

Grand Master

Ingeniería de Software





Grand Master Ingeniería de Software

- » Modalidad: No escolarizada (100% en línea)
- » Duración: 2 años
- » Titulación: TECH Universidad
- » Horario: a tu ritmo
- » Exámenes: online

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

Índice

01

Presentación del programa

pág. 4

02

¿Por qué estudiar en TECH?

pág. 8

03

Plan de estudios

pág. 12

04

Objetivos docentes

pág. 36

05

Salidas profesionales

pág. 42

06

Metodología de estudio

pág. 46

07

Cuadro docente

pág. 56

08

Titulación

pág. 60

01

Presentación del programa

La demanda de Software ha crecido exponencialmente en los últimos años. El surgimiento de nuevas plataformas digitales, hardware avanzado y la creciente virtualización de procesos cotidianos han planteado retos constantes para los ingenieros de Software. Con un público cada vez más habituado a las tecnologías emergentes y con expectativas más altas, los desarrolladores deben adaptarse y crear soluciones que respondan a las demandas del mercado. Esto exige un dominio técnico en múltiples áreas del conocimiento informático. Por ello, TECH ha diseñado este programa, que aborda competencias esenciales para enfrentar los desafíos actuales, incluyendo la planificación y diseño de Software, la implementación de medidas de seguridad, métodos de reutilización de Software y la administración de servidores web.



“

Desempeña un papel clave en el futuro tecnológico. Especialízate en Ingeniería de Software y crea sistemas transformadores”

La Ingeniería de Software se posiciona como una de las disciplinas más relevantes en la era tecnológica actual, ya que prácticamente todas las personas interactúan a diario con dispositivos digitales, impulsando un impacto significativo en las economías globales. En este contexto, el papel de los ingenieros de Software adquiere una gran relevancia, ya que la combinación de conocimientos técnicos y metodologías innovadoras permite crear soluciones efectivas y adaptadas a las necesidades del mercado. La alta especialización en este campo resulta fundamental para garantizar el éxito de proyectos tecnológicos, motivo por el cual TECH ha diseñado este completo programa universitario, pensado para ofrecer a los profesionales las herramientas más avanzadas y actualizadas.

Con este enfoque, el programa aborda los conceptos clave de la Ingeniería de Software, profundizando en aspectos fundamentales como el desarrollo de aplicaciones, las plataformas tecnológicas y las metodologías más avanzadas para la gestión de proyectos. La parte técnica adquiere un papel relevante, guiando al estudiante desde la concepción de ideas hasta su implementación en sistemas funcionales. Una oportunidad única para comprender la arquitectura del Software y la creación de soluciones innovadoras, al tiempo que se exploran las últimas tendencias en programación, desarrollo móvil y aplicaciones en la nube. Además, para fortalecer las capacidades de liderazgo, el contenido incluye los conceptos más actuales en gestión de proyectos y metodologías ágiles, un valor añadido para los profesionales con experiencia que buscan actualizarse o para quienes desean alcanzar posiciones estratégicas por primera vez.

Una de las principales ventajas de este programa es su modalidad 100% online, que elimina la necesidad de traslados o la adaptación a horarios rígidos. Esto permite al estudiante gestionar su aprendizaje de manera autónoma, organizando su tiempo y ritmo de estudio según sus propias necesidades, lo que resulta ideal para compaginarlo con otras responsabilidades diarias.

Este **Grand Master en Ingeniería de Software** contiene el programa educativo más completo y actualizado del mercado. Sus características más destacadas son:

- ♦ El desarrollo de casos prácticos presentados por expertos en Ingeniería de Software
- ♦ Los contenidos gráficos, esquemáticos y eminentemente prácticos con los que están concebidos recogen una información científica y 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 en la Dirección de Ingeniería de Software
- ♦ 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



Impulsa el desarrollo tecnológico con una gestión eficiente en Ingeniería de Software clave para el éxito organizacional"

“

Estudia a tu propio ritmo con un programa 100% online, disponible en cualquier momento y desde cualquier lugar del mundo”

Incluye en su cuadro docente a profesionales pertenecientes al ámbito de la Ingeniería de Software, que vierten en este programa 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 un estudio inmersivo programado para entrenarse ante situaciones reales.

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

Accede a la metodología más innovadora y actual para el aprendizaje en Ingeniería de Software, diseñada por TECH.

Consolida los conocimientos técnicos mediante una amplia variedad de recursos prácticos incluidos en este programa.



02

¿Por qué estudiar en TECH?

TECH es la mayor Universidad digital del mundo. Con un impresionante catálogo de más de 14.000 programas universitarios, disponibles en 11 idiomas, se posiciona como líder en empleabilidad, con una tasa de inserción laboral del 99%. Además, cuenta con un enorme claustro de más de 6.000 profesores de máximo prestigio internacional.



“

Estudia en la mayor universidad digital del mundo y asegura tu éxito profesional. El futuro empieza en TECH”

La mejor universidad online del mundo según FORBES

La prestigiosa revista Forbes, especializada en negocios y finanzas, ha destacado a TECH como «la mejor universidad online del mundo». Así lo han hecho constar recientemente en un artículo de su edición digital en el que se hacen eco del caso de éxito de esta institución, «gracias a la oferta académica que ofrece, la selección de su personal docente, y un método de aprendizaje innovador orientado a formar a los profesionales del futuro».

El mejor claustro docente top internacional

El claustro docente de TECH está integrado por más de 6.000 profesores de máximo prestigio internacional. Catedráticos, investigadores y altos ejecutivos de multinacionales, entre los cuales se destacan Isaiah Covington, entrenador de rendimiento de los Boston Celtics; Magda Romanska, investigadora principal de MetaLAB de Harvard; Ignacio Wistumba, presidente del departamento de patología molecular traslacional del MD Anderson Cancer Center; o D.W Pine, director creativo de la revista TIME, entre otros.

La mayor universidad digital del mundo

TECH es la mayor universidad digital del mundo. Somos la mayor institución educativa, con el mejor y más amplio catálogo educativo digital, cien por cien online y abarcando la gran mayoría de áreas de conocimiento. Ofrecemos el mayor número de titulaciones propias, titulaciones oficiales de posgrado y de grado universitario del mundo. En total, más de 14.000 títulos universitarios, en once idiomas distintos, que nos convierten en la mayor institución educativa del mundo.



Forbes
Mejor universidad
online del mundo

Plan
de estudios
más completo

Profesorado
TOP
Internacional


La metodología
más eficaz

nº1
Mundial
Mayor universidad
online del mundo

Los planes de estudio más completos del panorama universitario

TECH ofrece los planes de estudio más completos del panorama universitario, con temarios que abarcan conceptos fundamentales y, al mismo tiempo, los principales avances científicos en sus áreas científicas específicas. Asimismo, estos programas son actualizados continuamente para garantizar al alumnado la vanguardia académica y las competencias profesionales más demandadas. De esta forma, los títulos de la universidad proporcionan a sus egresados una significativa ventaja para impulsar sus carreras hacia el éxito.

Un método de aprendizaje único

TECH es la primera universidad que emplea el *Relearning* en todas sus titulaciones. Se trata de la mejor metodología de aprendizaje online, acreditada con certificaciones internacionales de calidad docente, dispuestas por agencias educativas de prestigio. Además, este disruptivo modelo académico se complementa con el "Método del Caso", configurando así una estrategia de docencia online única. También en ella se implementan recursos didácticos innovadores entre los que destacan vídeos en detalle, infografías y resúmenes interactivos.

La universidad online oficial de la NBA

TECH es la universidad online oficial de la NBA. Gracias a un acuerdo con la mayor liga de baloncesto, ofrece a sus alumnos programas universitarios exclusivos, así como una gran variedad de recursos educativos centrados en el negocio de la liga y otras áreas de la industria del deporte. Cada programa tiene un currículo de diseño único y cuenta con oradores invitados de excepción: profesionales con una distinguida trayectoria deportiva que ofrecerán su experiencia en los temas más relevantes.

Líderes en empleabilidad

TECH ha conseguido convertirse en la universidad líder en empleabilidad. El 99% de sus alumnos obtienen trabajo en el campo académico que ha estudiado, antes de completar un año luego de finalizar cualquiera de los programas de la universidad. Una cifra similar consigue mejorar su carrera profesional de forma inmediata. Todo ello gracias a una metodología de estudio que basa su eficacia en la adquisición de competencias prácticas, totalmente necesarias para el desarrollo profesional.



Google Partner Premier

El gigante tecnológico norteamericano ha otorgado a TECH la insignia Google Partner Premier. Este galardón, solo al alcance del 3% de las empresas del mundo, pone en valor la experiencia eficaz, flexible y adaptada que esta universidad proporciona al alumno. El reconocimiento no solo acredita el máximo rigor, rendimiento e inversión en las infraestructuras digitales de TECH, sino que también sitúa a esta universidad como una de las compañías tecnológicas más punteras del mundo.



La universidad mejor valorada por sus alumnos

Los alumnos han posicionado a TECH como la universidad mejor valorada del mundo en los principales portales de opinión, destacando su calificación más alta de 4,9 sobre 5, obtenida a partir de más de 1.000 reseñas. Estos resultados consolidan a TECH como la institución universitaria de referencia a nivel internacional, reflejando la excelencia y el impacto positivo de su modelo educativo.



03

Plan de estudios

El plan de estudios de este programa en Ingeniería de Software ha sido cuidadosamente diseñado por un equipo de expertos en informática y desarrollo tecnológico. Adicionalmente, a lo largo del programa, se abordan de manera integral los aspectos fundamentales de la Ingeniería de Software, cubriendo desde la planificación, diseño y desarrollo de aplicaciones hasta la seguridad informática avanzada y la administración eficiente de servidores. Los alumnos adquirirán habilidades técnicas especializadas y aprenderán a liderar proyectos complejos que respondan a las demandas de un mercado en constante evolución.





“

*Contribuye al avance tecnológico
desarrollando soluciones innovadoras
y efectivas en Ingeniería de Software”*

Módulo 1. Metodologías, desarrollo y calidad en la Ingeniería de Software

- 1.1. Introducción a la Ingeniería de Software
 - 1.1.1. Introducción
 - 1.1.2. La crisis del Software
 - 1.1.3. Diferencias entre la Ingeniería de Software y la ciencia de la computación
 - 1.1.4. Ética y responsabilidad profesional en la Ingeniería del Software
 - 1.1.5. Fábricas de Software
- 1.2. El proceso de desarrollo de Software
 - 1.2.1. Definición
 - 1.2.2. Modelo de proceso Software
 - 1.2.3. El proceso unificado de desarrollo de Software
- 1.3. Desarrollo de Software orientado a objetos
 - 1.3.1. Introducción
 - 1.3.2. Principios de la orientación a objetos
 - 1.3.3. Definición de objeto
 - 1.3.4. Definición de clase
 - 1.3.5. Análisis orientado a objetos vs diseño orientado a objetos
- 1.4. Desarrollo de Software basado en modelos
 - 1.4.1. La necesidad de modelar
 - 1.4.2. Modelado de sistemas Software
 - 1.4.3. Modelado de objetos
 - 1.4.4. UML
 - 1.4.5. Herramientas CASE
- 1.5. Modelado de aplicaciones y patrones de diseño con UML
 - 1.5.1. Modelado avanzado de requisitos
 - 1.5.2. Modelado estático avanzado
 - 1.5.3. Modelado dinámico avanzado
 - 1.5.4. Modelado de componentes
 - 1.5.5. Introducción a los patrones de diseño con UML
 - 1.5.6. Adapter
 - 1.5.7. Factory
 - 1.5.8. Singleton
 - 1.5.9. Strategy
 - 1.5.10. Composite
 - 1.5.11. Facade
 - 1.5.12. Observer
- 1.6. Ingeniería dirigida por modelos
 - 1.6.1. Introducción
 - 1.6.2. Metamodelado de sistemas
 - 1.6.3. MDA
 - 1.6.4. DSL
 - 1.6.5. Refinamientos de modelos con OCL
 - 1.6.6. Transformaciones de modelos
- 1.7. Ontologías en la Ingeniería de Software
 - 1.7.1. Introducción
 - 1.7.2. Ingeniería de la ontología
 - 1.7.3. Aplicación de las ontologías en la Ingeniería de Software
- 1.8. Metodologías ágiles para el desarrollo de Software, Scrum
 - 1.8.1. ¿Qué es la agilidad en el Software?
 - 1.8.2. El manifiesto ágil
 - 1.8.3. La hoja de ruta de un proyecto ágil
 - 1.8.4. El Product Owner
 - 1.8.5. Las historias de usuario
 - 1.8.6. Planificación y estimación ágil
 - 1.8.7. Mediciones en desarrollos ágiles
 - 1.8.8. Introducción al Scrum
 - 1.8.9. Los roles
 - 1.8.10. El product backlog
 - 1.8.11. El sprint
 - 1.8.12. Las reuniones
- 1.9. La metodología de desarrollo de Software Lean
 - 1.9.1. Introducción
 - 1.9.2. Kanban

- 1.10. Calidad y mejora del proceso Software
 - 1.10.1. Introducción
 - 1.10.2. Medición del Software
 - 1.10.3. Pruebas del Software
 - 1.10.4. Modelo de calidad de procesos Software: CMMI

Módulo 2. Gestión de proyectos de Software

- 2.1. Conceptos fundamentales de la dirección de proyectos y el ciclo de vida de la gestión de proyectos
 - 2.1.1. ¿Qué es un proyecto?
 - 2.1.2. Metodología común
 - 2.1.3. ¿Qué es la dirección/gestión de proyectos?
 - 2.1.4. ¿Qué es un plan de proyecto?
 - 2.1.5. Beneficios
 - 2.1.6. Ciclo de vida del proyecto
 - 2.1.7. Grupos de procesos o ciclo de vida de la gestión de los proyectos
 - 2.1.8. La relación entre los grupos de procesos y las áreas de conocimiento
 - 2.1.9. Relaciones entre el ciclo de vida del producto y del proyecto
- 2.2. El inicio y la planificación
 - 2.2.1. De la idea al proyecto
 - 2.2.2. Desarrollo del acta de proyecto
 - 2.2.3. Reunión de arranque del proyecto
 - 2.2.4. Tareas, conocimientos y habilidades en el proceso de inicio
 - 2.2.5. El plan de proyecto
 - 2.2.6. Desarrollo del plan básico. Pasos
 - 2.2.7. Tareas, conocimientos y habilidades en el proceso de planificación
- 2.3. La gestión de los stakeholders y del alcance
 - 2.3.1. Identificar a los interesados
 - 2.3.2. Desarrollar el plan para la gestión de los interesados
 - 2.3.3. Gestionar el compromiso de los interesados
 - 2.3.4. Controlar el compromiso de los interesados
 - 2.3.5. El objetivo del proyecto
 - 2.3.6. La gestión del alcance y su plan
 - 2.3.7. Recopilar los requisitos
 - 2.3.8. Definir el enunciado del alcance
 - 2.3.9. Crear la WBS (EDT)
 - 2.3.10. Verificar y controlar el alcance
- 2.4. El desarrollo del cronograma
 - 2.4.1. La gestión del tiempo y su plan
 - 2.4.2. Definir las actividades
 - 2.4.3. Establecimiento de la secuencia de las actividades
 - 2.4.4. Estimación de recursos de las actividades
 - 2.4.5. Estimación de la duración de las actividades
 - 2.4.6. Desarrollo del cronograma y cálculo del camino crítico
 - 2.4.7. Control del cronograma
- 2.5. El desarrollo del presupuesto y la respuesta a los riesgos
 - 2.5.1. Estimar los costes
 - 2.5.2. Desarrollar el presupuesto y la curva S
 - 2.5.3. Control de costes y método del valor ganado
 - 2.5.4. Los conceptos de riesgo
 - 2.5.5. Cómo hacer un análisis de riesgos
 - 2.5.6. El desarrollo del plan de respuesta
- 2.6. La gestión de la calidad
 - 2.6.1. Planificación de la calidad
 - 2.6.2. Aseguramiento de la calidad
 - 2.6.3. Control de la calidad
 - 2.6.4. Conceptos estadísticos básicos
 - 2.6.5. Herramientas de la gestión de la calidad
- 2.7. La comunicación y los recursos humanos
 - 2.7.1. Planificar la gestión de las comunicaciones
 - 2.7.2. Análisis de requisitos de comunicaciones
 - 2.7.3. Tecnología de las comunicaciones
 - 2.7.4. Modelos de comunicación

- 2.7.5. Métodos de comunicación
- 2.7.6. Plan de gestión de las comunicaciones
- 2.7.7. Gestionar las comunicaciones
- 2.7.8. La gestión de los recursos humanos
- 2.7.9. Principales actores y sus roles en los proyectos
- 2.7.10. Tipos de organizaciones
- 2.7.11. Organización del proyecto
- 2.7.12. El equipo de trabajo
- 2.8. El aprovisionamiento
 - 2.8.1. El proceso de adquisiciones
 - 2.8.2. Planificación
 - 2.8.3. Búsqueda de suministradores y solicitud de ofertas
 - 2.8.4. Adjudicación del contrato
 - 2.8.5. Administración del contrato
 - 2.8.6. Los contratos
 - 2.8.7. Tipos de contratos
 - 2.8.8. Negociación del contrato
- 2.9. Ejecución, monitorización y control y cierre
 - 2.9.1. Los grupos de procesos
 - 2.9.2. La ejecución del proyecto
 - 2.9.3. La monitorización y control del proyecto
 - 2.9.4. El cierre del proyecto
- 2.10. Responsabilidad profesional
 - 2.10.1. Responsabilidad profesional
 - 2.10.2. Características de la responsabilidad social y profesional
 - 2.10.3. Código deontológico del líder de proyectos
 - 2.10.4. Responsabilidad vs. PMP®
 - 2.10.5. Ejemplos de responsabilidad
 - 2.10.6. Beneficios de la profesionalización

Módulo 3. Plataformas de desarrollo del Software

- 3.1. Introducción al desarrollo de aplicaciones
 - 3.1.1. Aplicaciones de escritorio
 - 3.1.2. Lenguaje de programación
 - 3.1.3. Entornos de desarrollo integrado
 - 3.1.4. Aplicaciones web
 - 3.1.5. Aplicaciones móviles
 - 3.1.6. Aplicaciones en la nube
- 3.2. Desarrollo de aplicaciones e interfaz gráfica en Java
 - 3.2.1. Entornos de desarrollo integrados para Java
 - 3.2.2. Principales IDE para Java
 - 3.2.3. Introducción a la plataforma de desarrollo Eclipse
 - 3.2.4. Introducción a la plataforma de desarrollo NetBeans
 - 3.2.5. Modelo Vista Controlador para las interfaces gráficas de usuario
 - 3.2.6. Diseñar una interfaz gráfica en Eclipse
 - 3.2.7. Diseñar una interfaz gráfica en NetBeans
- 3.3. Depuración y pruebas en Java
 - 3.3.1. Pruebas y depuración de programas en Java
 - 3.3.2. Depuración en Eclipse
 - 3.3.3. Depuración en NetBeans
- 3.4. Desarrollo de aplicaciones e interfaz gráfica en .NET
 - 3.4.1. Net Framework
 - 3.4.2. Componentes de la plataforma de desarrollo .NET
 - 3.4.3. Visual Studio .NET
 - 3.4.4. Herramientas de .NET para GUI
 - 3.4.5. La GUI con Windows Presentation Foundation
 - 3.4.6. Depurar y compilar una aplicación de WPF

- 3.5. Programación para redes .NET
 - 3.5.1. Introducción a la programación para redes en .NET
 - 3.5.2. Peticiones y respuestas en .NET
 - 3.5.3. Uso de protocolos de aplicación en .NET
 - 3.5.4. Seguridad en la programación para redes en .NET
- 3.6. Entornos de desarrollo de aplicaciones móviles
 - 3.6.1. Aplicaciones móviles
 - 3.6.2. Aplicaciones móviles Android
 - 3.6.3. Pasos para el desarrollo en Android
 - 3.6.4. El IDE Android Studio
- 3.7. Desarrollo de aplicaciones en el entorno Android Studio
 - 3.7.1. Instalar e iniciar Android Studio
 - 3.7.2. Ejecución de una aplicación Android
 - 3.7.3. Desarrollo de la interfaz gráfica en Android Studio
 - 3.7.4. Iniciando actividades en Android Studio
- 3.8. Depuración y publicación de aplicaciones Android
 - 3.8.1. Depuración de una aplicación en Android Studio
 - 3.8.2. Memorizar aplicaciones en Android Studio
 - 3.8.3. Publicación de una aplicación en Google Play
- 3.9. Desarrollo de aplicaciones para la nube
 - 3.9.1. Cloud computing
 - 3.9.2. Niveles de cloud: SaaS, PaaS, IaaS
 - 3.9.3. Principales plataformas de desarrollo en la nube
 - 3.9.4. Referencias bibliográficas
- 3.10. Introducción a Google Cloud Platform
 - 3.10.1. Conceptos básicos de Google Cloud Platform
 - 3.10.2. Servicios de Google Cloud Platform
 - 3.10.3. Herramientas de Google Cloud Platform

Módulo 4. Computación en el cliente web

- 4.1. Introducción a HTML
 - 4.1.1. Estructura de un documento
 - 4.1.2. Color
 - 4.1.3. Texto
 - 4.1.4. Enlaces de hipertexto
 - 4.1.5. Imágenes
 - 4.1.6. Listas
 - 4.1.7. Tablas
 - 4.1.8. Marcos (frames)
 - 4.1.9. Formularios
 - 4.1.10. Elementos específicos para tecnologías móviles
 - 4.1.11. Elementos en desuso
- 4.2. Hojas de estilo web (CSS)
 - 4.2.1. Elementos y estructura de una hoja de estilos
 - 4.2.1.1. Creación de hojas de estilo
 - 4.2.1.2. Aplicación de estilos. Selectores
 - 4.2.1.3. Herencia de estilos y aplicación en cascada
 - 4.2.1.4. Formateado de páginas mediante estilos
 - 4.2.1.5. Estructura de páginas mediante estilos. El modelo de cajas
 - 4.2.2. Diseño de estilos para diferentes dispositivos
 - 4.2.3. Tipos de hojas de estilos: estáticas y dinámicas. Las pseudo-clases
 - 4.2.4. Buenas prácticas en el uso de hojas de estilo
- 4.3. Introducción e historia de JavaScript
 - 4.3.1. Introducción
 - 4.3.2. Historia de JavaScript
 - 4.3.3. Entorno de desarrollo que vamos a usar

- 4.4. Nociones básicas de programación web
 - 4.4.1. Sintaxis básica de JavaScript
 - 4.4.2. Tipos de datos primitivos y operadores
 - 4.4.3. Variables y ámbitos
 - 4.4.4. Cadenas de texto y template literals
 - 4.4.5. Números y booleanos
 - 4.4.6. Comparaciones
- 4.5. Estructuras complejas en JavaScript
 - 4.5.1. Vectores o arrays y objetos
 - 4.5.2. Conjuntos
 - 4.5.3. Mapas
 - 4.5.4. Disyuntivas
 - 4.5.5. Bucles
- 4.6. Funciones y objetos
 - 4.6.1. Definición e invocación de funciones
 - 4.6.2. Argumentos
 - 4.6.3. Funciones flecha
 - 4.6.4. Funciones de retrollamada o callback
 - 4.6.5. Funciones de orden superior
 - 4.6.6. Objetos literales
 - 4.6.7. El objeto this
 - 4.6.8. Objetos como espacios de nombres: el objeto Math y el objeto Date
- 4.7. El modelo de objetos del documento (DOM)
 - 4.7.1. ¿Qué es el DOM?
 - 4.7.2. Un poco de historia
 - 4.7.3. Navegación y obtención de elementos
 - 4.7.4. Un DOM virtual con JSDOM
 - 4.7.5. Selectores de consulta o query selectors
 - 4.7.6. Navegación mediante propiedades
 - 4.7.7. Asignación de atributos a los elementos
 - 4.7.8. Creación y modificación de nodos
 - 4.7.9. Actualización del estilo de los elementos del DOM
- 4.8. Desarrollo web moderno
 - 4.8.1. Flujo basado en eventos y listeners
 - 4.8.2. Toolkits web modernos y sistemas de alineamiento
 - 4.8.3. Modo estricto de JavaScript
 - 4.8.4. Algo más sobre funciones
 - 4.8.5. Promesas y funciones asíncronas
 - 4.8.6. Closures
 - 4.8.7. Programación funcional
 - 4.8.8. POO en JavaScript
- 4.9. Usabilidad web
 - 4.9.1. Introducción a la usabilidad
 - 4.9.2. Definición de usabilidad
 - 4.9.3. Importancia del diseño web centrado en el usuario
 - 4.9.4. Diferencias entre accesibilidad y usabilidad
 - 4.9.5. Ventajas y problemas en la combinación de accesibilidad y usabilidad
 - 4.9.6. Ventajas y dificultades en la implantación de sitios web usables
 - 4.9.7. Métodos de usabilidad
 - 4.9.8. Análisis de requerimiento de usuario
 - 4.9.9. Principios del diseño conceptual. Creación de prototipos orientados al usuario
 - 4.9.10. Pautas para la creación de sitios web usables
 - 4.9.10.1. Pautas de usabilidad de Jakob Nielsen
 - 4.9.10.2. Pautas de usabilidad de Bruce Tognazzini
 - 4.9.11. Evaluación de la usabilidad
- 4.10. Accesibilidad web
 - 4.10.1. Introducción
 - 4.10.2. Definición de accesibilidad web
 - 4.10.3. Tipos de discapacidades
 - 4.10.3.1. Discapacidades temporales o permanentes
 - 4.10.3.2. Discapacidades visuales
 - 4.10.3.3. Discapacidades auditivas
 - 4.10.3.4. Discapacidades motrices

- 4.10.3.5. Discapacidad neurológicas o cognitivas
- 4.10.3.6. Dificultades derivadas del envejecimiento
- 4.10.3.7. Limitaciones derivadas del entorno
- 4.10.3.8. Barreras que impiden el acceso a la web
- 4.10.4. Ayudas técnicas y productos de apoyo para superar las barreras
 - 4.10.4.1. Ayudas para personas ciegas
 - 4.10.4.2. Ayudas para persona con baja visión
 - 4.10.4.3. Ayudas para personas con daltonismo
 - 4.10.4.4. Ayudas para personas con discapacidad auditiva
 - 4.10.4.5. Ayudas para personas con discapacidad motriz
 - 4.10.4.6. Ayudas para personas con discapacidad cognitiva y neurológica
- 4.10.5. Ventajas y dificultades en la implantación de la accesibilidad web
- 4.10.6. Normativa y estándares sobre accesibilidad web
- 4.10.7. Organismos regulatorios de la accesibilidad web
- 4.10.8. Comparativa de normas y estándares
- 4.10.9. Guías para el cumplimiento de normativas y estándares
 - 4.10.9.1. Descripción de las pautas principales (imágenes, enlaces videos, etc.)
 - 4.10.9.2. Pautas para una navegación accesible
 - 4.10.9.2.1. Perceptibilidad
 - 4.10.9.2.2. Operatividad
 - 4.10.9.2.3. Comprensibilidad
 - 4.10.9.2.4. Robustez
- 4.10.10. Descripción del proceso de la conformidad en accesibilidad web
- 4.10.11. Niveles de conformidad
- 4.10.12. Criterios de conformidad
- 4.10.13. Requisitos de conformidad
- 4.10.14. Metodología de evaluación de la accesibilidad en sitios web

Módulo 5. Computación en servidor web

- 5.1. Introducción a la programación en el servidor: PHP
 - 5.1.1. Conceptos básicos de programación en el servidor
 - 5.1.2. Sintaxis básica de PHP
 - 5.1.3. Generación de contenido HTML con PHP
 - 5.1.4. Entornos de desarrollo y pruebas: XAMPP
- 5.2. PHP avanzado
 - 5.2.1. Estructuras de control con PHP
 - 5.2.2. Funciones en PHP
 - 5.2.3. Manejo de arrays en PHP
 - 5.2.4. Manejo de cadenas con PHP
 - 5.2.5. Orientación a objetos en PHP
- 5.3. Modelos de datos
 - 5.3.1. Concepto de dato. Ciclo de vida de los datos
 - 5.3.2. Tipos de datos
 - 5.3.2.1. Básicos
 - 5.3.2.2. Registros
 - 5.3.2.3. Dinámicos
- 5.4. El modelo relacional
 - 5.4.1. Descripción
 - 5.4.2. Entidades y tipos de entidades
 - 5.4.3. Elementos de datos. Atributos
 - 5.4.4. Relaciones: tipos, subtipos, cardinalidad
 - 5.4.5. Claves. Tipos de claves
 - 5.4.6. Normalización. Formas normales
- 5.5. Construcción del modelo lógico de datos
 - 5.5.1. Especificación de tablas
 - 5.5.2. Definición de columnas
 - 5.5.3. Especificación de claves
 - 5.5.4. Conversión a formas normales. Dependencias

- 5.6. El modelo físico de datos. Ficheros de datos
 - 5.6.1. Descripción de los ficheros de datos
 - 5.6.2. Tipos de ficheros
 - 5.6.3. Modos de acceso
 - 5.6.4. Organización de ficheros
- 5.7. Acceso a bases de datos desde PHP
 - 5.7.1. Introducción a MariaDB
 - 5.7.2. Trabajar con una base de datos MariaDB: el lenguaje SQL
 - 5.7.3. Acceder a la base de datos MariaDB desde PHP
 - 5.7.4. Introducción a MySQL
 - 5.7.5. Trabajar con una base de datos MySQL: el lenguaje SQL
 - 5.7.6. Acceder a la base de datos MySQL desde PHP
- 5.8. Interacción con el cliente desde PHP
 - 5.8.1. Formularios PHP
 - 5.8.2. Cookies
 - 5.8.3. Manejo de sesiones
- 5.9. Arquitectura de aplicaciones web
 - 5.9.1. El patrón Modelo Vista Controlador
 - 5.9.2. Controlador
 - 5.9.3. Modelo
 - 5.9.4. Vista
- 5.10. Introducción a los servicios web
 - 5.10.1. Introducción a XML
 - 5.10.2. Arquitecturas orientas a servicios (SOA): servicios web
 - 5.10.3. Creación de servicios web SOAP y REST
 - 5.10.4. El protocolo SOAP
 - 5.10.5. El protocolo REST

Módulo 6. Gestión de la seguridad

- 6.1. La seguridad de la información
 - 6.1.1. Introducción
 - 6.1.2. La seguridad de la información implica la confidencialidad, integridad y disponibilidad
 - 6.1.3. La seguridad es un asunto económico
 - 6.1.4. La seguridad es un proceso
 - 6.1.5. La clasificación de la información
 - 6.1.6. La seguridad en la información implica la gestión de los riesgos
 - 6.1.7. La seguridad se articula con controles de seguridad
 - 6.1.8. La seguridad es tanto física como lógica
 - 6.1.9. La seguridad implica a las personas
- 6.2. El profesional de la seguridad de la información
 - 6.2.1. Introducción
 - 6.2.2. La seguridad de la información como profesión
 - 6.2.3. Las certificaciones (ISC)2
 - 6.2.4. El estándar ISO 27001
 - 6.2.5. Buenas prácticas de seguridad en la gestión de servicios TI
 - 6.2.6. Modelos de madurez para la seguridad de la información
 - 6.2.7. Otras certificaciones, estándares y recursos profesionales
- 6.3. Control de accesos
 - 6.3.1. Introducción
 - 6.3.2. Requisitos del control de accesos
 - 6.3.3. Mecanismos de autenticación
 - 6.3.4. Métodos de autorización
 - 6.3.5. Contabilidad y auditoría de accesos
 - 6.3.6. Tecnologías «Triple A»

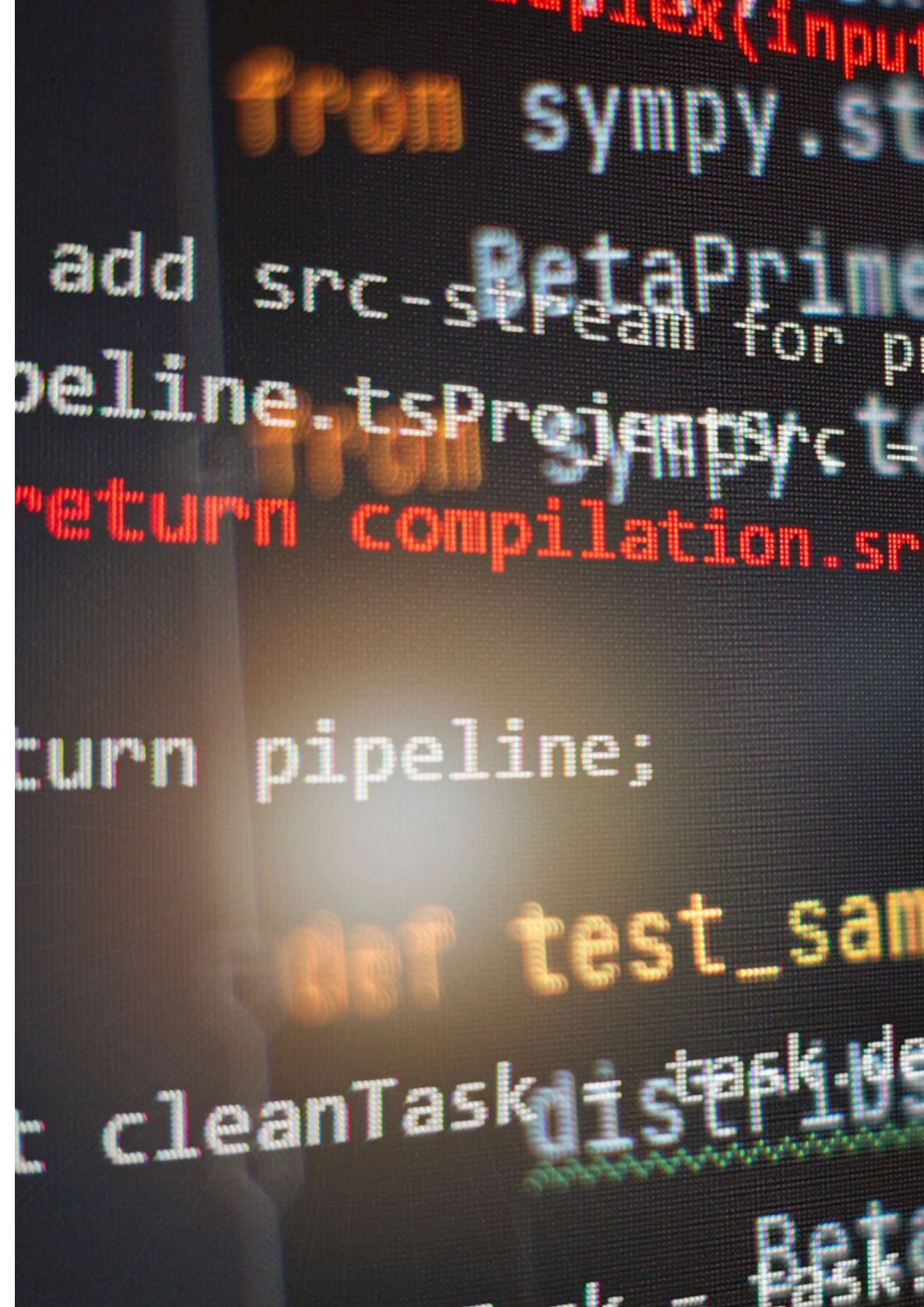
- 6.4. Programas, procesos y políticas de seguridad de la información
 - 6.4.1. Introducción
 - 6.4.2. Programas de gestión de la seguridad
 - 6.4.3. La gestión de riesgos
 - 6.4.4. Diseño de políticas de seguridad
- 6.5. Planes de continuidad de negocio
 - 6.5.1. Introducción a los PCN
 - 6.5.2. Fase I y II
 - 6.5.3. Fase III y IV
 - 6.5.4. Mantenimiento del PCN
- 6.6. Procedimientos para a correcta protección de la empresa
 - 6.6.1. Redes DMZ
 - 6.6.2. Sistemas de detección de intrusos
 - 6.6.3. Listas de control de accesos
 - 6.6.4. Aprender del atacante: Honeypot
- 6.7. Arquitectura de seguridad. Prevención
 - 6.7.1. Visión general. Actividades y modelo de capas
 - 6.7.2. Defensa perimetral (firewalls, WAFs, IPS, etc.)
 - 6.7.3. Defensa del punto final (equipos, servidores y servicios)
- 6.8. Arquitectura de seguridad. Detección
 - 6.8.1. Visión general detección y supervisión
 - 6.8.2. Logs, ruptura de tráfico cifrado, grabación y Siems
 - 6.8.3. Alertas e inteligencia
- 6.9. Arquitectura de seguridad. Reacción
 - 6.9.1. Reacción. Productos, servicios y recursos
 - 6.9.2. Gestión de Incidentes
 - 6.9.3. CERTS y CSIRTs

- 6.10. Arquitectura de seguridad. Recuperación
 - 6.10.1. Resiliencia, Conceptos, Requerimientos de Negocio y Normativa
 - 6.10.2. Soluciones IT de Resiliencia
 - 6.10.3. Gestión y Gobierno de las Crisis

Módulo 7. Seguridad en los sistemas de información

- 7.1. Una perspectiva global de la seguridad, la criptografía y los criptoanálisis clásicos
 - 7.1.1. La seguridad informática: Perspectiva histórica
 - 7.1.2. Pero, ¿Qué se entiende exactamente por seguridad?
 - 7.1.3. Historia de la criptografía
 - 7.1.4. Cifradores de sustitución
 - 7.1.5. Caso de estudio: La máquina Enigma
- 7.2. Criptografía simétrica
 - 7.2.1. Introducción y terminología básica
 - 7.2.2. Cifrado simétrico
 - 7.2.3. Modos de operación
 - 7.2.4. DES
 - 7.2.5. El nuevo estándar AES
 - 7.2.6. Cifrado en flujo
 - 7.2.7. Criptoanálisis
- 7.3. Criptografía asimétrica
 - 7.3.1. Orígenes de la criptografía de clave pública
 - 7.3.2. Conceptos básicos y funcionamiento
 - 7.3.3. El algoritmo RSA
 - 7.3.4. Certificados digitales
 - 7.3.5. Almacenamiento y gestión de claves

- 7.4. Ataques en redes
 - 7.4.1. Amenazas y ataques de una red
 - 7.4.2. Enumeración
 - 7.4.3. Interceptación de tráfico: sniffers
 - 7.4.4. Ataques de denegación de servicio
 - 7.4.5. Ataques de envenenamiento ARP
- 7.5. Arquitecturas de seguridad
 - 7.5.1. Arquitecturas de seguridad tradicionales
 - 7.5.2. Secure Socket Layer: SSL
 - 7.5.3. Protocolo SSH
 - 7.5.4. Redes Privadas Virtuales (VPNs)
 - 7.5.5. Mecanismos de protección de unidades de almacenamiento externo
 - 7.5.6. Mecanismos de protección hardware
- 7.6. Técnicas de protección de sistemas y desarrollo de código seguro
 - 7.6.1. Seguridad en Operaciones
 - 7.6.2. Recursos y controles
 - 7.6.3. Monitorización
 - 7.6.4. Sistemas de detección de intrusión
 - 7.6.5. IDS de host
 - 7.6.6. IDS de red
 - 7.6.7. IDS basados en firmas
 - 7.6.8. Sistemas señuelos
 - 7.6.9. Principios de seguridad básicos en el desarrollo de código
 - 7.6.10. Gestión del fallo
 - 7.6.11. Enemigo público número 1: el desbordamiento de búfer
 - 7.6.12. Chapuzas criptográficas



- 7.7. Botnets y spam
 - 7.7.1. Origen del problema
 - 7.7.2. Proceso del spam
 - 7.7.3. Envío del spam
 - 7.7.4. Refinamiento de las listas de direcciones de correo
 - 7.7.5. Técnicas de protección
 - 7.7.6. Servicio anti-spam ofrecidos por terceros
 - 7.7.7. Casos de estudio
 - 7.7.8. Spam exótico
- 7.8. Auditoría y ataques Web
 - 7.8.1. Recopilación de información
 - 7.8.2. Técnicas de ataque
 - 7.8.3. Herramientas
- 7.9. Malware y código malicioso
 - 7.9.1. ¿Qué es el malware?
 - 7.9.2. Tipos de malware
 - 7.9.3. Virus
 - 7.9.4. Criptovirus
 - 7.9.5. Gusanos
 - 7.9.6. Adware
 - 7.9.7. Spyware
 - 7.9.8. Hoaxes
 - 7.9.9. Pishing
 - 7.9.10. Troyanos
 - 7.9.11. La economía del malware
 - 7.9.12. Posibles soluciones
- 7.10. Análisis forense
 - 7.10.1. Recolección de evidencias
 - 7.10.2. Análisis de las evidencias
 - 7.10.3. Técnicas anti-forenses
 - 7.10.4. Caso de estudio práctico

Módulo 8. Seguridad en el Software

- 8.1. Problemas de la seguridad en el Software
 - 8.1.1. Introducción al problema de la seguridad en el Software
 - 8.1.2. Vulnerabilidades y su clasificación
 - 8.1.3. Propiedades Software seguro
 - 8.1.4. Referencias
- 8.2. Principios de diseño seguridad del Software
 - 8.2.1. Introducción
 - 8.2.2. Principios de diseño seguridad del Software
 - 8.2.3. Tipos de S-SDLC
 - 8.2.4. Seguridad del Software en las fases del S-SDLC
 - 8.2.5. Metodologías y estándares
 - 8.2.6. Referencias
- 8.3. Seguridad en el ciclo de vida del Software en las fases de requisitos y diseño
 - 8.3.1. Introducción
 - 8.3.2. Modelado de ataques
 - 8.3.3. Casos de abuso
 - 8.3.4. Ingeniería de requisitos de seguridad
 - 8.3.5. Análisis de riesgo. Arquitectónico
 - 8.3.6. Patrones de diseño
 - 8.3.7. Referencias
- 8.4. Seguridad en el ciclo de vida del Software en las fases de codificación, pruebas y operación
 - 8.4.1. Introducción
 - 8.4.2. Pruebas de seguridad basadas en riesgo
 - 8.4.3. Revisión de código
 - 8.4.4. Test de penetración
 - 8.4.5. Operaciones de seguridad
 - 8.4.6. Revisión externa
 - 8.4.7. Referencias
- 8.5. Codificación segura aplicaciones I
 - 8.5.1. Introducción
 - 8.5.2. Prácticas de codificación segura
 - 8.5.3. Manipulación y validación de entradas
 - 8.5.4. Desbordamiento de memoria
 - 8.5.5. Referencias
- 8.6. Codificación segura aplicaciones II
 - 8.6.1. Introducción
 - 8.6.2. Integers overflows, errores de truncado y problemas con conversiones de tipo entre números enteros
 - 8.6.3. Errores y excepciones
 - 8.6.4. Privacidad y confidencialidad
 - 8.6.5. Programas privilegiados
 - 8.6.6. Referencias
- 8.7. Seguridad en el desarrollo y en la nube
 - 8.7.1. Seguridad en el desarrollo; metodología y práctica
 - 8.7.2. Modelos PaaS, IaaS, CaaS y SaaS
 - 8.7.3. Seguridad en la nube y para servicios en la nube
- 8.8. Automatización y orquestación de seguridad (SOAR)
 - 8.8.1. Complejidad del tratamiento manual; necesidad de automatizar las tareas
 - 8.8.2. Productos y servicios
 - 8.8.3. Arquitectura SOAR
- 8.9. Seguridad en el Teletrabajo
 - 8.9.1. Necesidad y escenarios
 - 8.9.2. Productos y servicios
 - 8.9.3. Seguridad en el teletrabajo

Módulo 9. Calidad y auditoría de sistemas de información

- 9.1. Introducción a los Sistemas de Gestión de Seguridad de la Información
 - 9.1.1. Principios fundamentales de los SGSI
 - 9.1.2. Reglas de oro de los SGSI
 - 9.1.3. Papel de la auditoría informática en los SGSI
- 9.2. Planificación en la gestión de la seguridad
 - 9.2.1. Conceptos relativos a la gestión de la seguridad
 - 9.2.2. Clasificación de la información: objetivos, conceptos y roles
 - 9.2.3. Implementación de las políticas de seguridad: Políticas de seguridad, estándares y procedimientos
 - 9.2.4. Gestión del riesgo: Principios y análisis del riesgo de los activos de información
- 9.3. Principales mecanismos para la protección de activos información (I)
 - 9.3.1. Resumen de las principales herramientas criptográficas para la protección de la triada CID
 - 9.3.2. Consideración de los requisitos de privacidad, anonimato y gestión adecuada de la trazabilidad de usuarios
- 9.4. Principales mecanismos para la protección de activos información (II)
 - 9.4.1. Seguridad de las comunicaciones: protocolos, dispositivos y arquitecturas de seguridad
 - 9.4.2. Seguridad de los sistemas operativos
- 9.5. Controles internos de los SGSI
 - 9.5.1. Taxonomía de los controles SGSI: controles administrativos, lógicos y físicos
 - 9.5.2. Clasificación de los controles en función del modo de abordar la amenaza: Controles para la prevención, la detección y la corrección de amenazas
 - 9.5.3. Implantación de sistemas de control interno en los SGSI
- 9.6. Tipos de auditoría
 - 9.6.1. Diferencia entre auditoría y control interno
 - 9.6.2. Auditoría interna frente a auditoría externa
 - 9.6.3. Clasificación de la auditoría en función del objetivo y el tipo de análisis
- 9.7. Guionista y guion: Sujeto y objeto protegido por la Propiedad Intelectual
 - 9.7.1. Introducción a los test de penetración y al análisis forense
 - 9.7.2. Definición y relevancia de los conceptos de fingerprinting y footprinting

- 9.8. Análisis de vulnerabilidades y monitorización de tráfico de red
 - 9.8.1. Herramientas para el análisis de vulnerabilidades en sistemas
 - 9.8.2. Principales vulnerabilidades en el contexto de las aplicaciones web
 - 9.8.3. Análisis de protocolos de comunicaciones
- 9.9. El proceso de la auditoría informática
 - 9.9.1. Concepto de ciclo de vida en el desarrollo de sistemas
 - 9.9.2. Monitorización de actividad y de procesos: recolección y tratamiento de evidencias
 - 9.9.3. Metodología de la auditoría informática
 - 9.9.4. Proceso de una auditoría informática
 - 9.9.5. Identificación de los principales delitos y faltas en el contexto de las tecnologías de la información
 - 9.9.6. Investigación de delitos informáticos: introducción al análisis forense y su relación con la auditoría informática
- 9.10. Planes de continuidad de negocio y de recuperación frente a desastres
 - 9.10.1. Definición de plan de continuidad de negocio y del concepto de interrupción del negocio
 - 9.10.2. Recomendación NIST sobre los planes de continuidad de negocio
 - 9.10.3. Plan de recuperación ante desastres
 - 9.10.4. Proceso de plan de recuperación ante desastres

Módulo 10. Administración de servidores web

- 10.1. Introducción a servidores web
 - 10.1.1. ¿Que es un servidor web?
 - 10.1.2. Arquitectura y funcionamiento de un servidor web
 - 10.1.3. Recursos y contenidos en un servidor web
 - 10.1.4. Servidores de aplicaciones
 - 10.1.5. Servidores proxy
 - 10.1.6. Principales servidores web del mercado
 - 10.1.7. Estadística de uso servidores web
 - 10.1.8. Seguridad en servidores web
 - 10.1.9. Balanceo de carga en servidores web
 - 10.1.10. Referencias

- 10.2. Manejo del protocolo HTTP
 - 10.2.1. Funcionamiento y estructura
 - 10.2.2. Descripción de peticiones o request methods
 - 10.2.3. Códigos de estado
 - 10.2.4. Cabeceras
 - 10.2.5. Codificación del contenido. Páginas de códigos
 - 10.2.6. Realización de peticiones HTTP en Internet mediante un proxy, livehttpheaders o método similar, analizando el protocolo utilizado
- 10.3. Descripción de arquitecturas distribuidas en múltiples servidores
 - 10.3.1. Modelo de 3 capas
 - 10.3.2. Tolerancia a fallos
 - 10.3.3. Reparto de carga
 - 10.3.4. Almacenes de estado de sesión
 - 10.3.5. Almacenes de caché
- 10.4. Internet Information Services (IIS)
 - 10.4.1. ¿Que es IIS?
 - 10.4.2. Historia y evolución de IIS
 - 10.4.3. Principales ventajas y características de IIS7 y posteriores
 - 10.4.4. Arquitectura IIS7 y posteriores
- 10.5. Instalación, administración y configuración de IIS
 - 10.5.1. Preámbulo
 - 10.5.2. Instalación de Internet Information Services (IIS)
 - 10.5.3. Herramientas de administración de IIS
 - 10.5.4. Creación, configuración y administración de sitios web
 - 10.5.5. Instalación y manejo de extensiones en IIS
- 10.6. Seguridad avanzada en IIS
 - 10.6.1. Preámbulo
 - 10.6.2. Autenticación, autorización, y control de Acceso en IIS
 - 10.6.3. Configuración de un sitio web seguro en IIS con SSL
 - 10.6.4. Políticas de seguridad implementada en IIS 8.x

- 10.7. Introducción a Apache
 - 10.7.1. ¿Qué es Apache?
 - 10.7.2. Principales ventajas de Apache
 - 10.7.3. Características principales de Apache
 - 10.7.4. Arquitectura
- 10.8. Instalación y configuración de Apache
 - 10.8.1. Instalación inicial de apache
 - 10.8.2. Configuración de apache
- 10.9. Instalación y configuración de los diferentes módulos en Apache
 - 10.9.1. Instalación de módulos en Apache
 - 10.9.2. Tipos de módulos
 - 10.9.3. Configuración segura de Apache
- 10.10. Seguridad avanzada
 - 10.10.1. Autenticación, autorización y control de acceso
 - 10.10.2. Métodos de autenticación
 - 10.10.3. Configuración segura de Apache con SSL

Módulo 11. Seguridad en aplicaciones online

- 11.1. Vulnerabilidades y problemas de seguridad en las aplicaciones online
 - 11.1.1. Introducción a la seguridad en las aplicaciones online
 - 11.1.2. Vulnerabilidades de seguridad en el diseño de las aplicaciones web
 - 11.1.3. Vulnerabilidades de seguridad en la implementación de las aplicaciones web
 - 11.1.4. Vulnerabilidades de seguridad en el despliegue de las aplicaciones web
 - 11.1.5. Listas oficiales de vulnerabilidades de seguridad
- 11.2. Políticas y estándares para la seguridad de las aplicaciones online
 - 11.2.1. Pilares para la seguridad de las aplicaciones online
 - 11.2.2. Política de seguridad
 - 11.2.3. Sistema de gestión de seguridad de la información
 - 11.2.4. Ciclo de vida de desarrollo seguro de Software
 - 11.2.5. Estándares para la seguridad de las aplicaciones

- 11.3. Seguridad en el diseño de las aplicaciones web
 - 11.3.1. Introducción a la seguridad de las aplicaciones web
 - 11.3.2. Seguridad en el diseño de las aplicaciones web
- 11.4. Test de la seguridad y protección online de las aplicaciones web
 - 11.4.1. Análisis y test de la seguridad de las aplicaciones web
 - 11.4.2. Seguridad en el despliegue y producción de las aplicaciones web
- 11.5. Seguridad de los servicios web
 - 11.5.1. Introducción a la seguridad de los servicios web
 - 11.5.2. Funciones y tecnologías de la seguridad de los servicios web
- 11.6. Test de la seguridad y protección online de los servicios web
 - 11.6.1. Evaluación de la seguridad de los servicios web
 - 11.6.2. Protección online. Firewalls y gateways XML
- 11.7. Hacking ético, malware y forensic
 - 11.7.1. Hacking ético
 - 11.7.2. Análisis de Malware
 - 11.7.3. Análisis Forense
- 11.8. Resolución de incidentes sobre servicios web
 - 11.8.1. Monitorización
 - 11.8.2. Herramientas de medición del rendimiento
 - 11.8.3. Medidas de contención
 - 11.8.4. Análisis causa – raíz
 - 11.8.5. Gestión proactiva de problemas
- 11.9. Buenas prácticas para garantizar la seguridad en las aplicaciones
 - 11.9.1. Manual de buenas prácticas en el desarrollo de las aplicaciones online
 - 11.9.2. Manual de buenas prácticas en la implementación de las aplicaciones online
- 11.10. Errores comunes que perjudican la seguridad de las aplicaciones
 - 11.10.1. Errores comunes en el desarrollo
 - 11.10.2. Errores comunes en el hospedaje
 - 11.10.3. Errores comunes en la producción

Módulo 12. Ingeniería del Software

- 12.1. Introducción a la Ingeniería del Software y al modelado
 - 12.1.1. La naturaleza del Software
 - 12.1.2. La naturaleza única de las webapps
 - 12.1.3. Ingeniería del Software
 - 12.1.4. El proceso del Software
 - 12.1.5. La práctica de la Ingeniería del Software
 - 12.1.6. Mitos del Software
 - 12.1.7. Cómo comienza todo
 - 12.1.8. Conceptos orientados a objetos
 - 12.1.9. Introducción a UML
- 12.2. El proceso del Software
 - 12.2.1. Un modelo general de proceso
 - 12.2.2. Modelos de proceso prescriptivos
 - 12.2.3. Modelos de proceso especializado
 - 12.2.4. El proceso unificado
 - 12.2.5. Modelos del proceso personal y del equipo
 - 12.2.6. ¿Qué es la agilidad?
 - 12.2.7. ¿Qué es un proceso ágil?
 - 12.2.8. Scrum
 - 12.2.9. Conjunto de herramientas para el proceso ágil
- 12.3. Principios que guían la práctica de la Ingeniería del Software
 - 12.3.1. Principios que guían el proceso
 - 12.3.2. Principios que guían la práctica
 - 12.3.3. Principios de comunicación
 - 12.3.4. Principios de planificación
 - 12.3.5. Principios de modelado
 - 12.3.6. Principios de construcción
 - 12.3.7. Principios de despliegue

- 12.4. Comprensión de los requisitos
 - 12.4.1. Ingeniería de requisitos
 - 12.4.2. Establecer las bases
 - 12.4.3. Indagación de los requisitos
 - 12.4.4. Desarrollo de casos de uso
 - 12.4.5. Elaboración del modelo de los requisitos
 - 12.4.6. Negociación de los requisitos
 - 12.4.7. Validación de los requisitos
- 12.5. Modelado de los requisitos: Escenarios, información y clases de análisis
 - 12.5.1. Análisis de los requisitos
 - 12.5.2. Modelado basado en escenarios
 - 12.5.3. Modelos UML que proporcionan el caso de uso
 - 12.5.4. Conceptos de modelado de datos
 - 12.5.5. Modelado basado en clases
 - 12.5.6. Diagramas de clases
- 12.6. Modelado de los requisitos: Flujo, comportamiento y patrones
 - 12.6.1. Requisitos que modelan las estrategias
 - 12.6.2. Modelado orientado al flujo
 - 12.6.3. Diagramas de estado
 - 12.6.4. Creación de un modelo de comportamiento
 - 12.6.5. Diagramas de secuencia
 - 12.6.6. Diagramas de comunicación
 - 12.6.7. Patrones para el modelado de requisitos
- 12.7. Conceptos de diseño
 - 12.7.1. Diseño en el contexto de la Ingeniería del Software
 - 12.7.2. El proceso de diseño
 - 12.7.3. Conceptos de diseño
 - 12.7.4. Conceptos de diseño orientado a objetos
 - 12.7.5. El modelo del diseño
- 12.8. Diseño de la arquitectura
 - 12.8.1. Arquitectura del Software
 - 12.8.2. Géneros arquitectónicos
 - 12.8.3. Estilos arquitectónicos
 - 12.8.4. Diseño arquitectónico
 - 12.8.5. Evolución de los diseños alternativos para la arquitectura
 - 12.8.6. Mapeo de la arquitectura con el uso del flujo de datos
- 12.9. Diseño en el nivel de componentes y basado en patrones
 - 12.9.1. ¿Qué es un componente?
 - 12.9.2. Diseño de componentes basados en clase
 - 12.9.3. Realización del diseño en el nivel de componentes
 - 12.9.4. Diseño de componentes tradicionales
 - 12.9.5. Desarrollo basado en componentes
 - 12.9.6. Patrones de diseño
 - 12.9.7. Diseño de Software basado en patrones
 - 12.9.8. Patrones arquitectónicos
 - 12.9.9. Patrones de diseño en el nivel de componentes
 - 12.9.10. Patrones de diseño de la interfaz de usuario
- 12.10. Calidad del Software y administración de proyectos
 - 12.10.1. Calidad
 - 12.10.1. Calidad del Software
 - 12.10.2. El dilema de la calidad del Software
 - 12.10.3. Lograr la calidad del Software
 - 12.10.4. Aseguramiento de la calidad del Software
 - 12.10.5. El espectro administrativo
 - 12.10.6. El personal
 - 12.10.7. El producto
 - 12.10.8. El proceso
 - 12.10.9. El proyecto
 - 12.10.10. Principios y prácticas

Módulo 13. Ingeniería del Software avanzada

- 13.1. Programación extrema
 - 13.1.1. Justificación y visión general de XP
 - 13.1.2. El ciclo de vida en XP
 - 13.1.3. Los cinco valores básicos
 - 13.1.4. Las doce prácticas básicas en XP
 - 13.1.5. Roles de los participantes
 - 13.1.6. XP Industrial
 - 13.1.7. Valoración crítica de XP
- 13.2. Desarrollo de Software basado en reutilización
 - 13.2.1. La reutilización del Software
 - 13.2.2. Niveles de reutilización de código
 - 13.2.3. Técnicas concretas de reutilización
 - 13.2.4. Desarrollo basado en componentes
 - 13.2.5. Beneficios y problemas de la reutilización
 - 13.2.6. Planificación de la reutilización
- 13.3. Patrones de arquitectura de sistemas y de diseño de Software
 - 13.3.1. El diseño arquitectónico
 - 13.3.2. Patrones arquitectónicos generales
 - 13.3.3. Arquitecturas tolerantes a fallos
 - 13.3.4. Arquitecturas de sistemas distribuidos
 - 13.3.5. Los patrones de diseño
 - 13.3.6. Patrones de Gamma
 - 13.3.7. Patrones de diseño de interacción
- 13.4. Arquitectura de aplicaciones en la nube
 - 13.4.1. Fundamentos de Cloud Computing
 - 13.4.2. Calidad de las aplicaciones en la nube
 - 13.4.3. Estilos de arquitectura
 - 13.4.4. Patrones de diseño
- 13.5. Pruebas del Software: TDD, ATDD y BDD
 - 13.5.1. Verificación y validación del Software
 - 13.5.2. Las pruebas de Software
 - 13.5.3. Test Driven Development (TDD)
 - 13.5.4. Acceptance Test Driven Development (ATDD)
 - 13.5.5. Behavior Driven Development (BDD)
 - 13.5.6. BDD y Cucumber
- 13.6. La mejora del proceso de Software
 - 13.6.1. La mejora del proceso de Software
 - 13.6.2. El proceso de mejora de procesos
 - 13.6.3. Modelos de madurez
 - 13.6.4. El modelo CMMI
 - 13.6.5. CMMI V2.0.
 - 13.6.6. CMMI y Ágil
- 13.7. La calidad del producto Software: SQuaRE
 - 13.7.1. La calidad del Software
 - 13.7.2. Modelos de calidad del producto Software
 - 13.7.3. Familia ISO/IEC 25000
 - 13.7.4. ISO/IEC 25010: modelo y características de calidad
 - 13.7.5. ISO/IEC 25012: la calidad de los datos
 - 13.7.6. ISO/IEC 25020: medición de la calidad del Software
 - 13.7.7. ISO/IEC 25022, 25023 y 25024: métricas de calidad del Software y de los datos
 - 13.7.8. ISO/IEC 25040: evaluación del Software
 - 13.7.9. El proceso de certificación
- 13.8. Introducción a DevOps
 - 13.8.1. Concepto de DevOps
 - 13.8.2. Prácticas principales

Módulo 14. Ingeniería de requisitos

- 14.1. Introducción a la Ingeniería de requisitos
 - 14.1.1. La importancia de los requisitos
 - 14.1.2. Concepto de requisito
 - 14.1.3. Dimensiones de los requisitos
 - 14.1.4. Niveles y tipos de requisitos
 - 14.1.5. Características de los requisitos
 - 14.1.6. La Ingeniería de requisitos
 - 14.1.7. El proceso de Ingeniería de requisitos
 - 14.1.8. Frameworks para Ingeniería de requisitos
 - 14.1.9. Buenas prácticas en Ingeniería de requisitos
 - 14.1.10. El analista de negocio
- 14.2. Las fuentes de los requisitos
 - 14.2.1. La red de requisitos
 - 14.2.2. Los stakeholders
 - 14.2.3. Los requisitos de negocio
 - 14.2.4. Documento de visión y alcance
- 14.3. Técnicas de elicitación de requisitos
 - 14.3.1. La elicitación de requisitos
 - 14.3.2. Problemas de la elicitación de requisitos
 - 14.3.3. Contextos de descubrimiento
 - 14.3.4. Entrevistas
 - 14.3.5. Observación y «aprendizaje»
 - 14.3.6. Etnografía
 - 14.3.7. Workshops
 - 14.3.8. Focus groups
 - 14.3.9. Cuestionarios
 - 14.3.10. Brainstorming y técnicas creativas
 - 14.3.11. Medios grupales
 - 14.3.12. Análisis de interfaces del sistema
 - 14.3.13. Análisis de documentos y «arqueología»
 - 14.3.14. Casos de uso y escenarios
 - 14.3.15. Los prototipos
 - 14.3.16. La Ingeniería inversa
 - 14.3.17. Reutilización de requisitos
 - 14.3.18. Buenas prácticas de la elicitación
- 14.4. Requisitos de los usuarios
 - 14.4.1. Personas
 - 14.4.2. Casos de uso e historias de usuario
 - 14.4.3. Escenarios
 - 14.4.5. Tipos de escenarios
 - 14.4.6. Cómo descubrir escenarios
- 14.5. Técnicas de Prototipado
 - 14.5.1. El prototipado
 - 14.5.2. Prototipos según su alcance
 - 14.5.3. Prototipos según su temporalidad
 - 14.5.4. La fidelidad de un prototipo
 - 14.5.5. Prototipos de interfaz de usuario
 - 14.5.6. Evaluación de prototipos
- 14.6. Análisis de requisitos
 - 14.6.1. El análisis de requisitos
 - 14.6.2. Buenas prácticas del análisis de requisitos
 - 14.6.3. El diccionario de datos
 - 14.6.4. Priorización de requisitos
- 14.7. Documentación de los requisitos
 - 14.7.1. El documento especificación de requisitos
 - 14.7.2. Estructura y contenidos de un SRS
 - 14.7.3. Documentación en lenguaje natural
 - 14.7.4. EARS: Easy Approach to Requirements Syntax
 - 14.7.5. Los requisitos no funcionales
 - 14.7.6. Atributos y plantillas en forma de tabla
 - 14.7.7. Buenas prácticas de especificación

- 14.8. Validación y negociación de requisitos
 - 14.8.1. Validación de requisitos
 - 14.8.2. Técnicas de validación de requisitos
 - 14.8.3. Negociación de requisitos
- 14.9. Modelado y gestión de requisitos
 - 14.9.1. El modelado de requisitos
 - 14.9.2. La perspectiva del usuario
 - 14.9.3. La perspectiva de los datos
 - 14.9.4. La perspectiva funcional u orientada al flujo
 - 14.9.5. La perspectiva del comportamiento
 - 14.9.6. La volatilidad de los requisitos
 - 14.9.7. Proceso de gestión de requisitos
 - 14.9.8. Herramientas para gestión de requisitos
 - 14.9.9. Buenas prácticas en la gestión de requisitos
- 14.10. Sistemas críticos y especificación formal
 - 14.10.1. Los sistemas críticos
 - 14.10.2. Especificación dirigida por riesgos
 - 14.10.3. Especificación formal

Módulo 15. Procesos de Ingeniería del Software

- 15.1. Marco de Ingeniería Software
 - 15.1.1. Características del Software
 - 15.1.2. Los procesos principales en Ingeniería del Software
 - 15.1.3. Modelos de proceso de desarrollo Software
 - 15.1.4. Marco de referencia estándar para el proceso de desarrollo de Software: La norma ISO/IEC 12207
- 15.2. Proceso Unificado de desarrollo Software
 - 15.2.1. Proceso Unificado
 - 15.2.2. Dimensiones del Proceso Unificado
 - 15.2.3. Proceso de desarrollo dirigido por casos de uso
 - 15.2.4. Flujos de trabajo fundamentales de Procesos Unificados
- 15.3. Planificación en el contexto de desarrollo de Software ágil
 - 15.3.1. Características del desarrollo Software ágil
 - 15.3.2. Diferentes horizontes temporales de planificación en el desarrollo ágil
 - 15.3.3. Marco de desarrollo ágil Scrum y horizontes temporales de planificación
 - 15.3.4. Historias de usuario como unidad de planificación y estimación
 - 15.3.5. Técnicas comunes para derivar una estimación
 - 15.3.6. Escalas para interpretar las estimaciones
 - 15.3.7. Planning poker
 - 15.3.8. Tipos de planificaciones comunes: planificación de entregas y planificación de iteración
- 15.4. Estilos de diseño de Software distribuido y arquitecturas Software orientadas a servicios
 - 15.4.1. Modelos de comunicación en sistemas Software distribuidos
 - 15.4.2. Capa intermedia o middleware
 - 15.4.3. Patrones de arquitectura para sistemas distribuidos
 - 15.4.4. Proceso general de diseño de servicios Software
 - 15.4.5. Aspectos de diseño de servicios Software
 - 15.4.6. Composición de servicios
 - 15.4.7. Arquitectura de servicios web
 - 15.4.8. Componentes de Infraestructura y SOA
- 15.5. Introducción al desarrollo Software dirigido por modelos
 - 15.5.1. El concepto de modelo
 - 15.5.2. Desarrollo Software dirigido por modelos
 - 15.5.3. Marco de referencia de desarrollo dirigido por modelos MDA
 - 15.5.4. Elementos de un modelo de transformación
- 15.6. Diseño de interfaces gráficas de usuario
 - 15.6.1. Principios de diseño de interfaces de usuario
 - 15.6.2. Patrones de diseño arquitectónico para sistemas interactivos: Modelo Vista Controlador (MVC)
 - 15.6.3. Experiencia de usuario (UX User Experience)

- 15.6.4. Diseño centrado en el usuario
- 15.6.5. Proceso de análisis y diseño de la interfaz gráfica de usuario
- 15.6.6. Usabilidad de interfaces de usuario
- 15.6.7. Accesibilidad en interfaces de usuario
- 15.7. Diseño de aplicaciones web
 - 15.7.1. Características de las aplicaciones web
 - 15.7.2. Interfaz de usuario de una aplicación web
 - 15.7.3. Diseño de navegación
 - 15.7.4. Protocolo de interacción base para aplicaciones web
 - 15.7.5. Estilos de arquitectura para aplicaciones web
- 15.8. Estrategias y técnicas de pruebas Software y factores de calidad del Software
 - 15.8.1. Estrategias de prueba
 - 15.8.2. Diseños de casos de prueba
 - 15.8.3. Relación coste calidad
 - 15.8.4. Modelos de calidad
 - 15.8.5. Familia de normas ISO/IEC 25000 (SQuaRE)
 - 15.8.6. Modelo de calidad de producto (ISO 2501n)
 - 15.8.7. Modelos de calidad de datos (ISO 2501n)
 - 15.8.8. Gestión de la calidad del Software
- 15.9. Introducción a las métricas en Ingeniería Software
 - 15.9.1. Conceptos básicos: Medidas, métricas e indicadores
 - 15.9.2. Tipos de métricas en Ingeniería Software
 - 15.9.3. El proceso de medición
 - 15.9.4. ISO 25024. Métricas externas y de calidad en uso
 - 15.9.5. Métrica orientada a objetos
- 15.10. Mantenimiento y reIngeniería Software
 - 15.10.1. Proceso de mantenimiento
 - 15.10.2. Marco estándar de proceso de mantenimiento. ISO/EIEC 14764
 - 15.10.3. Modelo de proceso de reIngeniería de Software
 - 15.10.4. Ingeniería inversa

Módulo 16. Integración de sistemas

- 16.1. Introducción a los sistemas de información en la empresa
 - 16.1.1. El papel de los sistemas de información
 - 16.1.2. ¿Qué es un sistema de información?
 - 16.1.3. Dimensiones de los sistemas de información
 - 16.1.4. Procesos de negocio y sistemas de información
 - 16.1.5. El departamento de SI/TI
- 16.2. Oportunidades y necesidades de los sistemas de información en la empresa
 - 16.2.1. Organizaciones y sistemas de información
 - 16.2.2. Características de las organizaciones
 - 16.2.3. Impacto de los sistemas de información en la empresa
 - 16.2.4. Sistemas de información para lograr una ventaja competitiva
 - 16.2.5. Uso de los sistemas en la administración y gestión de la empresa
- 16.3. Conceptos básicos de sistemas y tecnologías de la información
 - 16.3.1. Datos, información y conocimiento
 - 16.3.2. Tecnología y sistemas de información
 - 16.3.3. Componentes de la tecnología
 - 16.3.4. Clasificación y tipos de sistemas de información
 - 16.3.5. Arquitecturas basadas en servicios y procesos de negocio
 - 16.3.6. Formas de integración de sistemas
- 16.4. Sistemas para la gestión integrada de recursos de la empresa
 - 16.4.1. Necesidades de la empresa
 - 16.4.2. Un sistema de información integrado para la empresa
 - 16.4.3. Adquisición vs. Desarrollo
 - 16.4.4. Implantación de un ERP
 - 16.4.5. Implicaciones para la dirección
 - 16.4.6. Principales proveedores de ERP
- 16.5. Sistemas de información para la gestión de la cadena de suministro y las relaciones con clientes
 - 16.5.1. Definición de cadena de suministro
 - 16.5.2. Gestión efectiva de la cadena de suministro
 - 16.5.3. El papel de los sistemas de información

- 16.5.4. Soluciones para la gestión de cadena de suministro
- 16.5.5. La gestión de relaciones con los clientes
- 16.5.6. El papel de los sistemas de información
- 16.5.7. Implantación de un sistema CRM
- 16.5.8. Factores críticos de éxito en la implantación de CRM
- 16.5.9. CRM, e-CRM y otras tendencias
- 16.6. La toma de decisiones de inversión en TIC y planificación de sistemas de información
 - 16.6.1. Criterios para la decisión de inversión en TIC
 - 16.6.2. Vinculación del proyecto con la gerencia y plan de negocios
 - 16.6.3. Implicaciones de la dirección
 - 16.6.4. Rediseño de los procesos de negocio
 - 16.6.5. Decisión de metodologías de implantación desde la dirección
 - 16.6.6. Necesidad de planificación de los sistemas de información
 - 16.6.7. Objetivos, participantes y momentos
 - 16.6.8. Estructura y desarrollo del plan de sistemas
 - 16.6.9. Seguimiento y actualización
- 16.7. Consideraciones de seguridad en el uso de las TIC
 - 16.7.1. Análisis de riesgos
 - 16.7.2. La seguridad en los sistemas de información
 - 16.7.3. Consejos prácticos
- 16.8. Viabilidad de aplicación de proyectos de TIC y aspectos financieros en proyectos de sistemas de información
 - 16.8.1. Descripción y objetivos
 - 16.8.2. Participantes en el EVS
 - 16.8.3. Técnicas y prácticas
 - 16.8.4. Estructura de costes
 - 16.8.5. La proyección financiera
 - 16.8.6. Presupuestos
- 16.9. Business Intelligence
 - 16.9.1. ¿Qué es la inteligencia de negocio?
 - 16.9.2. Estrategia e implantación de BI
 - 16.9.3. Presente y futuro en BI

- 16.10. ISO/IEC 12207
 - 16.10.1. ¿Qué es «ISO/IEC 12207»?
 - 16.10.2. Análisis de los sistemas de información
 - 16.10.3. Diseño del sistema de información
 - 16.10.4. Implantación y aceptación del sistema de información

Módulo 17. Reutilización de Software

- 17.1. Panorama general de la reutilización de Software
 - 17.1.1. En qué consiste la reutilización del Software
 - 17.1.2. Ventajas e inconvenientes de la reutilización de Software
 - 17.1.3. Principales técnicas de reutilización de Software
- 17.2. Introducción a los patrones de diseño
 - 17.2.1. ¿Qué es un patrón de diseño?
 - 17.2.2. Catálogo de los principales patrones de diseño
 - 17.2.3. Cómo usar patrones para resolver problemas de diseño
 - 17.2.4. Cómo seleccionar el mejor patrón de diseño
- 17.3. Patrones de creación
 - 17.3.1. Patrones de creación
 - 17.3.2. Patrón Abstract Factory
 - 17.3.3. Ejemplo de implementación del Patrón Abstract Factory
 - 17.3.4. Patrón Builder
 - 17.3.5. Ejemplo de implementación del Builder
 - 17.3.6. Patrón Abstract Factory vs. Builder
- 17.4. Patrones de creación (II)
 - 17.4.1. Patrón Factory Method
 - 17.4.2. Factory Method vs. Abstract Factory
 - 17.4.3. Patrón Singleton
- 17.5. Patrones estructurales
 - 17.5.1. Patrones estructurales
 - 17.5.2. Patrón Adapter
 - 17.5.3. Patrón Bridge

- 17.6. Patrones estructurales (II)
 - 17.6.1. Patrón Composite
 - 17.6.2. Patrón Decorador
- 17.7. Patrones estructurales (III)
 - 17.7.1. Patrón Facade
 - 17.7.2. Patrón Proxy
- 17.8. Patrones de comportamiento
 - 17.8.1. Concepto de los patrones de comportamiento
 - 17.8.2. Patrón de comportamiento: Cadena de responsabilidad
 - 17.8.3. Patrón de comportamiento Orden
- 17.9. Patrones de comportamiento (II)
 - 17.9.1. Patrón Intérprete o Interpreter
 - 17.9.2. Patrón Iterador
 - 17.9.3. Patrón Observador
 - 17.9.4. Patrón Estrategia
- 17.10. Frameworks
 - 17.10.1. Concepto de framework
 - 17.10.2. Desarrollo mediante frameworks
 - 17.10.3. Patrón Model View Controller
 - 17.10.4. Frameworks para diseño de interfaces gráficas de usuario
 - 17.10.5. Frameworks para el desarrollo de aplicaciones web
 - 17.10.6. Frameworks para la gestión de la persistencia de objetos en bases de datos

Módulo 18. Servicios de tecnología de la información

- 18.1. La transformación digital (I)
 - 18.1.1. La innovación empresarial
 - 18.1.2. La gestión de la producción
 - 18.1.3. La gestión financiera
- 18.2. La transformación digital (II)
 - 18.2.1. El marketing
 - 18.2.2. La gestión de RR. HH
 - 18.2.3. Un sistema de información integrado

- 18.3. Caso de estudio
 - 18.3.1. Presentación de la empresa
 - 18.3.2. Metodologías para analizar la adquisición de TI
 - 18.3.3. Determinación de costos, beneficios y riesgos
 - 18.3.4. Evaluación económica de la inversión
- 18.4. El gobierno y la gestión de las TIC
 - 18.4.1. Definición de gobierno de las tecnologías y sistemas de la información
 - 18.4.2. Diferencia entre gobierno y gestión de las TSI
 - 18.4.3. Marcos para el gobierno y la gestión de las TSI
 - 18.4.4. Las normas y el gobierno y la gestión de las TSI
- 18.5. El gobierno corporativo de las TIC
 - 18.5.1. ¿Qué es el buen gobierno corporativo?
 - 18.5.2. Antecedentes de gobierno de las TIC
 - 18.5.3. La Norma ISO/IEC 38500:2008
 - 18.5.4. Implementación de un buen gobierno TIC
 - 18.5.5. Gobierno TIC y mejores prácticas
 - 18.5.6. Gobierno corporativo. Resumen y tendencias
- 18.6. Objetivos de Control para la Información y Tecnologías Relacionadas (COBIT)
 - 18.6.1. Marco de aplicación
 - 18.6.2. Dominio: Planificación y organización
 - 18.6.3. Dominio: Adquisición e implementación
 - 18.6.4. Dominio: Entrega y soporte
 - 18.6.5. Dominio: Supervisión y evaluación
 - 18.6.6. Aplicación de la guía COBIT
- 18.7. La Biblioteca de Infraestructura de Tecnologías de Información (ITIL)
 - 18.7.1. Introducción a ITIL
 - 18.7.2. Estrategia del servicio
 - 18.7.3. Diseño del servicio
 - 18.7.4. Transición del servicio
 - 18.7.5. Operación del servicio
 - 18.7.6. Mejora del servicio

- 18.8. El sistema de gestión de servicios
 - 18.8.1. Principios básicos de UNE-ISO/IEC 20000-1
 - 18.8.2. La estructura de la serie de normas ISO/IEC 20000
 - 18.8.3. Requisitos del Sistema de Gestión del Servicio (SGS)
 - 18.8.4. Diseño y transición de servicios nuevos o modificados
 - 18.8.5. Procesos de provisión del servicio
 - 18.8.6. Grupos de procesos
- 18.9. El sistema de gestión de activos de Software
 - 18.9.1. Justificación de la necesidad
 - 18.9.2. Antecedentes
 - 18.9.3. Presentación de la norma 19770
 - 18.9.4. Implantación de la gestión
- 18.10. Gestión de la continuidad del negocio
 - 18.10.1. Plan de la continuidad del negocio
 - 18.10.2. Implementación de un BCM

“

Aprovecha tus conocimientos previos y aplica por la convalidación de asignaturas ya superadas, optimizando tu experiencia en este programa”

04

Objetivos docentes

Este programa en Ingeniería de Software ha sido diseñado para proporcionar a los profesionales del ámbito informático las herramientas avanzadas necesarias para impulsar sus carreras hacia el desarrollo de Software moderno, adaptado a las demandas cambiantes del mercado. A través de conocimientos técnicos especializados, los estudiantes incrementarán significativamente sus oportunidades de crecimiento profesional, abriendo puertas a roles clave en las principales empresas del sector.

eli

mir
mod
bpy
pri

```
mirror_mod.use_x = False  
mirror_mod.use_y = True  
mirror_mod.use_z = False  
if _operation == "MIRROR_Z":  
    mirror_mod.use_x = False  
    mirror_mod.use_y = False  
    mirror_mod.use_z = True
```

```
#selection at the end -add back the deselection  
mirror_ob.select= 1  
modifier_ob.select=1  
bpy.context.scene.objects.active = mirror_ob  
print("Selected" + str(modifier_ob))  
#mirror_ob.select = 0  
bpy.context.selected_objects = [mirror_ob]
```

“

*Un programa pensado para convertirse
en el impulso definitivo hacia una carrera
exitosa en la Ingeniería de Software”*



Objetivos generales

- ♦ Adquirir nuevas competencias necesarias y demandadas en cuanto a nuevas tecnologías y últimas novedades en Software
- ♦ Complementar los conocimientos adquiridos con habilidades en el campo de la computación y la estructura de computadoras, todo ello incluyendo la base matemática, estadística y física imprescindible en una Ingeniería
- ♦ Ampliar los conocimientos en Ingeniería de Software y Sistemas Informáticos con las últimas novedades y metodología más innovadora
- ♦ Abordar proyectos y entornos de Software complejos, sabiendo aportar soluciones inteligentes a problemáticas diversas



Impulsa tu carrera con un programa de especialización diseñado para liderar en la dirección estratégica de la Ingeniería de Software”





Objetivos específicos

Módulo 1. Metodologías, desarrollo y calidad en la Ingeniería de Software

- ♦ Capacitar en el uso de metodologías ágiles y tradicionales en el desarrollo de Software
- ♦ Asegurar la calidad del producto final mediante la aplicación de estándares y buenas prácticas en cada fase del ciclo de vida del Software

Módulo 2. Gestión de proyectos de Software

- ♦ Desarrollar las competencias necesarias para planificar, ejecutar y gestionar proyectos de Software
- ♦ Aplicar metodologías de gestión de proyectos y herramientas de planificación para asegurar la entrega oportuna y dentro del presupuesto

Módulo 3. Plataformas de desarrollo del Software

- ♦ Proporcionar conocimientos sobre las diferentes plataformas de desarrollo de Software
- ♦ Abordar el uso de entornos de desarrollo, *frameworks* y herramientas que optimicen la creación de aplicaciones robustas y escalables

Módulo 4. Computación en el cliente web

- ♦ Capacitar en el desarrollo de aplicaciones web del lado del cliente, utilizando tecnologías como HTML, CSS y JavaScript
- ♦ Crear interfaces interactivas y optimizadas para la experiencia del usuario

Módulo 5. Computación en servidor web

- ♦ Desarrollar las habilidades necesarias para gestionar y desarrollar aplicaciones del lado del servidor web
- ♦ Manejar tecnologías como PHP, Node.js o Python, y gestionando bases de datos y servicios backend

Módulo 6. Gestión de la seguridad

- ♦ Brindar los conocimientos y habilidades para gestionar la seguridad en los entornos de desarrollo de Software
- ♦ Aplicar políticas y estrategias de protección de datos, accesos y seguridad de la infraestructura tecnológica

Módulo 7. Seguridad en los sistemas de información

- ♦ Proporcionar una comprensión profunda de la seguridad en los sistemas de información
- ♦ Analizar desde la protección de redes y servidores hasta la implementación de protocolos y normas de seguridad en las infraestructuras tecnológicas

Módulo 8. Seguridad en el Software

- ♦ Capacitar en el desarrollo de Software seguro
- ♦ Identificar, mitigar y prevenir vulnerabilidades en el código y en los procesos de desarrollo de aplicaciones

Módulo 9. Calidad y auditoría de sistemas de información

- ♦ Desarrollar competencias para llevar a cabo auditorías de calidad en los sistemas de información
- ♦ Garantizar la efectividad de los controles de seguridad, el cumplimiento de las normativas y la optimización de los procesos tecnológicos

Módulo 10. Administración de servidores web

- ♦ Capacitar en la gestión y administración de servidores web
- ♦ Profundizar en el mantenimiento de servidores que alojan aplicaciones y servicios en línea

Módulo 11. Seguridad en aplicaciones online

- ♦ Brindar los conocimientos necesarios para asegurar aplicaciones online
- ♦ Abordar aspectos como la autenticación, protección de datos, gestión de accesos y prevención de ataques en entornos web

Módulo 12. Ingeniería del Software

- ♦ Proporcionar una comprensión integral de los principios y prácticas de la ingeniería del Software, capacitando
- ♦ Diseñar, desarrollar y mantener aplicaciones de Software eficientes, escalables y de alta calidad

Módulo 13. Ingeniería del Software avanzada

- ♦ Profundizar en las técnicas avanzadas de Ingeniería del Software
- ♦ Ahondar en las metodologías de desarrollo sofisticadas, patrones de diseño, arquitectura de Software y optimización de procesos para proyectos complejos

Módulo 14. Ingeniería de requisitos

- ♦ Desarrollar habilidades en la identificación, análisis y documentación de los requisitos del Software
- ♦ Garantizar que los productos finales satisfagan las necesidades y expectativas de los usuarios



Módulo 15. Procesos de Ingeniería del Software

- ♦ Capacitar en los procesos fundamentales de la ingeniería del Software, desde la planificación y el diseño hasta la implementación, pruebas y mantenimiento
- ♦ Mejorar la calidad y la eficiencia del ciclo de vida del Software

Módulo 16. Integración de sistemas

- ♦ Integrar diferentes sistemas de Software, bases de datos y aplicaciones
- ♦ Crear soluciones tecnológicas cohesionadas y eficientes para resolver problemas empresariales complejos

Módulo 17. Reutilización de Software

- ♦ Aplicar técnicas y principios para la reutilización de componentes y módulos de Software en nuevos proyectos
- ♦ Optimizar el tiempo de desarrollo y los costos mediante la creación de bibliotecas o *frameworks* reutilizables

Módulo 18. Servicios de tecnología de la información

- ♦ Diseñar, implementar y gestionar servicios de tecnología de la información
- ♦ Mejorar la infraestructura tecnológica para garantizar la continuidad del negocio y la eficiencia operativa

05

Salidas profesionales

Tras finalizar este programa Universitario, los profesionales contarán con una sólida comprensión de las técnicas, metodologías y herramientas necesarias para abordar proyectos de desarrollo de Software modernos y adaptados a las necesidades actuales del mercado. Los egresados estarán capacitados para diseñar, implementar y gestionar soluciones tecnológicas avanzadas, garantizando la calidad, seguridad y eficiencia en todas las fases del desarrollo. Además, podrán liderar equipos multidisciplinarios, colaborar en entornos ágiles y tomar decisiones estratégicas basadas en estándares internacionales.



```
45 <input type="radio" class="radio" name="sex" id="sex" value="M" /> ชาย  
46 <input type="radio" class="radio" name="sex" id="sex" value="F" /> หญิง </div>  
47  
48 คำตอบใช่: <br/>  
49 <input type="checkbox" class="checkbox" name="intr1" id="intr1" /> ใช่  
50 <input type="checkbox" class="checkbox" name="intr2" id="intr2" /> ไม่ใช่ <br/>  
51  
52  
53 ชื่อที่อยู่: <br/>  
54 <textarea class="text" name="address" id="address" rows="4" cols="50"></textarea><br/>  
55
```

“

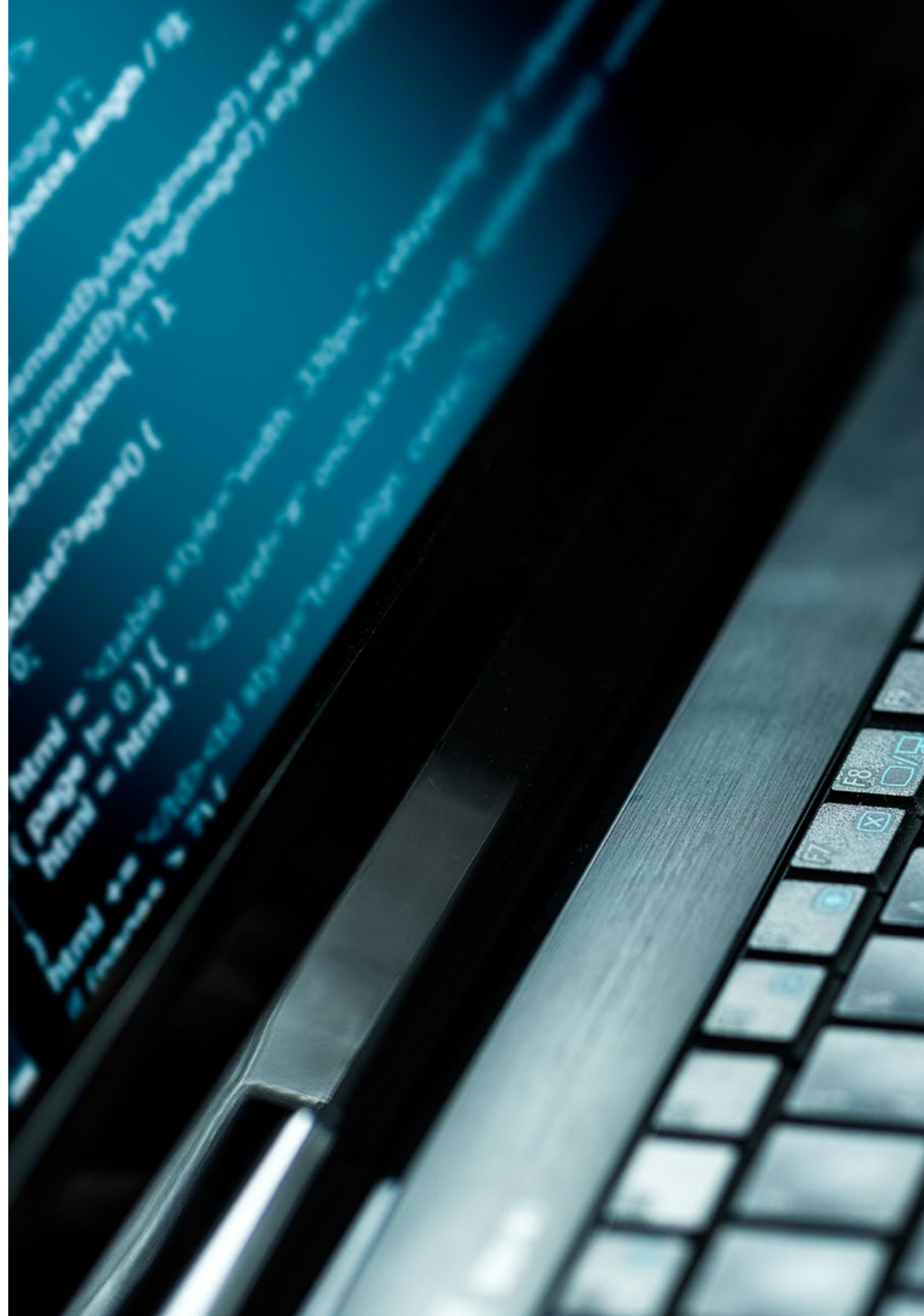
Aplicarás enfoques innovadores y soluciones efectivas que optimicen el desarrollo y la gestión de Software en proyectos tecnológicos complejos”

Perfil del egresado

El egresado del programa en Ingeniería de Software es un profesional altamente capacitado para enfrentar los desafíos tecnológicos del desarrollo de Software moderno. Adicionalmente, posee un conocimiento profundo de metodologías avanzadas, herramientas de desarrollo y estándares internacionales, necesarios para crear soluciones innovadoras y eficaces. Además, está preparado para diseñar, implementar y gestionar proyectos tecnológicos de alto impacto, colaborar con equipos multidisciplinarios y liderar iniciativas estratégicas en entornos dinámicos y competitivos, garantizando la calidad, seguridad y escalabilidad de las aplicaciones desarrolladas.

Combinarás conocimiento teórico con habilidades prácticas en diseño de Software y metodologías de desarrollo avanzadas.

- ♦ **Diseño y desarrollo de Software:** Capacidad para diseñar e implementar aplicaciones avanzadas, integrando patrones de diseño, modelado UML y plataformas de desarrollo como Java y .NET
- ♦ **Gestión de proyectos tecnológicos:** Habilidad para planificar, supervisar y dirigir proyectos tecnológicos utilizando metodologías ágiles como Scrum y aplicando estándares internacionales ISO/IEC
- ♦ **Seguridad en el desarrollo:** Comprensión profunda de las mejores prácticas en seguridad informática, gestión de riesgos y protección de datos en todas las etapas del ciclo de vida del Software
- ♦ **Implementación en entornos complejos:** Capacidad para desarrollar aplicaciones móviles, web y basadas en la nube, utilizando tecnologías como REST, SOA y MVC
- ♦ **Resolución de problemas:** Aplicación de pensamiento crítico y estrategias innovadoras para identificar y solucionar desafíos tecnológicos en proyectos de Software
- ♦ **Competencia digital:** Dominio de herramientas tecnológicas avanzadas y plataformas digitales para el desarrollo, pruebas y gestión de sistemas





Después de realizar el Grand Master, podrás desempeñar tus conocimientos y habilidades en los siguientes cargos:

1. **Director de proyectos de Software:** Responsable de la planificación y ejecución de proyectos complejos, asegurando el cumplimiento de objetivos de calidad, tiempo y presupuesto.
2. **Arquitecto de Software:** Especialista en el diseño y construcción de sistemas complejos y escalables.
3. **Ingeniero de seguridad informática:** Encargado de proteger sistemas, aplicaciones y datos frente a vulnerabilidades y ataques.
4. **Desarrollador full-stack:** Profesional capaz de crear soluciones integrales tanto en el lado del cliente como del servidor.
5. **Consultor tecnológico:** Asesor experto en la implementación de tecnologías avanzadas y estrategias de transformación digital.
6. **Especialista en aplicaciones en la nube:** Diseñador e implementador de soluciones basadas en SaaS, PaaS e IaaS.
7. **Administrador de sistemas:** Gestor de servidores, redes y plataformas tecnológicas, optimizando su rendimiento y seguridad.
8. **Analista de calidad de Software:** Evaluador de la funcionalidad, usabilidad y rendimiento de aplicaciones tecnológicas.
9. **Asesor en Ingeniería de requisitos:** Especialista en el análisis, documentación y validación de requisitos en proyectos de Software.



Completa este programa y accede a posiciones clave dentro de la industria tecnológica, destacándote como un líder en Ingeniería de Software”

06

Metodología de estudio

TECH es la primera universidad en el mundo que combina la metodología de los **case studies** con el **Relearning**, un sistema de aprendizaje 100% online basado en la reiteración dirigida.

Esta disruptiva estrategia pedagógica ha sido concebida para ofrecer a los profesionales la oportunidad de actualizar conocimientos y desarrollar competencias de un modo intensivo y riguroso. Un modelo de aprendizaje que coloca al estudiante en el centro del proceso académico y le otorga todo el protagonismo, adaptándose a sus necesidades y dejando de lado las metodologías más convencionales.



“

TECH te prepara para afrontar nuevos retos en entornos inciertos y lograr el éxito en tu carrera”

El alumno: la prioridad de todos los programas de TECH

En la metodología de estudios de TECH el alumno es el protagonista absoluto. Las herramientas pedagógicas de cada programa han sido seleccionadas teniendo en cuenta las demandas de tiempo, disponibilidad y rigor académico que, a día de hoy, no solo exigen los estudiantes sino los puestos más competitivos del mercado.

Con el modelo educativo asincrónico de TECH, es el alumno quien elige el tiempo que destina al estudio, cómo decide establecer sus rutinas y todo ello desde la comodidad del dispositivo electrónico de su preferencia. El alumno no tendrá que asistir a clases en vivo, a las que muchas veces no podrá acudir. Las actividades de aprendizaje las realizará cuando le venga bien. Siempre podrá decidir cuándo y desde dónde estudiar.

“

*En TECH NO tendrás clases en directo
(a las que luego nunca puedes asistir)”*



Los planes de estudios más exhaustivos a nivel internacional

TECH se caracteriza por ofrecer los itinerarios académicos más completos del entorno universitario. Esta exhaustividad se logra a través de la creación de temarios que no solo abarcan los conocimientos esenciales, sino también las innovaciones más recientes en cada área.

Al estar en constante actualización, estos programas permiten que los estudiantes se mantengan al día con los cambios del mercado y adquieran las habilidades más valoradas por los empleadores. De esta manera, quienes finalizan sus estudios en TECH reciben una preparación integral que les proporciona una ventaja competitiva notable para avanzar en sus carreras.

Y además, podrán hacerlo desde cualquier dispositivo, pc, tableta o smartphone.

“

El modelo de TECH es asincrónico, de modo que te permite estudiar con tu pc, tableta o tu smartphone donde quieras, cuando quieras y durante el tiempo que quieras”

Case studies o Método del caso

El método del caso ha sido el sistema de aprendizaje más utilizado por las mejores escuelas de negocios del mundo. Desarrollado en 1912 para que los estudiantes de Derecho no solo aprendiesen las leyes a base de contenidos teóricos, su función era también presentarles situaciones complejas reales. Así, podían tomar decisiones y emitir juicios de valor fundamentados sobre cómo resolverlas. En 1924 se estableció como método estándar de enseñanza en Harvard.

Con este modelo de enseñanza es el propio alumno quien va construyendo su competencia profesional a través de estrategias como el *Learning by doing* o el *Design Thinking*, utilizadas por otras instituciones de renombre como Yale o Stanford.

Este método, orientado a la acción, será aplicado a lo largo de todo el itinerario académico que el alumno emprenda junto a TECH. De ese modo se enfrentará a múltiples situaciones reales y deberá integrar conocimientos, investigar, argumentar y defender sus ideas y decisiones. Todo ello con la premisa de responder al cuestionamiento de cómo actuaría al posicionarse frente a eventos específicos de complejidad en su labor cotidiana.



Método Relearning

En TECH los *case studies* son potenciados con el mejor método de enseñanza 100% online: el *Relearning*.

Este método rompe con las técnicas tradicionales de enseñanza para poner al alumno en el centro de la ecuación, proveyéndole del mejor contenido en diferentes formatos. De esta forma, consigue repasar y reiterar los conceptos clave de cada materia y aprender a aplicarlos en un entorno real.

En esta misma línea, y de acuerdo a múltiples investigaciones científicas, la reiteración es la mejor manera de aprender. Por eso, TECH ofrece entre 8 y 16 repeticiones de cada concepto clave dentro de una misma lección, presentada de una manera diferente, con el objetivo de asegurar que el conocimiento sea completamente afianzado durante el proceso de estudio.

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



Un Campus Virtual 100% online con los mejores recursos didácticos

Para aplicar su metodología de forma eficaz, TECH se centra en proveer a los egresados de materiales didácticos en diferentes formatos: textos, vídeos interactivos, ilustraciones y mapas de conocimiento, entre otros. Todos ellos, diseñados por profesores cualificados que centran el trabajo en combinar casos reales con la resolución de situaciones complejas mediante simulación, el estudio de contextos aplicados a cada carrera profesional y el aprendizaje basado en la reiteración, a través de audios, presentaciones, animaciones, imágenes, etc.

Y es que las últimas evidencias científicas en el ámbito de las Neurociencias apuntan a la importancia de tener en cuenta el lugar y el contexto donde se accede a los contenidos antes de iniciar un nuevo aprendizaje. Poder ajustar esas variables de una manera personalizada favorece que las personas puedan recordar y almacenar en el hipocampo los conocimientos para retenerlos a largo plazo. Se trata de un modelo denominado *Neurocognitive context-dependent e-learning* que es aplicado de manera consciente en esta titulación universitaria.

Por otro lado, también en aras de favorecer al máximo el contacto mentor-alumno, se proporciona un amplio abanico de posibilidades de comunicación, tanto en tiempo real como en diferido (mensajería interna, foros de discusión, servicio de atención telefónica, email de contacto con secretaría técnica, chat y videoconferencia).

Asimismo, este completísimo Campus Virtual permitirá que el alumnado de TECH organice sus horarios de estudio de acuerdo con su disponibilidad personal o sus obligaciones laborales. De esa manera tendrá un control global de los contenidos académicos y sus herramientas didácticas, puestas en función de su acelerada actualización profesional.



La modalidad de estudios online de este programa te permitirá organizar tu tiempo y tu ritmo de aprendizaje, adaptándolo a tus horarios”

La eficacia del método se justifica con cuatro logros fundamentales:

1. Los alumnos que siguen este método no solo consiguen la asimilación de conceptos, sino un desarrollo de su capacidad mental, mediante ejercicios de evaluación de situaciones reales y aplicación de conocimientos.
2. El aprendizaje se concreta de una manera sólida en capacidades prácticas que permiten al alumno una mejor integración en el mundo real.
3. Se consigue una asimilación más sencilla y eficiente de las ideas y conceptos, gracias al planteamiento de situaciones que han surgido de la realidad.
4. La sensación de eficiencia del esfuerzo invertido se convierte en un estímulo muy importante para el alumnado, que se traduce en un interés mayor en los aprendizajes y un incremento del tiempo dedicado a trabajar en el curso.

La metodología universitaria mejor valorada por sus alumnos

Los resultados de este innovador modelo académico son constatables en los niveles de satisfacción global de los egresados de TECH.

La valoración de los estudiantes sobre la calidad docente, calidad de los materiales, estructura del curso y sus objetivos es excelente. No en valde, la institución se convirtió en la universidad mejor valorada por sus alumnos según el índice global score, obteniendo un 4,9 de 5.

Accede a los contenidos de estudio desde cualquier dispositivo con conexión a Internet (ordenador, tablet, smartphone) gracias a que TECH está al día de la vanguardia tecnológica y pedagógica.

Podrás aprender con las ventajas del acceso a entornos simulados de aprendizaje y el planteamiento de aprendizaje por observación, esto es, Learning from an expert.



Así, en este programa estarán disponibles los mejores materiales educativos, preparados a conciencia:



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 que creará nuestra manera de trabajo online, con las técnicas más novedosas que nos permiten ofrecerte una gran calidad, en cada una de las piezas que pondremos a tu servicio.



Prácticas de habilidades y competencias

Realizarás 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.



Resúmenes interactivos

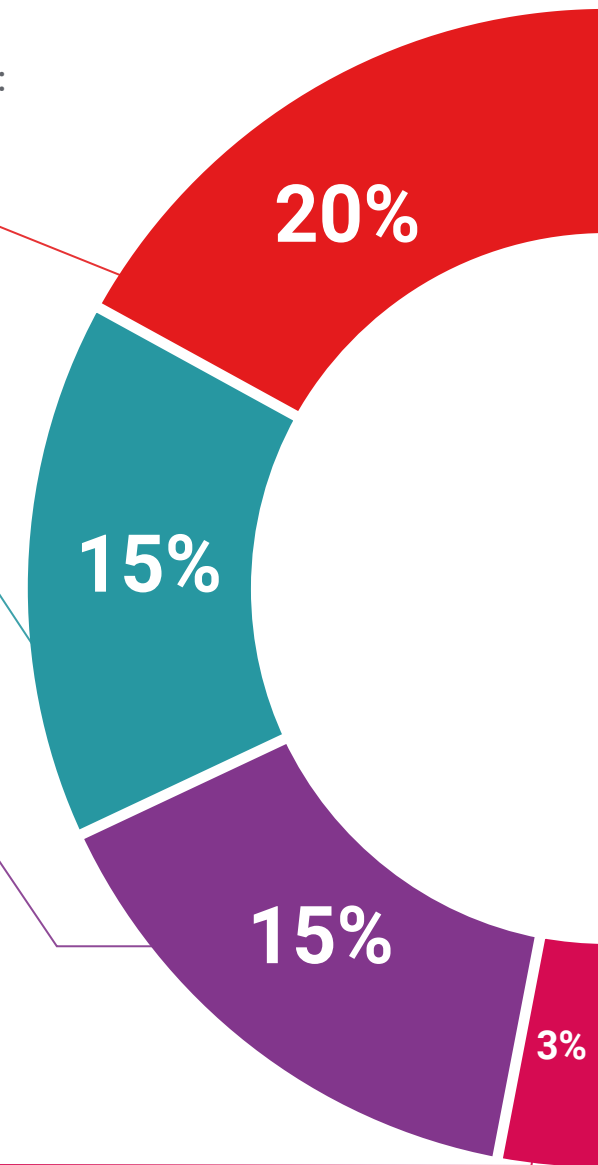
Presentamos los contenidos de manera atractiva y dinámica en píldoras multimedia que incluyen audio, vídeos, imágenes, esquemas y mapas conceptuales con el fin de afianzar el conocimiento.

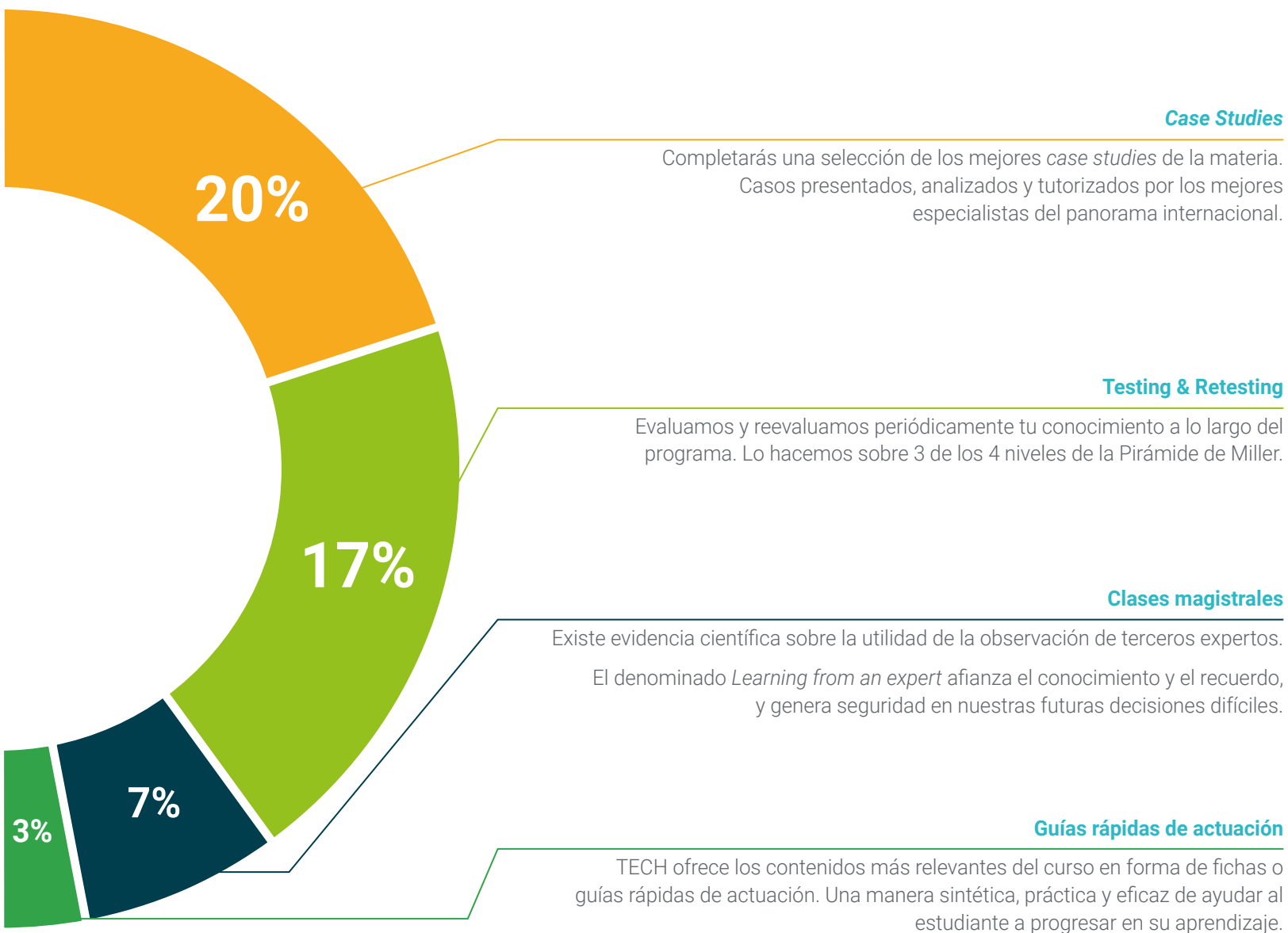
Este sistema exclusivo educativo para la presentación de contenidos multimedia fue premiado por Microsoft como "Caso de éxito en Europa".



Lecturas complementarias

Artículos recientes, documentos de consenso, guías internacionales... En nuestra biblioteca virtual tendrás acceso a todo lo que necesitas para completar tu capacitación.





07

Cuadro docente

Los docentes de este programa en Ingeniería de Software destacan por su profundo conocimiento técnico y su experiencia práctica en la industria del Software. De hecho, han liderado proyectos significativos en empresas líderes del sector, lo que les permite compartir, no solo teorías avanzadas, sino también aplicaciones prácticas y casos reales que enriquecerán la preparación de los egresados. Además, su compromiso con la innovación y la investigación les impulsa a mantenerse actualizados en las últimas tendencias y metodologías de desarrollo.

“

*Aprende de docentes expertos en
Informática e Ingeniería de Software, con
experiencia y conocimientos avanzados”*

Directora Invitada Internacional

Darren Pulsipher es un **arquitecto de software** altamente experimentado, un innovador con una destacada trayectoria internacional en el **desarrollo de software y firmware**. De hecho, posee habilidades altamente desarrolladas en **comunicación, gestión de proyectos y negocios**, lo que le ha permitido liderar importantes iniciativas a nivel global.

Asimismo, ha ocupado altos cargos de gran responsabilidad a lo largo de su carrera, como el de **Arquitecto Jefe de Soluciones para el Sector Público** en Intel Corporation, donde ha promovido **negocios modernos, procesos y tecnologías** para clientes, socios y usuarios del **sector público**. Además, ha fundado Yoly Inc., donde también se ha desempeñado como **CEO**, trabajando para desarrollar una **herramienta de agregación y diagnóstico de redes sociales** basada en el **Software Como Servicio (SaaS)**, utilizando para ello tecnologías de **Big Data** y **Web 2.0**.

Adicionalmente, ha ejercido en otras empresas, como **Director Sénior de Ingeniería**, en Dell Technologies, donde ha dirigido la **Unidad de Negocios de Big Data en la Nube**, liderando los equipos en **Estados Unidos y China** para la gestión de proyectos de gran envergadura y la reestructuración de divisiones empresariales para su integración exitosa. Igualmente, ha trabajado como **Director de Tecnologías de la Información (Chief Information Officer)** en XanGo, donde ha gestionado proyectos tales como el **soporte de Help Desk**, el **soporte de producción** y el **desarrollo de soluciones**.

Entre las múltiples especialidades en las que es experto, sobresalen la tecnología **Edge to Cloud**, la **ciberseguridad**, la **Inteligencia Artificial Generativa**, el **desarrollo de software**, la **tecnología de redes**, el **desarrollo nativo en la nube** y el **ecosistema de contenedores**. Conocimientos que ha compartido a través del **pódcast y boletín semanal "Embracing Digital Transformation"**, que él mismo ha producido y presentado, ayudando a las organizaciones a navegar con éxito en la **transformación digital** mediante el aprovechamiento de las **personas, los procesos y la tecnología**.



D. Pulsipher, Darren

- Arquitecto Jefe de Soluciones para el Sector Público en Intel, California, Estados Unidos
- Presentador y Productor de *"Embracing Digital Transformation"*, California
- Fundador y CEO en Yoly Inc., Arkansas
- Director Sénior de Ingeniería en Dell Technologies, Arkansas
- Director de Tecnologías de la Información (*Chief Information Officer*) en XanGo, Utah
- Arquitecto Sénior en Cadence Design Systems, California
- Gerente Sénior de Procesos de Proyectos en Lucent Technologies, California
- Ingeniero de Software en Cemax-Icon, California
- Ingeniero de Software en ISG Technologies, Canadá
- MBA en Gestión de Tecnología por la Universidad de Phoenix
- Licenciado en Ciencias de la Computación e Ingeniería Eléctrica por la Universidad Brigham Young



Gracias a TECH podrás aprender con los mejores profesionales del mundo"

08 Titulación

El Grand Master en Ingeniería de Software garantiza, además de la capacitación más rigurosa y actualizada, el acceso a un título de Grand Master expedido por TECH Universidad.



“

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

Este **Grand Master en Ingeniería de Software** contiene el programa más completo y actualizado del mercado.

Tras la superación de la evaluación, el alumno recibirá por correo postal* con acuse de recibo su correspondiente título de **Grand Master** emitido por **TECH Universidad**.

Este título expedido por **TECH Universidad** expresará la calificación que haya obtenido en el Grand Master, y reunirá los requisitos comúnmente exigidos por las bolsas de trabajo, oposiciones y comités evaluadores de carreras profesionales.

Título: **Grand Master en Ingeniería de Software**

Modalidad: **No escolarizada (100% en línea)**

Duración: **2 años**



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



Grand Master Ingeniería de Software

- » Modalidad: No escolarizada (100% en línea)
- » Duración: 2 años
- » Titulación: TECH Universidad
- » Horario: a tu ritmo
- » Exámenes: online

Grand Master

Ingeniería de Software