.1.5. El departamento de SI/TI
- 6.2. Oportunidades y necesidades de los Sistemas de Información en la empresa
 - 6.2.1. Organizaciones y Sistemas de Información
 - 6.2.2. Características de las organizaciones
 - 6.2.3. Impacto de los Sistemas de Información en la empresa
 - 6.2.4. Sistemas de información para lograr una ventaja competitiva
 - 6.2.5. Uso de los sistemas en la administración y gestión de la empresa
- 6.3. Conceptos básicos de sistemas y tecnologías de la información
 - 6.3.1. Datos, información y conocimiento
 - 6.3.2. Tecnología y Sistemas de Información
 - 6.3.3. Componentes de la tecnología
 - 6.3.4. Clasificación y tipos de Sistemas de Información
 - 6.3.5. Arquitecturas basadas en servicios y procesos de negocio
 - 6.3.6. Formas de integración de sistemas
- 6.4. Sistemas para la gestión integrada de recursos de la empresa
 - 6.4.1. Necesidades de la empresa
 - 6.4.2. Un sistema de información integrado para la empresa
 - 6.4.3. Adquisición vs. Desarrollo
 - 6.4.4. Implantación de un ERP
 - 6.4.5. Implicaciones para la dirección
 - 6.4.6. Principales proveedores de ERP
- 6.5. Sistemas de información para la gestión de la cadena de suministro y las relaciones con clientes
 - 6.5.1. Definición de cadena de suministro
 - 6.5.2. Gestión efectiva de la cadena de suministro
 - 6.5.3. El papel de los Sistemas de Información
 - 6.5.4. Soluciones para la gestión de cadena de suministro
 - 6.5.5. La gestión de relaciones con los clientes
 - 6.5.6. El papel de los Sistemas de Información
 - 6.5.7. Implantación de un sistema CRM
 - 6.5.8. Factores críticos de éxito en la implantación de CRM
 - 6.5.9. CRM, e-CRM y otras tendencias
- 6.6. La toma de decisiones de inversión en TIC y planificación de Sistemas de Información
 - 6.6.1. Criterios para la decisión de inversión en TIC
 - 6.6.2. Vinculación del proyecto con la gerencia y plan de negocios
 - 6.6.3. Implicaciones de la dirección
 - 6.6.4. Rediseño de los procesos de negocio
 - 6.6.5. Decisión de metodologías de implantación desde la dirección
 - 6.6.6. Necesidad de planificación de los Sistemas de Información
 - 6.6.7. Objetivos, participantes y momentos
 - 6.6.8. Estructura y desarrollo del plan de sistemas
 - 6.6.9. Seguimiento y actualización
- 6.7. Consideraciones de seguridad en el uso de las TIC
 - 6.7.1. Análisis de riesgos
 - 6.7.2. La seguridad en los Sistemas de Información
 - 6.7.3. Consejos prácticos
- 6.8. Viabilidad de aplicación de proyectos de TIC y aspectos financieros en proyectos de Sistemas de Información
 - 6.8.1. Descripción y objetivos
 - 6.8.2. Participantes en el EVS
 - 6.8.3. Técnicas y prácticas
 - 6.8.4. Estructura de costes
 - 6.8.5. La proyección financiera
 - 6.8.6. Presupuestos

- 6.9. *Business Intelligence*
 - 6.9.1. ¿Qué es la inteligencia de negocio?
 - 6.9.2. Estrategia e implantación de BI
 - 6.9.3. Presente y futuro en BI
- 6.10. ISO/IEC 12207
 - 6.10.1. ¿Qué es «ISO/IEC 12207»?
 - 6.10.2. Análisis de los Sistemas de Información
 - 6.10.3. Diseño del Sistema de Información
 - 6.10.4. Implantación y aceptación del Sistema de Información

Módulo 7. Reutilización de software

- 7.1. Panorama general de la reutilización de software
 - 7.1.1. ¿En qué consiste la reutilización del software?
 - 7.1.2. Ventajas e inconvenientes de la reutilización de software
 - 7.1.3. Principales técnicas de reutilización de software
- 7.2. Introducción a los patrones de diseño
 - 7.2.1. ¿Qué es un patrón de diseño?
 - 7.2.2. Catálogo de los principales patrones de diseño
 - 7.2.3. ¿Cómo usar patrones para resolver problemas de diseño?
 - 7.2.4. ¿Cómo seleccionar el mejor patrón de diseño?
- 7.3. Patrones de creación I
 - 7.3.1. Patrones de creación
 - 7.3.2. Patrón *Abstract Factory*
 - 7.3.3. Ejemplo de implementación del Patrón *Abstract Factory*
 - 7.3.4. Patrón *Builder*
 - 7.3.5. Ejemplo de implementación del *Builder*
 - 7.3.6. Patrón *Abstract Factory vs.Builder*
- 7.4. Patrones de creación II
 - 7.4.1. Patrón *Factory Method*
 - 7.4.2. *Factory Method vs. Abstract Factory*
 - 7.4.3. Patrón *Singleton*
- 7.5. Patrones estructurales I
 - 7.5.1. Patrones estructurales
 - 7.5.2. Patrón *Adapter*
 - 7.5.3. Patrón *Bridge*
- 7.6. Patrones estructurales II
 - 7.6.1. Patrón *Composite*
 - 7.6.2. Patrón Decorador
- 7.7. Patrones estructurales III
 - 7.7.1. Patrón *Facade*
 - 7.7.2. Patrón *Proxy*
- 7.8. Patrones de comportamiento I
 - 7.8.1. Concepto de los patrones de comportamiento
 - 7.8.2. Patrón de comportamiento: cadena de responsabilidad
 - 7.8.3. Patrón de comportamiento orden
- 7.9. Patrones de comportamiento II
 - 7.9.1. Patrón Intérprete o *Interpreter*
 - 7.9.2. Patrón iterador
 - 7.9.3. Patrón observador
 - 7.9.4. Patrón estrategia
- 7.10. *Frameworks*
 - 7.10.1. Concepto de *Framework*
 - 7.10.2. Desarrollo mediante *Frameworks*
 - 7.10.3. Patrón *Model View Controller*
 - 7.10.4. *Framework* para diseño de interfaces gráficas de usuario
 - 7.10.5. *Frameworks* para el desarrollo de aplicaciones web
 - 7.10.6. *Frameworks* para la gestión de la persistencia de objetos en bases de datos

Módulo 8. Servicios de tecnología de la información

- 8.1. La transformación digital I
 - 8.1.1. La innovación empresarial
 - 8.1.2. La gestión de la producción
 - 8.1.3. La gestión financiera
- 8.2. La transformación digital II
 - 8.2.1. El Marketing
 - 8.2.2. La gestión de RR.HH.
 - 8.2.3. Un sistema de información integrado
- 8.3. Caso de estudio
 - 8.3.1. Presentación de la empresa
 - 8.3.2. Metodologías para analizar la adquisición de TI
 - 8.3.3. Determinación de costos, beneficios y riesgos
 - 8.3.4. Evaluación económica de la inversión
- 8.4. El gobierno y la gestión de las TIC
 - 8.4.1. Definición de gobierno de las tecnologías y sistemas de la información
 - 8.4.2. Diferencia entre gobierno y gestión de las TIC
 - 8.4.3. Marcos para el gobierno y la gestión de las TIC
 - 8.4.4. Las normas y el gobierno y la gestión de las TIC
- 8.5. El gobierno corporativo de las TIC
 - 8.5.1. ¿Qué es el buen gobierno corporativo?
 - 8.5.2. Antecedentes de gobierno de las TIC
 - 8.5.3. La Norma ISO/IEC 38500:2008
 - 8.5.4. Implementación de un buen gobierno TIC
 - 8.5.5. Gobierno TIC y mejores prácticas
 - 8.5.6. Gobierno corporativo. Resumen y tendencias
- 8.6. Objetivos de Control para la Información y Tecnologías Relacionadas (COBIT)
 - 8.6.1. Marco de aplicación
 - 8.6.2. Dominio: planificación y organización
 - 8.6.3. Dominio: adquisición e implementación
 - 8.6.4. Dominio: entrega y soporte
 - 8.6.5. Dominio: supervisión y evaluación
 - 8.6.6. Aplicación de la guía COBIT



- 8.7. La Biblioteca de Infraestructura de Tecnologías de Información (ITIL)
 - 8.7.1. Introducción a ITIL
 - 8.7.2. Estrategia del servicio
 - 8.7.3. Diseño del servicio
 - 8.7.4. Transición del servicio
 - 8.7.5. Operación del servicio
 - 8.7.6. Mejora del servicio
- 8.8. El sistema de gestión de servicios
 - 8.8.1. Principios básicos de UNE-ISO/IEC 20000-1
 - 8.8.2. La estructura de la serie de normas ISO/IEC 20000
 - 8.8.3. Requisitos del Sistema de Gestión del Servicio (SGS)
 - 8.8.4. Diseño y transición de servicios nuevos o modificados
 - 8.8.5. Procesos de provisión del servicio
 - 8.8.6. Grupos de procesos
- 8.9. El sistema de gestión de activos de software
 - 8.9.1. Justificación de la necesidad
 - 8.9.2. Antecedentes
 - 8.9.3. Presentación de la norma 19770
 - 8.9.4. Implantación de la gestión
- 8.10. Gestión de la continuidad del negocio
 - 8.10.1. Plan de la continuidad del negocio
 - 8.10.2. Implementación de un BCM

Módulo 9. Seguridad en los Sistemas de Información

- 9.1. Una perspectiva global de la seguridad, la criptografía y los criptoanálisis clásicos
 - 9.1.1. La seguridad informática: perspectiva histórica
 - 9.1.2. Pero ¿qué se entiende exactamente por seguridad?
 - 9.1.3. Historia de la criptografía
 - 9.1.4. Cifradores de sustitución
 - 9.1.5. Caso de estudio: la máquina Enigma

- 9.2. Criptografía simétrica
 - 9.2.1. Introducción y terminología básica
 - 9.2.2. Cifrado simétrico
 - 9.2.3. Modos de operación
 - 9.2.4. DES
 - 9.2.5. El nuevo estándar AES
 - 9.2.6. Cifrado en flujo
 - 9.2.7. Criptoanálisis
- 9.3. Criptografía asimétrica
 - 9.3.1. Orígenes de la criptografía de clave pública
 - 9.3.2. Conceptos básicos y funcionamiento
 - 9.3.3. El algoritmo RSA
 - 9.3.4. Certificados digitales
 - 9.3.5. Almacenamiento y gestión de claves
- 9.4. Ataques en redes
 - 9.4.1. Amenazas y ataques de una red
 - 9.4.2. Enumeración
 - 9.4.3. Interceptación de tráfico: *Sniffers*
 - 9.4.4. Ataques de denegación de servicio
 - 9.4.5. Ataques de envenenamiento ARP
- 9.5. Arquitecturas de seguridad
 - 9.5.1. Arquitecturas de seguridad tradicionales
 - 9.5.2. Secure Socket Layer: SSL
 - 9.5.3. Protocolo SSH
 - 9.5.4. Redes Privadas Virtuales (VPN)
 - 9.5.5. Mecanismos de protección de unidades de almacenamiento externo
 - 9.5.6. Mecanismos de protección hardware
- 9.6. Técnicas de protección de sistemas y desarrollo de código seguro
 - 9.6.1. Seguridad en operaciones
 - 9.6.2. Recursos y controles
 - 9.6.3. Monitorización
 - 9.6.4. Sistemas de detección de intrusión
 - 9.6.5. IDS de *Host*
 - 9.6.6. IDS de red
 - 9.6.7. IDS basados en firmas
 - 9.6.8. Sistemas señuelos
 - 9.6.9. Principios de seguridad básicos en el desarrollo de código
 - 9.6.10. Gestión del fallo
 - 9.6.11. Enemigo público número 1: el desbordamiento de búfer
 - 9.6.12. Chapuzas criptográficas
- 9.7. *Botnets* y spam
 - 9.7.1. Origen del problema
 - 9.7.2. Proceso del spam
 - 9.7.3. Envío del spam
 - 9.7.4. Refinamiento de las listas de direcciones de correo
 - 9.7.5. Técnicas de protección
 - 9.7.6. Servicio antispam ofrecidos por terceros
 - 9.7.7. Casos de estudio
 - 9.7.8. Spam exótico
- 9.8. Auditoría y ataques web
 - 9.8.1. Recopilación de información
 - 9.8.2. Técnicas de ataque
 - 9.8.3. Herramientas

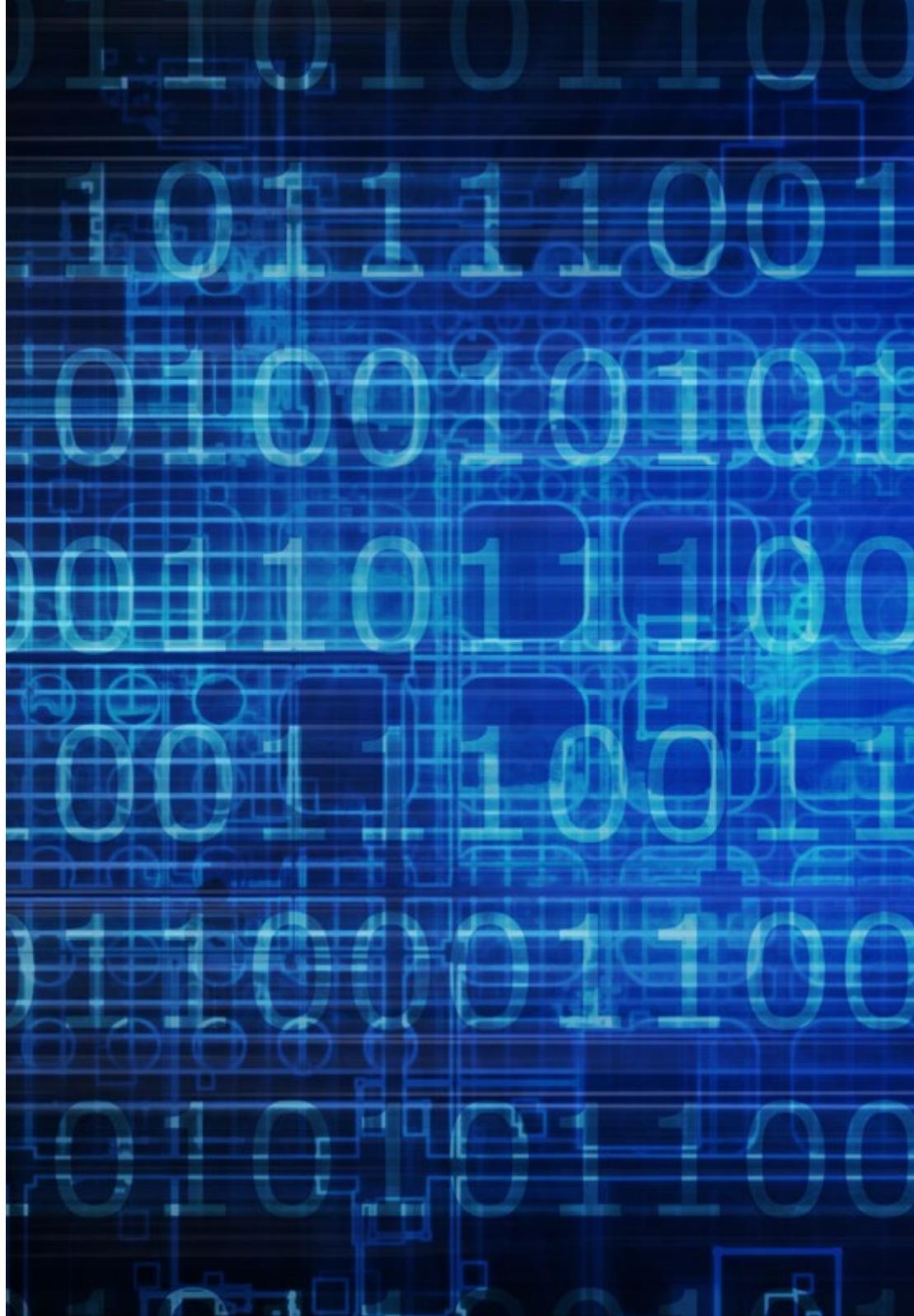
- 9.9. *Malware* y código malicioso
 - 9.9.1. ¿Qué es el *Malware*?
 - 9.9.2. Tipos de *Malware*
 - 9.9.3. Virus
 - 9.9.4. Criptovirus
 - 9.9.5. Gusanos
 - 9.9.6. *Adware*
 - 9.9.7. *Spyware*
 - 9.9.8. *Hoaxes*
 - 9.9.9. *Pishing*
 - 9.9.10. Troyanos
 - 9.9.11. La economía del *Malware*
 - 9.9.12. Posibles soluciones
- 9.10. Análisis forense
 - 9.10.1. Recolección de evidencias
 - 9.10.2. Análisis de las evidencias
 - 9.10.3. Técnicas antiforenses
 - 9.10.4. Caso de estudio práctico

Módulo 10. Gestión de proyectos

- 10.1. Conceptos fundamentales de la dirección de proyectos y el ciclo de vida de la gestión de proyectos
 - 10.1.1. ¿Qué es un proyecto?
 - 10.1.2. Metodología común
 - 10.1.3. ¿Qué es la dirección/gestión de proyectos?
 - 10.1.4. ¿Qué es un plan de proyecto?
 - 10.1.5. Beneficios
 - 10.1.6. Ciclo de vida del proyecto
 - 10.1.7. Grupos de procesos o ciclo de vida de la gestión de los proyectos
 - 10.1.8. La relación entre los grupos de procesos y las áreas de conocimiento
 - 10.1.9. Relaciones entre el ciclo de vida del producto y del proyecto

- 10.2. El inicio y la planificación
 - 10.2.1. De la idea al proyecto
 - 10.2.2. Desarrollo del acta de proyecto
 - 10.2.3. Reunión de arranque del proyecto
 - 10.2.4. Tareas, conocimientos y habilidades en el proceso de inicio
 - 10.2.5. El plan de proyecto
 - 10.2.6. Desarrollo del plan básico. Pasos
 - 10.2.7. Tareas, conocimientos y habilidades en el proceso de planificación
- 10.3. La gestión de los *Stakeholders* y del alcance
 - 10.3.1. Identificar a los interesados
 - 10.3.2. Desarrollar el plan para la gestión de los interesados
 - 10.3.3. Gestionar el compromiso de los interesados
 - 10.3.4. Controlar el compromiso de los interesados
 - 10.3.5. El objetivo del proyecto
 - 10.3.6. La gestión del alcance y su plan
 - 10.3.7. Recopilar los requisitos
 - 10.3.8. Definir el enunciado del alcance
 - 10.3.9. Crear la WBS (EDT)
 - 10.3.10. Verificar y controlar el alcance
- 10.4. El desarrollo del cronograma
 - 10.4.1. La gestión del tiempo y su plan
 - 10.4.2. Definir las actividades
 - 10.4.3. Establecimiento de la secuencia de las actividades
 - 10.4.4. Estimación de recursos de las actividades
 - 10.4.5. Estimación de la duración de las actividades
 - 10.4.6. Desarrollo del cronograma y cálculo del camino crítico
 - 10.4.7. Control del cronograma

- 10.5. El desarrollo del presupuesto y la respuesta a los riesgos
 - 10.5.1. Estimar los costes
 - 10.5.2. Desarrollar el presupuesto y la curva S
 - 10.5.3. Control de costes y método del valor ganado
 - 10.5.4. Los conceptos de riesgo
 - 10.5.5. ¿Cómo hacer un análisis de riesgos?
 - 10.5.6. El desarrollo del plan de respuesta
- 10.6. La gestión de la calidad
 - 10.6.1. Planificación de la calidad
 - 10.6.2. Aseguramiento de la calidad
 - 10.6.3. Control de la calidad
 - 10.6.4. Conceptos estadísticos básicos
 - 10.6.5. Herramientas de la gestión de la calidad
- 10.7. La comunicación y los Recursos Humanos
 - 10.7.1. Planificar la gestión de las comunicaciones
 - 10.7.2. Análisis de requisitos de comunicaciones
 - 10.7.3. Tecnología de las comunicaciones
 - 10.7.4. Modelos de comunicación
 - 10.7.5. Métodos de comunicación
 - 10.7.6. Plan de gestión de las comunicaciones
 - 10.7.7. Gestionar las comunicaciones
 - 10.7.8. La gestión de los recursos humanos
 - 10.7.9. Principales actores y sus roles en los proyectos
 - 10.7.10. Tipos de organizaciones
 - 10.7.11. Organización del proyecto
 - 10.7.12. El equipo de trabajo



- 10.8. El aprovisionamiento
 - 10.8.1. El proceso de adquisiciones
 - 10.8.2. Planificación
 - 10.8.3. Búsqueda de proveedores y solicitud de ofertas
 - 10.8.4. Adjudicación del contrato
 - 10.8.5. Administración del contrato
 - 10.8.6. Los contratos
 - 10.8.7. Tipos de contratos
 - 10.8.8. Negociación del contrato
- 10.9. Ejecución, monitorización y control y cierre
 - 10.9.1. Los grupos de procesos
 - 10.9.2. La ejecución del proyecto
 - 10.9.3. La monitorización y control del proyecto
 - 10.9.4. El cierre del proyecto
- 10.10. Responsabilidad profesional
 - 10.10.1. Responsabilidad profesional
 - 10.10.2. Características de la responsabilidad social y profesional
 - 10.10.3. Código deontológico del líder de proyectos
 - 10.10.4. Responsabilidad vs. PMP®
 - 10.10.5. Ejemplos de responsabilidad
 - 10.10.6. Beneficios de la profesionalización

“*No lo pienses más y apuesta por una titulación con la que triunfarás en el sector de la Ingeniería del Software mediante el diseño de programas y aplicaciones a la vanguardia del mercado tecnológico actual*”

05 Metodología

Este programa de capacitación ofrece una forma diferente de aprender. Nuestra metodología se desarrolla a través de un modo de aprendizaje de forma cíclica: **el Relearning**.

Este sistema de enseñanza es utilizado, por ejemplo, en las facultades de medicina más prestigiosas del mundo y se ha considerado uno de los más eficaces por publicaciones de gran relevancia como el ***New England Journal of Medicine***.





Descubre el Relearning, un sistema que abandona el aprendizaje lineal convencional para llevarte a través de sistemas cíclicos de enseñanza: una forma de aprender que ha demostrado su enorme eficacia, especialmente en las materias que requieren memorización”

Estudio de Caso para contextualizar todo el contenido

Nuestro programa ofrece un método revolucionario de desarrollo de habilidades y conocimientos. Nuestro objetivo es afianzar competencias en un contexto cambiante, competitivo y de alta exigencia.

“

Con TECH podrás experimentar una forma de aprender que está moviendo los cimientos de las universidades tradicionales de todo el mundo”



Accederás a un sistema de aprendizaje basado en la reiteración, con una enseñanza natural y progresiva a lo largo de todo el temario.



El alumno aprenderá, mediante actividades colaborativas y casos reales, la resolución de situaciones complejas en entornos empresariales reales.

Un método de aprendizaje innovador y diferente

El presente programa de TECH es una enseñanza intensiva, creada desde 0, que propone los retos y decisiones más exigentes en este campo, ya sea en el ámbito nacional o internacional. Gracias a esta metodología se impulsa el crecimiento personal y profesional, dando un paso decisivo para conseguir el éxito. El método del caso, técnica que sienta las bases de este contenido, garantiza que se sigue la realidad económica, social y profesional más vigente.

“*Nuestro programa te prepara para afrontar nuevos retos en entornos inciertos y lograr el éxito en tu carrera*”

El método del caso ha sido el sistema de aprendizaje más utilizado por las mejores escuelas de Informática del mundo desde que éstas existen. Desarrollado en 1912 para que los estudiantes de Derecho no solo aprendiesen las leyes a base de contenidos teóricos, el método del caso consistió en presentarles situaciones complejas reales para que tomaran decisiones y emitiesen juicios de valor fundamentados sobre cómo resolverlas. En 1924 se estableció como método estándar de enseñanza en Harvard.

Ante una determinada situación, ¿qué debería hacer un profesional? Esta es la pregunta a la que te enfrentamos en el método del caso, un método de aprendizaje orientado a la acción. A lo largo del curso, los estudiantes se enfrentarán a múltiples casos reales. Deberán integrar todos sus conocimientos, investigar, argumentar y defender sus ideas y decisiones.

Relearning Methodology

TECH aúna de forma eficaz la metodología del Estudio de Caso con un sistema de aprendizaje 100% online basado en la reiteración, que combina elementos didácticos diferentes en cada lección.

Potenciamos el Estudio de Caso con el mejor método de enseñanza 100% online: el Relearning.

En 2019 obtuvimos los mejores resultados de aprendizaje de todas las universidades online en español en el mundo.

En TECH aprenderás con una metodología vanguardista concebida para capacitar a los directivos del futuro. Este método, a la vanguardia pedagógica mundial, se denomina Relearning.

Nuestra universidad es la única en habla hispana licenciada para emplear este exitoso método. En 2019, conseguimos mejorar los niveles de satisfacción global de nuestros alumnos (calidad docente, calidad de los materiales, estructura del curso, objetivos...) con respecto a los indicadores de la mejor universidad online en español.



En nuestro programa, el aprendizaje no es un proceso lineal, sino que sucede en espiral (aprender, desaprender, olvidar y reaprender). Por eso, se combinan cada uno de estos elementos de forma concéntrica. Con esta metodología se han capacitado más de 650.000 graduados universitarios con un éxito sin precedentes en ámbitos tan distintos como la bioquímica, la genética, la cirugía, el derecho internacional, las habilidades directivas, las ciencias del deporte, la filosofía, el derecho, la ingeniería, el periodismo, la historia o los mercados e instrumentos financieros. Todo ello en un entorno de alta exigencia, con un alumnado universitario de un perfil socioeconómico alto y una media de edad de 43,5 años.

El Relearning te permitirá aprender con menos esfuerzo y más rendimiento, implicándote más en tu capacitación, desarrollando el espíritu crítico, la defensa de argumentos y el contraste de opiniones: una ecuación directa al éxito.

A partir de la última evidencia científica en el ámbito de la neurociencia, no solo sabemos organizar la información, las ideas, las imágenes y los recuerdos, sino que sabemos que el lugar y el contexto donde hemos aprendido algo es fundamental para que seamos capaces de recordarlo y almacenarlo en el hipocampo, para retenerlo en nuestra memoria a largo plazo.

De esta manera, y en lo que se denomina Neurocognitive context-dependent e-learning, los diferentes elementos de nuestro programa están conectados con el contexto donde el participante desarrolla su práctica profesional.



Este programa ofrece los mejores materiales educativos, preparados a conciencia para los profesionales:



Material de estudio

Todos los contenidos didácticos son creados por los especialistas que van a impartir el curso, específicamente para él, de manera que el desarrollo didáctico sea realmente específico y concreto.

Estos contenidos son aplicados después al formato audiovisual, para crear el método de trabajo online de TECH. Todo ello, con las técnicas más novedosas que ofrecen piezas de gran calidad en todos y cada uno los materiales que se ponen a disposición del alumno.



Clases magistrales

Existe evidencia científica sobre la utilidad de la observación de terceros expertos.

El denominado Learning from an Expert afianza el conocimiento y el recuerdo, y genera seguridad en las futuras decisiones difíciles.



Prácticas de habilidades y competencias

Realizarán actividades de desarrollo de competencias y habilidades específicas en cada área temática. Prácticas y dinámicas para adquirir y desarrollar las destrezas y habilidades que un especialista precisa desarrollar en el marco de la globalización que vivimos.



Lecturas complementarias

Artículos recientes, documentos de consenso y guías internacionales, entre otros. En la biblioteca virtual de TECH el estudiante tendrá acceso a todo lo que necesita para completar su capacitación.





Case studies

Completarán una selección de los mejores casos de estudio elegidos expresamente para esta titulación. Casos presentados, analizados y tutorizados por los mejores especialistas del panorama internacional.



Resúmenes interactivos

El equipo de TECH presenta los contenidos de manera atractiva y dinámica en píldoras multimedia que incluyen audios, vídeos, imágenes, esquemas y mapas conceptuales con el fin de afianzar el conocimiento.

Este exclusivo sistema educativo para la presentación de contenidos multimedia fue premiado por Microsoft como "Caso de éxito en Europa".



Testing & Retesting

Se evalúan y reevalúan periódicamente los conocimientos del alumno a lo largo del programa, mediante actividades y ejercicios evaluativos y autoevaluativos para que, de esta manera, el estudiante compruebe cómo va consiguiendo sus metas.



06

Titulación

El Máster Título Propio en Ingeniería del Software Avanzada, garantiza además de la capacitación más rigurosa y actualizada, el acceso a un título de Máster Propio expedido por TECH Global University.



“

Supera con éxito este programa y recibe tu titulación universitaria sin desplazamientos ni farragosos trámites”

Este programa te permitirá obtener el título propio de **Máster Título Propio en Ingeniería del Software Avanzada** avalado por **TECH Global University**, la mayor Universidad digital del mundo.

TECH Global University, es una Universidad Oficial Europea reconocida públicamente por el Gobierno de Andorra (*boletín oficial*). Andorra forma parte del Espacio Europeo de Educación Superior (EEES) desde 2003. El EEES es una iniciativa promovida por la Unión Europea que tiene como objetivo organizar el marco formativo internacional y armonizar los sistemas de educación superior de los países miembros de este espacio. El proyecto promueve unos valores comunes, la implementación de herramientas conjuntas y fortaleciendo sus mecanismos de garantía de calidad para potenciar la colaboración y movilidad entre estudiantes, investigadores y académicos.

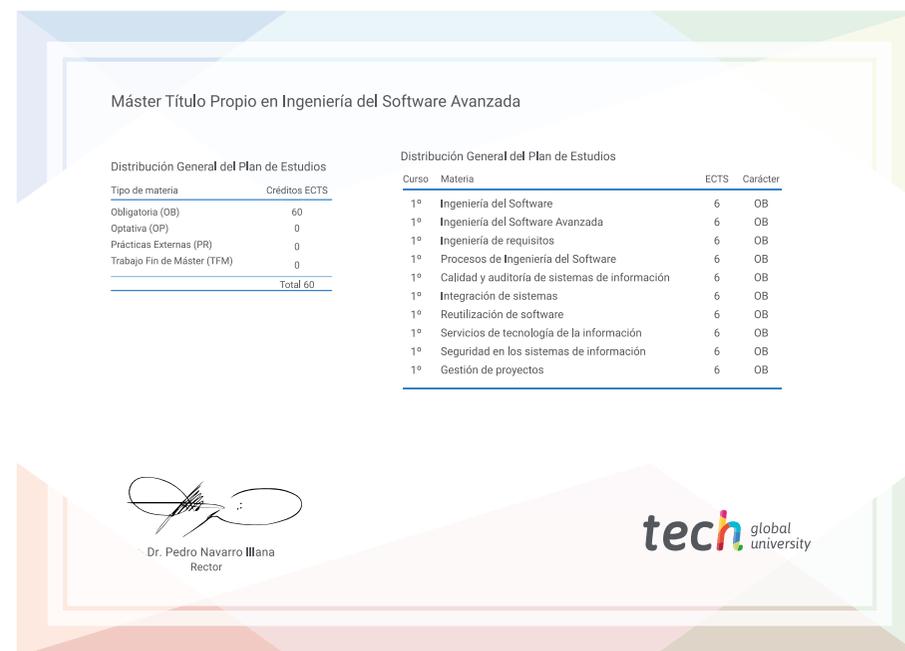
Este título propio de **TECH Global University**, es un programa europeo de formación continua y actualización profesional que garantiza la adquisición de las competencias en su área de conocimiento, confiriendo un alto valor curricular al estudiante que supere el programa.

Título: **Máster Título Propio en Ingeniería del Software Avanzada**

Modalidad: **online**

Duración: **12 meses**

Acreditación: **60 ECTS**



*Apostilla de La Haya. En caso de que el alumno solicite que su título en papel recabe la Apostilla de La Haya, TECH Global University realizará las gestiones oportunas para su obtención, con un coste adicional.



Máster Título Propio Ingeniería del Software Avanzada

- » Modalidad: online
- » Duración: 12 meses
- » Titulación: TECH Global University
- » Acreditación: 60 ECTS
- » Horario: a tu ritmo
- » Exámenes: online

Máster Título Propio

Ingeniería del Software Avanzada