Grand Master

Ingeniería y Calidad del Software

tech global university

tech global university

Grand Master
Ingeniería y Calidad del Software

- Modalidad: **online**
- Duración: **2 años**
- Titulación: **TECH Global University**
- Acreditación: **120 ECTS**
- Horario: **a tu ritmo**
- Exámenes: **online**

Acceso web: www.techtitute.com/informatica/grand-master/grand-master-ingenieria-calidad-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. 34

05 Salidas profesionales — pág. 40

06 Metodología de estudio — pág. 44

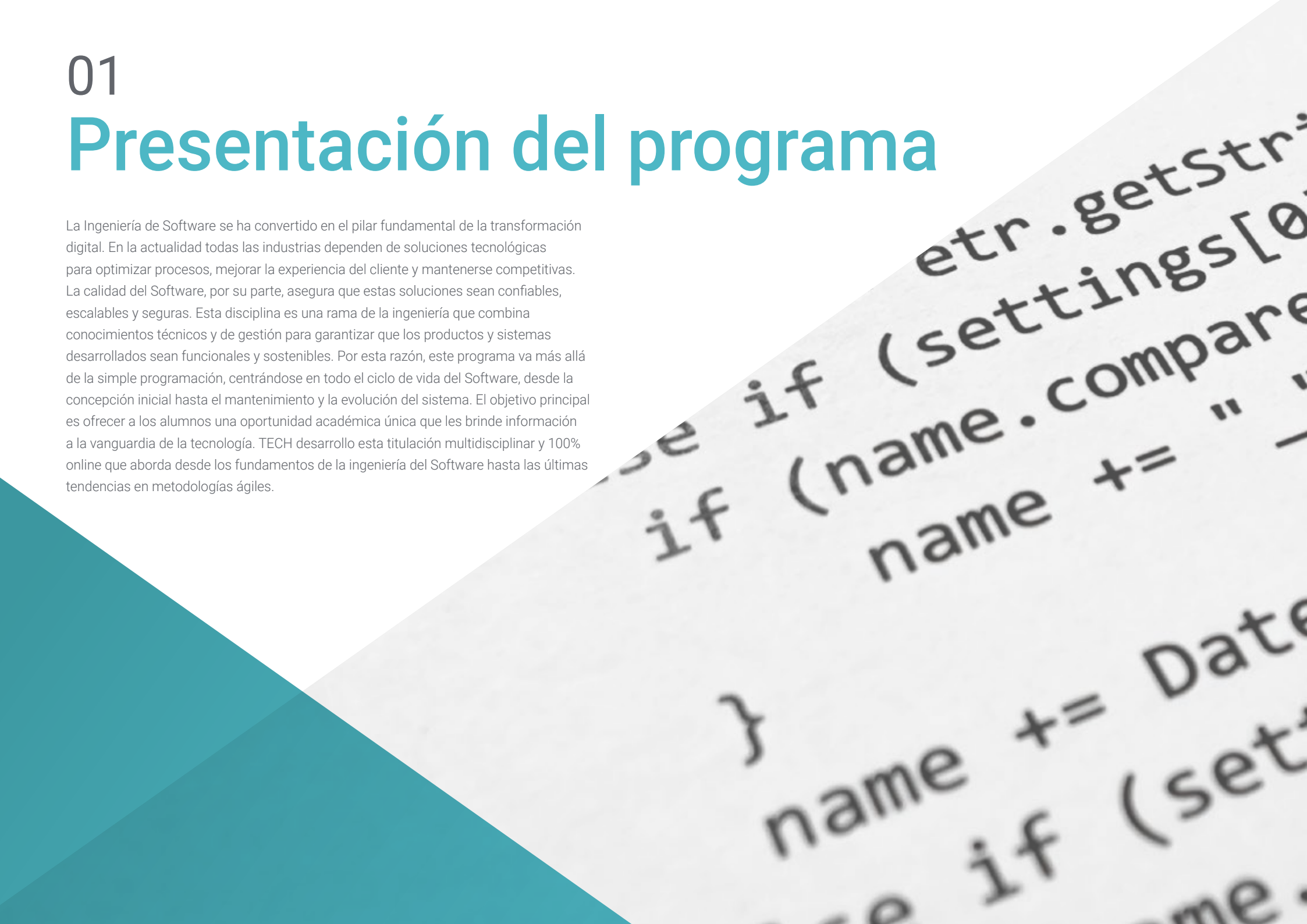
07 Cuadro docente — pág. 54

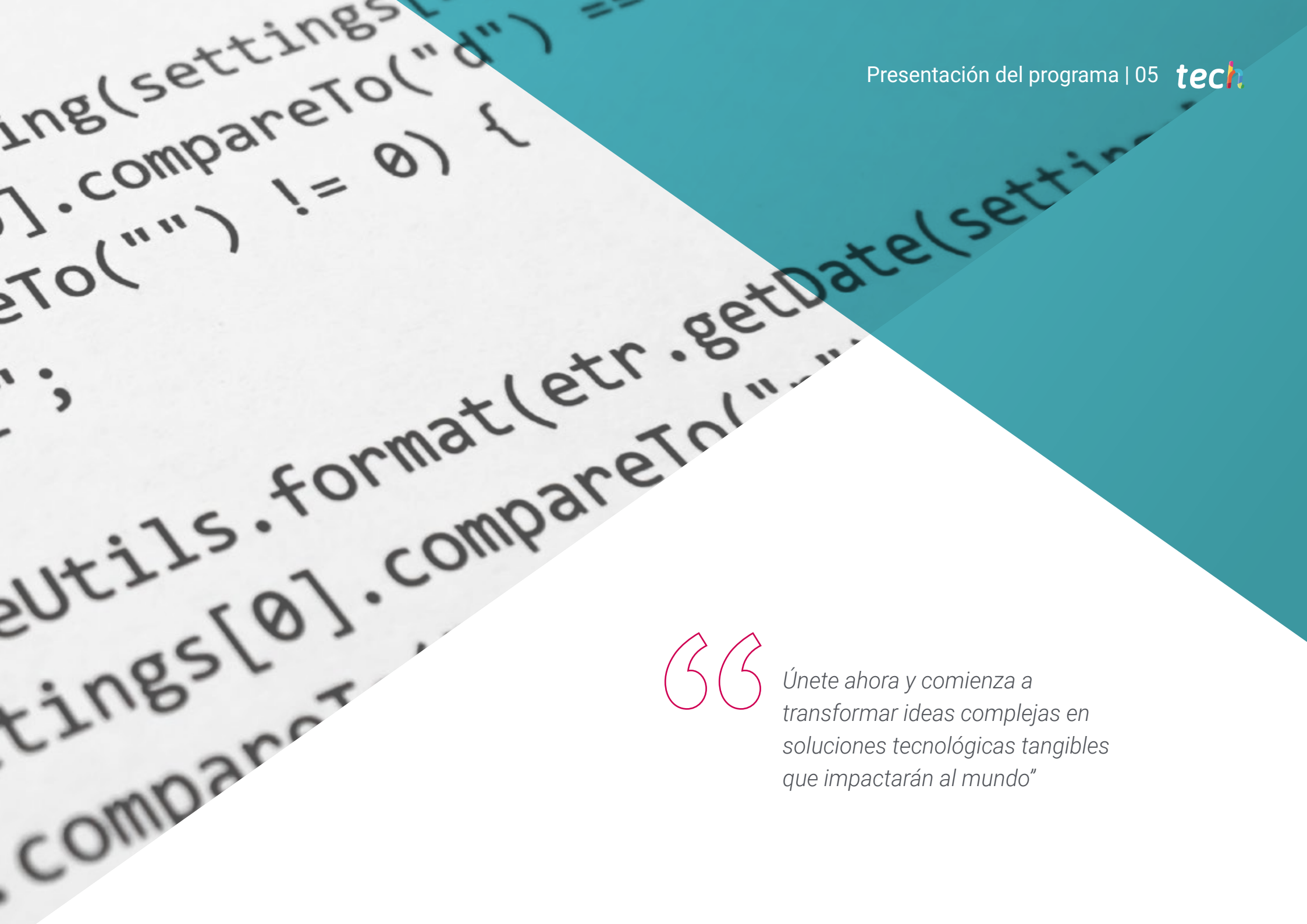
08 Titulación — pág. 60

01

Presentación del programa

La Ingeniería de Software se ha convertido en el pilar fundamental de la transformación digital. En la actualidad todas las industrias dependen de soluciones tecnológicas para optimizar procesos, mejorar la experiencia del cliente y mantenerse competitivas. La calidad del Software, por su parte, asegura que estas soluciones sean confiables, escalables y seguras. Esta disciplina es una rama de la ingeniería que combina conocimientos técnicos y de gestión para garantizar que los productos y sistemas desarrollados sean funcionales y sostenibles. Por esta razón, este programa va más allá de la simple programación, centrándose en todo el ciclo de vida del Software, desde la concepción inicial hasta el mantenimiento y la evolución del sistema. El objetivo principal es ofrecer a los alumnos una oportunidad académica única que les brinde información a la vanguardia de la tecnología. TECH desarrollo esta titulación multidisciplinar y 100% online que aborda desde los fundamentos de la ingeniería del Software hasta las últimas tendencias en metodologías ágiles.





Únete ahora y comienza a transformar ideas complejas en soluciones tecnológicas tangibles que impactarán al mundo”

La calidad del Software garantiza que los sistemas no solo cumplan con los requisitos funcionales, sino que también sean intuitivos, seguros y sostenibles a largo plazo. Este aspecto resulta especialmente relevante en sectores críticos, como el financiero, la salud o el transporte, donde los fallos pueden derivar en consecuencias graves. Además, al priorizar la calidad, se asegura que las empresas puedan adaptarse de manera ágil a los constantes avances tecnológicos y responder eficazmente a las crecientes demandas del mercado.

Mediante el uso de metodologías como el desarrollo ágil, *DevOps* y la implementación de estándares internacionales de calidad, la ingeniería de Software garantiza la entrega de productos en tiempos más cortos. Además, los costos controlados y el nivel de calidad que minimiza errores críticos han sido ampliados con la integración de tecnologías emergentes, como la inteligencia artificial, el *cloud computing* y la ciberseguridad. En este contexto, el programa que ha diseñado TECH está orientado a formar profesionales altamente capacitados en el diseño, desarrollo, gestión y aseguramiento de la calidad del Software. Para adquirir las habilidades necesarias, el temario del Grand Master incluye los conceptos más actualizados sobre gestión de proyectos tecnológicos y dirección estratégica. Este enfoque representa un valor añadido tanto para ingenieros que ya ocupan puestos de responsabilidad y desean actualizar sus conocimientos, como para aquellos que aspiran a liderar equipos y proyectos por primera vez en este ámbito.

Una de las principales ventajas que ofrece este programa es que será 100% online, lo que elimina la necesidad de traslados y de ajustarse a horarios específicos. Además, los estudiantes contarán con el método de aprendizaje *Relearning*, el cual se adapta a su ritmo de estudio. Este enfoque flexible resulta de gran utilidad, ya que permite a los alumnos organizar sus obligaciones diarias, ya sean profesionales o familiares, de manera eficiente, logrando así un desarrollo completo.

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

- El desarrollo de casos prácticos presentados por expertos en informática
- 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 Ingeniería y Calidad del 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

Con TECH, no solo aprenderás a desarrollar Software, sino a crear sistemas que marcan la diferencia en la vida de las personas y las empresas"

Domina las habilidades y herramientas más avanzadas de la ingeniería con la metodología didáctica más novedosa del panorama académico actual"

Eleva tus expectativas profesionales aprendiendo de manera 100% online, sin interferir con tus responsabilidades personales y familiares.

Conviértete en un líder profesional en ingeniería, preparado para aprender desde cualquier lugar del mundo.

Incluye en su cuadro docente a profesionales pertenecientes al ámbito de la informática, 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.

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.

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 estudio del Grand Master de Ingeniería y Calidad del Software está diseñado para proporcionar una especialización integral y avanzada en todas las áreas clave de la ingeniería del Software. Los primeros módulos se centran en los fundamentos, abarcando desde el diseño de Software y la gestión de requisitos hasta las arquitecturas tecnológicas y las metodologías ágiles. A medida que avanza el programa, el alumnado se sumerge en áreas más especializadas como la automatización de pruebas, la integración continua y el aseguramiento de la calidad. Además, se incorporan asignaturas sobre gestión de proyectos tecnológicos, donde los participantes aprenden a dirigir equipos multidisciplinarios.

Este Grand Master te prepara para ser el experto que marca la diferencia en el sector de la Ingeniería y la Calidad del Software"

Módulo 1. Calidad del Software. Niveles de desarrollo TRL

1.1. Elementos que influyen en la Calidad de Software (I). La Deuda Técnica
- 1.1.1. La Deuda Técnica. Causas y consecuencias
- 1.1.2. Calidad del Software. Principios generales
- 1.1.3. Software sin principios y con principios de calidad
  - 1.1.3.1. Consecuencias
  - 1.1.3.2. Necesidad de aplicación de principios de calidad en el Software
- 1.1.4. Calidad del Software. Tipología
- 1.1.5. Software de Calidad. Rasgos específicos

1.2. Elementos que influyen en la Calidad de Software (II). Costes asociados
- 1.2.1. Calidad del Software. Elementos influyentes
- 1.2.2. Calidad del Software. Ideas erróneas
- 1.2.3. Calidad del Software. Costes asociados

1.3. Modelos de Calidad del Software (I). Gestión del conocimiento
- 1.3.1. Modelos de calidad generales
  - 1.3.1.1. Gestión de la calidad total
  - 1.3.1.2. Modelo Europeo de Excelencia Empresarial (EFQM)
  - 1.3.1.3. Modelo Seis-sigma
- 1.3.2. Modelos de la Gestión del Conocimiento
  - 1.3.2.1. Modelo Dyba
  - 1.3.2.2. Modelo Seks
- 1.3.3. Factoría de experiencia y Paradigma QIP
- 1.3.4. Modelos de calidad en el uso (25010)

1.4. Modelos de Calidad del Software (III). Calidad en datos, procesos y modelos SEI
- 1.4.1. Modelo de calidad de datos
- 1.4.2. Modelado del proceso Software
- 1.4.3. Software & *Systems Process Engineering Metamodel Specification* (SPEM)
- 1.4.4. Modelos del SEI
  - 1.4.4.1. CMMI
  - 1.4.4.2. SCAMPI
  - 1.4.4.3. IDEAL

1.5. Normas ISO de Calidad del Software (I). Análisis de los estándares
- 1.5.1. Normas ISO 9000
  - 1.5.1.1. Normas ISO 9000
  - 1.5.1.2. Familia ISO de Normas de Calidad (9000)
- 1.5.2. Otras normas ISO relacionadas con Calidad
- 1.5.3. Normas de Modelado de Calidad (ISO 2501)
- 1.5.4. Normas de Medida de la Calidad (ISO 2502n)

1.6. Normas ISO de Calidad del Software (II). Requisitos y evaluación
- 1.6.1. Normas sobre Requisitos de Calidad (2503n)
- 1.6.2. Normas sobre Evaluación de la Calidad (2504n)
- 1.6.3. ISO/IEC 24744:2007

1.7. Niveles de desarrollo TRL (I). Niveles el 1 al 4
- 1.7.1. Niveles TRL
- 1.7.2. Nivel 1: principios básicos
- 1.7.3. Nivel 2: concepto y/o aplicación
- 1.7.4. Nivel 3: función crítica analítica
- 1.7.5. Nivel 4: validación de componente en entorno de laboratorio

1.8. Niveles de desarrollo TRL (II). Niveles del 5 al 9
- 1.8.1. Nivel 5: validación de componente en entorno relevante
- 1.8.2. Nivel 6: modelo sistema/subsistema
- 1.8.3. Nivel 7: demostración en entorno real
- 1.8.4. Nivel 8: sistema completo y certificado
- 1.8.5. Nivel 9: éxito en el entorno real

1.9. Niveles de Desarrollo TRL. Usos
- 1.9.1. Ejemplo de empresa con entorno de laboratorio
- 1.9.2. Ejemplo de empresa I+D+i
- 1.9.3. Ejemplo de empresa de I+D+i industrial
- 1.9.4. Ejemplo de empresa mixta laboratorio-ingeniería

1.10. Calidad del Software. Detalles clave
- 1.10.1. Detalles metodológicos
- 1.10.2. Detalles técnicos
- 1.10.3. Detalles en la gestión de proyectos Software
  - 1.10.3.1. Calidad de los Sistemas Informáticos
  - 1.10.3.2. Calidad del Producto Software
  - 1.10.3.3. Calidad del Proceso Software

Módulo 2. Desarrollo de proyectos Software. Documentación funcional y técnica

2.1. Gestión de proyectos
- 2.1.1. Gestión de proyectos en la Calidad del Software
- 2.1.2. Gestión de proyectos. Ventajas
- 2.1.3. Gestión de proyectos. Tipología

2.2. Metodología en la gestión del proyecto
- 2.2.1. Metodología en la gestión del proyecto
- 2.2.2. Metodologías de proyectos. Tipología
- 2.2.3. Metodologías en la gestión de proyectos. Aplicación

2.3. Fase de Identificación de Requisitos
- 2.3.1. Identificación de los requisitos de un proyecto
- 2.3.2. Gestión de las reuniones de un proyecto
- 2.3.3. Documentación a aportar

2.4. Modelo
- 2.4.1. Fase inicial
- 2.4.2. Fase de análisis
- 2.4.3. Fase de construcción
- 2.4.4. Fase de pruebas
- 2.4.5. Entrega

2.5. Modelo de Datos a utilizar
- 2.5.1. Determinación del nuevo Modelo de Datos
- 2.5.2. Identificación del Plan de Migración de Datos
- 2.5.3. Juego de datos

2.6. Repercusiones en otros proyectos
- 2.6.1. Repercusión de un proyecto. Ejemplos

2.7. *MUST* del proyecto
- 2.7.1. *MUST* de proyecto
- 2.7.2. Identificación de los *MUST* del proyecto
- 2.7.3. Identificación de los puntos de ejecución para la entrega de un proyecto

2.8. El equipo para la construcción del proyecto
- 2.8.1. Roles a intervenir según el proyecto
- 2.8.2. Contacto con RRHH para contratación
- 2.8.3. Entregables y calendario del proyecto

2.9. Aspectos ´técnicos de un proyecto Software
- 2.9.1. Arquitecto del proyecto. Aspectos Técnicos
- 2.9.2. Líderes Técnicos
- 2.9.3. Construcción del proyecto Software
- 2.9.4. Evaluación de la calidad del código Sonar

2.10. Entregables del proyecto
- 2.10.1. Análisis funcional
- 2.10.2. Modelo de datos
- 2.10.3. Diagrama de estados
- 2.10.4. Documentación técnica

Módulo 3. *Testing* de Software. Automatización de Pruebas

3.1. Modelos de calidad del Software
- 3.1.1. Calidad de producto
- 3.1.2. Calidad de proceso
- 3.1.3. Calidad de uso

3.2. Calidad de proceso
- 3.2.1. Calidad de proceso
- 3.2.2. Modelos de madurez
- 3.2.3. Normativa ISO 15504
  - 3.2.3.1. Propósitos
  - 3.2.3.2. Contexto
  - 3.2.3.3. Etapas

3.3. Normativa ISO/IEC 15504
- 3.3.1. Categorías de proceso
- 3.3.2. Proceso de Desarrollo. Ejemplo
- 3.3.3. Fragmento de perfil
- 3.3.4. Etapas

3.4. CMMI (*Capability Maturity Model Integration*)
- 3.4.1. CMMI. Integración de Modelos de Madurez de Capacidades
- 3.4.2. Modelos y áreas. Tipología
- 3.4.3. Áreas de proceso
- 3.4.4. Niveles de Capacidad
- 3.4.5. Administración de procesos
- 3.4.6. Administración de proyectos

3.5. Gestión de cambios y repositorios
- 3.5.1. Gestión de cambios en Software
  - 3.5.1.1. Ítem de configuración. Integración continua
  - 3.5.1.2. Líneas
  - 3.5.1.3. Flujogramas
  - 3.5.1.4. *Branches*
- 3.5.2. Repositorio
  - 3.5.2.1. Control de versiones
  - 3.5.2.2. Equipo de trabajo y uso del repositorio
  - 3.5.2.3. Integración continua en el repositorio

3.6. *Team Foundation Server* (TFS)
- 3.6.1. Instalación y configuración
- 3.6.2. Creación de un proyecto de equipo
- 3.6.3. Incorporación de contenido al control de código fuente
- 3.6.4. *TFS on Cloud*

3.7. *Testing*
- 3.7.1. Motivación para la realización de pruebas
- 3.7.2. Pruebas de verificación
- 3.7.3. Pruebas beta
- 3.7.4. Implementación y mantenimiento

3.8. Pruebas de carga
- 3.8.1. *Load testing*
- 3.8.2. Pruebas con *LoadView*
- 3.8.3. Pruebas con K6 *Cloud*
- 3.8.4. Pruebas con *Loader*

3.9. Pruebas unitarias de estrés y de resistencia
- 3.9.1. Motivación de las pruebas unitarias
- 3.9.2. Herramientas para *unit testing*
- 3.9.3. Motivación de las pruebas de estrés
- 3.9.4. Pruebas usando *StressTesting*
- 3.9.5. Motivación para las pruebas de resistencia
- 3.9.6. Pruebas usando *LoadRunner*

3.10. La escalabilidad. Diseño de Software escalable
- 3.10.1. La escalabilidad y la arquitectura del Software
- 3.10.2. La independencia entre capas
- 3.10.3. El Acoplamiento entre capas. Patrones de arquitectura

Módulo 4. Metodologías de gestión de proyectos Software. Metodologías Waterfall frente a Metodologías Agiles

4.1. Metodología Waterfall
- 4.1.1. Metodología Waterfall
- 4.1.2. Metodología Waterfall. Influencia en la Calidad del Software
- 4.1.3. Metodología Waterfall. Ejemplos

4.2. Metodología Agile
- 4.2.1. Metodología Agile
- 4.2.2. Metodología Agile. Influencia en la Calidad del Software
- 4.2.3. Metodología Agile. Ejemplos

4.3. Metodología SCRUM
- 4.3.1. Metodología SCRUM
- 4.3.2. Manifiesto SCRUM
- 4.3.3. Aplicación de SCRUM

4.4. Panel Kanban
- 4.4.1. Método Kanban
- 4.4.2. Panel Kanban
- 4.4.3. Panel Kanban. Ejemplo de aplicación

4.5. Gestión de Proyecto en Waterfall
- 4.5.1. Fases en un proyecto
- 4.5.2. Visión en un proyecto Waterfall
- 4.5.3. Entregables a tener en cuenta

4.6. Gestión de proyecto en SCRUM
- 4.6.1. Fases en un proyecto SCRUM
- 4.6.2. Visión en un proyecto SCRUM
- 4.6.3. Entregables a considerar

4.7. Waterfall vs SCRUM. Comparativa
- 4.7.1. Planteamiento de un proyecto piloto
- 4.7.2. Proyecto aplicando Waterfall. Ejemplo
- 4.7.3. Proyecto aplicando SCRUM. Ejemplo

4.8. Visión del Cliente
- 4.8.1. Documentos en un Waterfall
- 4.8.2. Documentos en un SCRUM
- 4.8.3. Comparativa

4.9. Estructura de Kanban
- 4.9.1. Historias de Usuario
- 4.9.2. Backlog
- 4.9.3. Análisis de Kanban

4.10. Proyectos Híbridos
- 4.10.1. Construcción del proyecto
- 4.10.2. Gestión proyecto
- 4.10.3. Entregables a considerar

Módulo 5.TDD (*Test Driven Developement*). Diseño de Software guiado por las pruebas

5.1. TDD. *Test Driven Development*
- 5.1.1. TDD. *Test Driven Development*
- 5.1.2. TDD. Influencia del TDD en la calidad
- 5.1.3. Diseño y desarrollo basado en Pruebas. Ejemplos

5.2. Ciclo de TDD
- 5.2.1. Elección de un requisito
- 5.2.2. Realización de pruebas. Tipologías
  - 5.2.2.1. Pruebas unitarias
  - 5.2.2.2. Pruebas de Integración
  - 5.2.2.3. Pruebas *End To End*
- 5.2.3. Verificación de la prueba. Fallos
- 5.2.4. Creación de la Implementación
- 5.2.5. Ejecución de las pruebas automatizadas
- 5.2.6. Eliminación de la duplicación
- 5.2.7. Actualización de la lista de requisitos
- 5.2.8. Repetición del ciclo TDD
- 5.2.9. Ciclo TDD. Ejemplo teórico-práctico

5.3. Estrategias de Implementación de TDD
- 5.3.1. Implementación falsa
- 5.3.2. Implementación triangular
- 5.3.3. Implementación obvia

5.4. TDD. Uso. Ventajas e Inconvenientes
- 5.4.1. Ventajas de uso
- 5.4.2. Limitaciones de uso
- 5.4.3. Balance de calidad en la implementación

5.5. TDD. Buenas prácticas
- 5.5.1. Reglas TDD
- 5.5.2. Regla 1: tener un test previo que falle antes de codificar en producción
- 5.5.3. Regla 2: no escribir más de un test unitario
- 5.5.4. Regla 3: no escribir más código de lo necesario
- 5.5.5. Errores y anti patrones a evitar en una TDD

5.6. Simulación de proyecto real para usar TDD (I)
- 5.6.1. Descripción general del proyecto (Empresa A)
- 5.6.2. Aplicación de la TDD
- 5.6.3. Ejercicios propuestos
- 5.6.4. Ejercicios. *Feedback*

5.7. Simulación de proyecto real para usar TDD (II)
- 5.7.1. Descripción general del proyecto (Empresa B)
- 5.7.2. Aplicación de la TDD
- 5.7.3. Ejercicios propuestos
- 5.7.4. Ejercicios. *Feedback*

5.8. Simulación de proyecto real para usar TDD (III)
- 5.8.1. Descripción general del proyecto (Empresa C)
- 5.8.2. Aplicación de la TDD
- 5.8.3. Ejercicios propuestos
- 5.8.4. Ejercicios. *Feedback*

5.9. Alternativas a TDD. *Test Driven Development*
- 5.9.1. TCR (*Test Commit Revert*)
- 5.9.2. BDD (*Behavior Driven Development*)
- 5.9.3. ATDD (*Acceptance Test Driven Development*)
- 5.9.4. TDD. Comparativa teórica

5.10. TDD TCR, BDD y ATDD. Comparación práctica
- 5.10.1. Definición del problema
- 5.10.2. Resolución con TCR
- 5.10.3. Resolución con BDD
- 5.10.4. Resolución con ATDD

Módulo 6. DevOps. Gestión de Calidad del Software

6.1. DevOps. Gestión de Calidad del Software
- 6.1.1. DevOps
- 6.1.2. DevOps y Calidad del Software
- 6.1.3. DevOps. Beneficios de la cultura DevOps

6.2. DevOps. Relación con Agile
- 6.2.1. Entrega acelerada
- 6.2.2. Calidad
- 6.2.3. Reducción de costes

6.3. Puesta en marcha de DevOps
- 6.3.1. Identificación de problemas
- 6.3.2. Implantación en una compañía
- 6.3.3. Métricas de Implantación

6.4. Ciclo de entrega de Software
- 6.4.1. Métodos de Diseño
- 6.4.2. Convenios
- 6.4.3. Hoja de ruta

6.5. Desarrollo de Código Libre de Errores
- 6.5.1. Código mantenible
- 6.5.2. Patrones de desarrollo
- 6.5.3. *Testing* de Código
- 6.5.4. Desarrollo de Software a Nivel de Código. Buenas prácticas

6.6. Automatización
- 6.6.1. Automatización. Tipos de pruebas
- 6.6.2. Coste de la automatización y mantenimiento
- 6.6.3. Automatización. Mitigando errores

6.7. Despliegues
- 6.7.1. Valoración de objetivos
- 6.7.2. Diseño de un proceso automático y adaptado
- 6.7.3. Retroalimentación y capacidad de respuesta

6.8. Gestión de Incidentes
- 6.8.1. Preparación para incidentes
- 6.8.2. Análisis y resolución del incidente
- 6.8.3. Cómo evitar futuros errores

6.9. Automatización de despliegues
- 6.9.1. Preparación para despliegues automáticos
- 6.9.2. Evaluación de la salud del proceso automático
- 6.9.3. Métricas y capacidad de vuelta atrás

6.10. Buenas prácticas. Evolución de DevOps
- 6.10.1. Guía de buenas prácticas aplicando DevOps
- 6.10.2. DevOps. Metodología para el equipo
- 6.10.3. Evitando nichos

Módulo 7. DevOps e integración continua. Soluciones prácticas avanzadas en desarrollo de Software

7.1. Flujo de la entrega de Software
- 7.1.1. Identificación de actores y artefactos
- 7.1.2. Diseño del flujo de entrega de Software
- 7.1.3. Flujo de entrega de Software. Requisitos entre etapas

7.2. Automatización de procesos
- 7.2.1. Integración continua
- 7.2.2. Despliegue continuo
- 7.2.3. Configuración de entornos y gestión de eecretos

7.3. Pipelines declarativos
- 7.3.1. Diferencias entre pipelines tradicionales, como código y declarativos
- 7.3.2. Pipelines declarativos
- 7.3.3. Pipelines declarativos en Jenkins
- 7.3.4. Comparación de proveedores de integración continua

7.4. Puertas de calidad y retroalimentación enriquecida
- 7.4.1. Puertas de calidad
- 7.4.2. Estándares de Calidad con puertas de calidad. Mantenimiento
- 7.4.3. Requisitos de negocio en las solicitudes de integración

7.5. Gestión de artefactos
- 7.5.1. Artefactos y ciclo de Vida
- 7.5.2. Sistemas de almacenamiento y gestión de artefactos
- 7.5.3. Seguridad en la gestión de artefactos

7.6. Despliegue continuo
- 7.6.1. Despliegue continuo como contenedores
- 7.6.2. Despliegue continuo con PaaS

7.7. Mejora del tiempo de ejecución del Pipeline: análisis estático y *Git Hooks*
- 7.7.1. Análisis estático
- 7.7.2. Reglas de estilo del código
- 7.7.3. *Git Hooks* y *Tests Unitarios*
- 7.7.4. El impacto de la infraestructura

7.8. Vulnerabilidades en contenedores
- 7.8.1. Vulnerabilidades en contenedores
- 7.8.2. Escaneo de imágenes
- 7.8.3. Informes periódicos y alertas

Módulo 8. Diseño de Bases de Datos (BD). Normalización y rendimiento. Calidad del Software

8.1. Diseño de bases de datos
- 8.1.1. Bases de Datos. Tipología
- 8.1.2. Bases de datos usados actualmente
  - 8.1.2.1. Relacionales
  - 8.1.2.2. Clave-Valor
  - 8.1.2.3. Basadas en grafos
- 8.1.3. La Calidad del dato

8.2. Diseño del Modelo Entidad-Relación (I)
- 8.2.1. Modelo de Entidad-Relación. Calidad y documentación
- 8.2.2. Entidades
  - 8.2.2.1. Entidad fuerte
  - 8.2.2.2. Entidad débil
- 8.2.3. Atributos
- 8.2.4. Conjunto de Relaciones
  - 8.2.4.1. 1 a 1
  - 8.2.4.2. 1 a muchos
  - 8.2.4.3. Muchos a 1
  - 8.2.4.4. Muchos a muchos
- 8.2.5. Claves
  - 8.2.5.1. Clave primaria
  - 8.2.5.2. Clave foránea
  - 8.2.5.3. Clave primaria entidad débil
- 8.2.6. Restricciones
- 8.2.7. Cardinalidad
- 8.2.8. Herencia
- 8.2.9. Agregación

8.3. Modelo Entidad-Relación (II). Herramientas
- 8.3.1. Modelo Entidad-Relación. Herramientas
- 8.3.2. Modelo Entidad-Relación. Ejemplo práctico
- 8.3.3. Modelo Entidad-Relación factible
  - 8.3.3.1. Muestra visual
  - 8.3.3.2. Muestra en representación de tablas

8.4. Normalización de la base de datos (BD) (I). Consideraciones en Calidad del Software
- 8.4.1. Normalización de la BD y calidad
- 8.4.2. Dependencias
  - 8.4.2.1. Dependencia funcional
  - 8.4.2.2. Propiedades de la dependencia funcional
  - 8.4.2.3. Propiedades deducidas
- 8.4.3. Claves

8.5. Normalización de la base de datos (BD) (II). Formas Normales y Reglas del Codd
- 8.5.1. Formas Normales
  - 8.5.1.1. Primera Forma Normal (1FN)
  - 8.5.1.2. Segunda Forma Normal (2FN)
  - 8.5.1.3. Tercera Forma Normal (3FN)
  - 8.5.1.4. Forma normal de Boyce-Codd (FNBC)
  - 8.5.1.5. Cuarta Forma Normal (4FN)
  - 8.5.1.6. Quinta Forma Normal (5FN)
- 8.5.2. Reglas de Codd
  - 8.5.2.1. Regla 1: información
  - 8.5.2.2. Regla 2: acceso garantizado
  - 8.5.2.3. Regla 3: tratamiento sistemático de los valores nulos
  - 8.5.2.4. Regla 4: descripción de la base de datos
  - 8.5.2.5. Regla 5: sub-lenguaje integral
  - 8.5.2.6. Regla 6: actualización de vistas
  - 8.5.2.7. Regla 7: insertar y actualizar
  - 8.5.2.8. Regla 8: independencia física
  - 8.5.2.9. Regla 9: independencia lógica
  - 8.5.2.10. Regla 10: independencia de la integridad
    - 8.5.2.10.1. Reglas de integridad
  - 8.5.2.11. Regla 11: distribución
  - 8.5.2.12. Regla 12: no-subversión
- 8.5.3. Ejemplo práctico

8.6. Almacén de datos / Sistema OLAP
- 8.6.1. Almacén de datos
- 8.6.2. Tabla de hechos
- 8.6.3. Tabla de dimensiones
- 8.6.4. Creación del sistema OLAP. Herramientas

8.7. Rendimiento de la base de datos (BD)
- 8.7.1. Optimización de índices
- 8.7.2. Optimización de consultas
- 8.7.3. Particionado de tablas

8.8. Simulación de proyecto real para diseño BD (I)
- 8.8.1. Descripción general del proyecto (Empresa A)
- 8.8.2. Aplicación del diseño de base de datos
- 8.8.3. Ejercicios propuestos
- 8.8.4. Ejercicios propuestos. *Feedback*

8.9. Simulación de proyecto real para diseño BD (II)
- 8.9.1. Descripción general del proyecto (Empresa B)
- 8.9.2. Aplicación del diseño de bases de datos
- 8.9.3. Ejercicios Propuestos
- 8.9.4. Ejercicios Propuestos. *Feedback*

8.10. Relevancia de la optimización de BBDD en la Calidad del Software
- 8.10.1. Optimización del Diseño
- 8.10.2. Optimización del Código de Consultas
- 8.10.3. Optimización del Código de Procedimientos almacenados
- 8.10.4. Influencia de los *Triggers* en la Calidad del Software. Recomendaciones de uso

Módulo 9. Diseño de arquitecturas escalables. La arquitectura en el ciclo de vida del Software

9.1. Diseño de arquitecturas escalables (I)
- 9.1.1. Arquitecturas escalables
- 9.1.2. Principios de una arquitectura escalable
  - 9.1.2.1. Confiable
  - 9.1.2.2. Escalable
  - 9.1.2.3. Mantenible
- 9.1.3. Tipos de escalabilidad
  - 9.1.3.1. Vertical
  - 9.1.3.2. Horizontal
  - 9.1.3.3. Combinado

9.2. Arquitecturas DDD (*Domain-Driven Design*)
- 9.2.1. El Modelo DDD. Orientación al Dominio
- 9.2.2. Capas, reparto de responsabilidad y patrones de diseño
- 9.2.3. Desacoplamiento como base de la calidad

9.3. Diseño de arquitecturas escalables (II). Benéficos, limitaciones y estrategias de diseño
- 9.3.1. Arquitectura escalable. Beneficios
- 9.3.2. Arquitectura escalable. Limitaciones
- 9.3.3. Estrategias para el desarrollo de arquitecturas escalables (tabla descriptiva)

9.4. Ciclo de vida del Software (I). Etapas
- 9.4.1. Ciclo de vida del Software
  - 9.4.1.1. Etapa de planificación
  - 9.4.1.2. Etapa de análisis
  - 9.4.1.3. Etapa de diseño
  - 9.4.1.4. Etapa de implementación
  - 9.4.1.5. Etapa de pruebas
  - 9.4.1.6. Etapa de instalación/despliegue
  - 9.4.1.7. Etapa de uso y mantenimiento

9.5. Modelos de ciclos de vida del Software
- 9.5.1. Modelo en cascada
- 9.5.2. Modelo repetitivo
- 9.5.3. Modelo en espiral
- 9.5.4. Modelo Big Bang

9.6. Ciclo de vida del Software (II). Automatización
- 9.6.1. Ciclos de Vida de Desarrollo de Software. Soluciones
  - 9.6.1.1. Integración y desarrollo continuos (CI/CD)
  - 9.6.1.2. Metodologías Agile
  - 9.6.1.3. DevOps / Operaciones de Producción
- 9.6.2. Tendencias futuras
- 9.6.3. Ejemplos prácticos

9.7. Arquitectura Software en el ciclo de vida del Software
- 9.7.1. Beneficios
- 9.7.2. Limitaciones
- 9.7.3. Herramientas

9.8. Simulación de proyecto real para diseño de arquitectura Software (I)
- 9.8.1. Descripción general del proyecto (empresa A)
- 9.8.2. Aplicación del diseño de arquitectura del Software
- 9.8.3. Ejercicios propuestos
- 9.8.4. Ejercicios propuestos. *Feedback*

9.9. Simulación de proyecto real para para diseño de arquitectura Software (II)
- 9.9.1. Descripción general del proyecto (Empresa B)
- 9.9.2. Aplicación del diseño de arquitectura del Software
- 9.9.3. Ejercicios propuestos
- 9.9.4. Ejercicios propuestos *Feedback*

9.10. Simulación de proyecto real para para diseño de arquitectura Software (III)
- 9.10.1. Descripción general del proyecto (Empresa C)
- 9.10.2. Aplicación del diseño de arquitectura del Software
- 9.10.3. Ejercicios Propuestos
- 9.10.4. Ejercicios Propuestos. *Feedback*

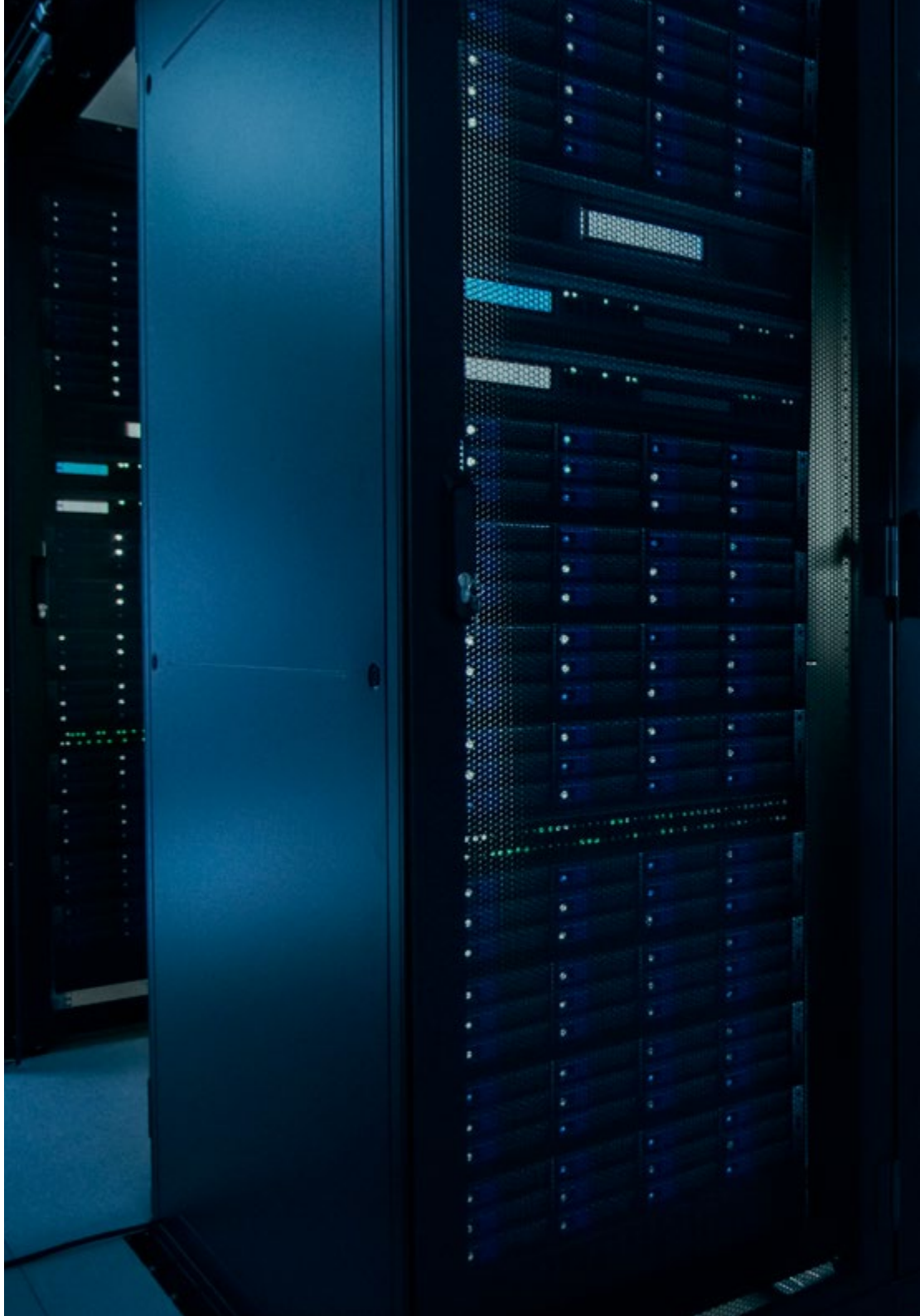
Módulo 10. Criterios de Calidad ISO, IEC 9126. Métrica de Calidad del Software

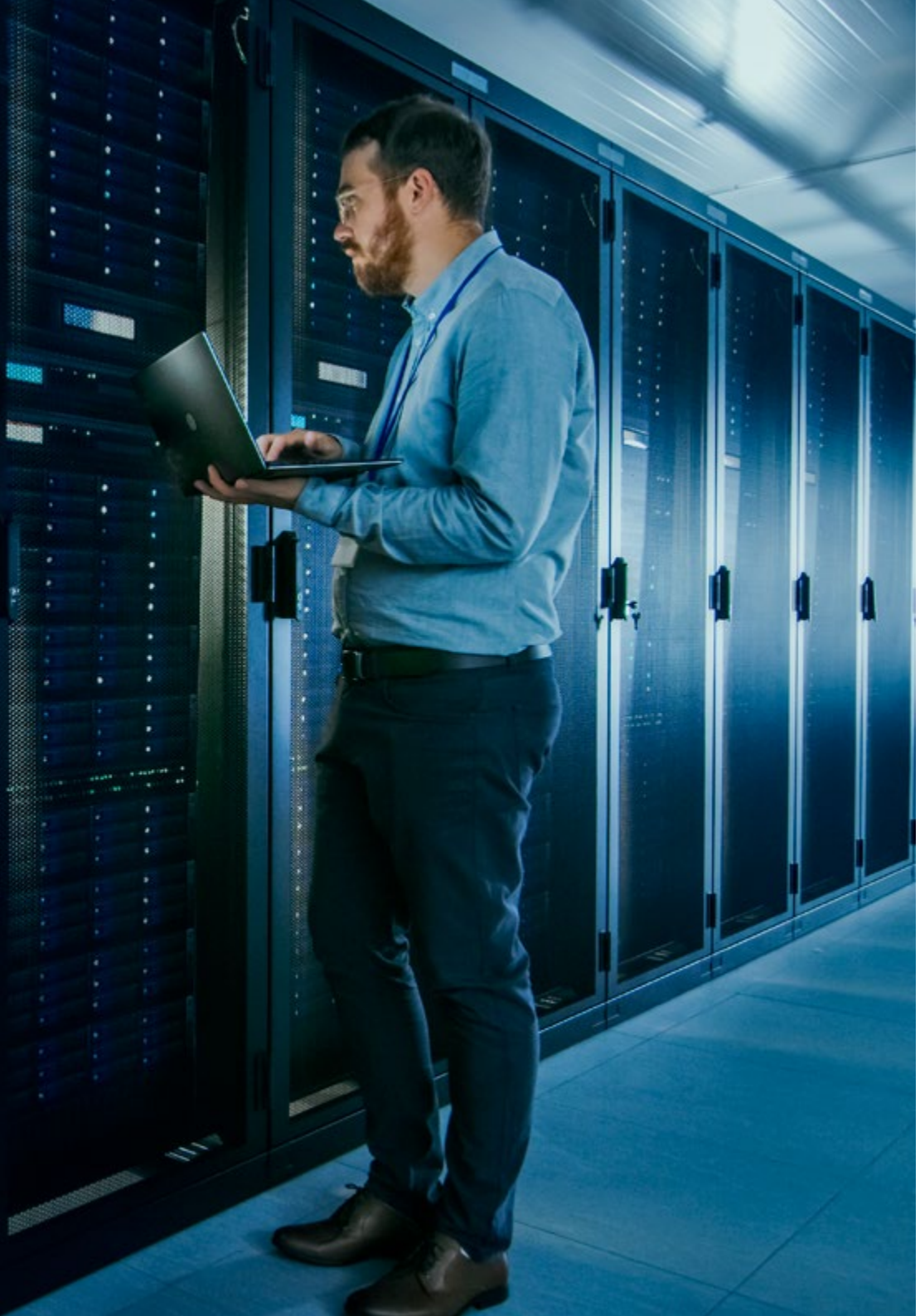
10.1. Criterios de calidad. Norma ISO, IEC 9126
- 10.1.1. Criterio de calidad
- 10.1.2. Calidad del Software. Justificación. Norma ISO, IEC 9126
- 10.1.3. La Medición de la calidad del Software como Indicador clave

10.2. Criterios de la calidad del Software. Características
- 10.2.1. Fiabilidad
- 10.2.2. Funcionalidad
- 10.2.3. Eficiencia
- 10.2.4. Usabilidad
- 10.2.5. Mantenibilidad
- 10.2.6. Portabilidad

10.3. Norma ISO, IEC 9126 (I). Presentación
- 10.3.1. Descripción de la Norma ISO, IEC 9126
- 10.3.2. Funcionalidad
- 10.3.3. Fiabilidad
- 10.3.4. Usabilidad
- 10.3.5. Mantenibilidad

10.3.6. Portabilidad
10.3.7. Calidad en uso
10.3.8. Métricas de calidad del Software
10.3.9. Métricas de calidad en ISO 9126
10.4. Norma ISO, IEC 9126 (II). Modelos McCall y Boehm
10.4.1. Modelo McCall: Factores de calidad
10.4.2. Modelo Boehm
10.4.3. Nivel intermedio. Características
10.5. Métrica de calidad del Software (I). Elementos
10.5.1. Medida
10.5.2. Métrica
10.5.3. Indicador
10.5.3.1. Tipos de indicadores
10.5.4. Medidas y modelos
10.5.5. Alcance de las métricas del Software
10.5.6. Clasificación de las métricas del Software
10.6. Medición de calidad del Software (II). Práctica de la medición
10.6.1. Recogida de datos métricos
10.6.2. Medición de atributos internos del producto
10.6.3. Medición de atributos externos del producto
10.6.4. Medición de recursos
10.6.5. Métricas para sistemas orientados a objetos
10.7. Diseño de un indicador único de calidad del Software
10.7.1. Indicador único como calificador global
10.7.2. Desarrollo del indicador, justificación y Aplicación
10.7.3. Ejemplo de aplicación. Necesidad de conocer el detalle
10.8. Simulación de proyecto real para medición de calidad (I)
10.8.1. Descripción general del proyecto (empresa A)
10.8.2. Aplicación de la medición de calidad
10.8.3. Ejercicios propuestos
10.8.4. Ejercicios propuestos. *Feedback*





10.9. Simulación de proyecto real para medición de calidad (II)
- 10.9.1. Descripción general del proyecto (empresa B)
- 10.9.2. Aplicación de la medición de calidad
- 10.9.3. Ejercicios propuestos
- 10.9.4. Ejercicios propuestos. *Feedback*

10.10. Simulación de proyecto real para medición de calidad (III)
- 10.10.1. Descripción general del proyecto (empresa C)
- 10.10.2. Aplicación de la medición de calidad
- 10.10.3. Ejercicios propuestos
- 10.10.4. Ejercicios propuestos. *Feedback*

Módulo 11. Metodologías, desarrollo y calidad en la Ingeniería de Software

11.1. Desarrollo de Software basado en modelos
- 11.1.1. La necesidad de
- 11.1.3. Modelado de objetos
- 11.1.4. UML
- 11.1.5. Herramientas CASE

11.2. Modelado de aplicaciones y patrones de diseño con UML
- 11.2.1. Modelado avanzado de requisitos
- 11.2.2. Modelado estático avanzado
- 11.2.3. Modelado dinámico avanzado
- 11.2.4. Modelado de componentes
- 11.2.5. Introducción a los patrones de diseño con UML
- 11.2.6. Adapter
- 11.2.7. Factory
- 11.2.8. Singleton
- 11.2.9. Strategy
- 11.2.10. Composite
- 11.2.11. Facade
- 11.2.12. Observer

11.3. Ingeniería dirigida por modelos
- 11.3.1. Introducción
- 11.3.2. Metamodelado de sistemas
- 11.3.3. MDA
- 11.3.4. DSL
- 11.3.5. Refinamientos de modelos con OCL
- 11.3.6. Transformaciones de modelos

11.4. Ontologías en la ingeniería de Software
- 11.4.1. Introducción
- 11.4.2. Ingeniería de la ontología
- 11.4.3. Aplicación de las ontologías en la ingeniería de Software

Módulo 12. Gestión de proyectos de Software

12.1. La gestión de los *stakeholders* y del alcance
- 12.1.1. Identificar a los interesados
- 12.1.2. Desarrollar el plan para la gestión de los interesados
- 12.1.3. Gestionar el compromiso de los interesados
- 12.1.4. Controlar el compromiso de los interesados
- 12.1.5. El objetivo del proyecto
- 12.1.6. La gestión del alcance y su plan
- 12.1.7. Recopilar los requisitos
- 12.1.8. Definir el enunciado del alcance
- 12.1.9. Crear la WBS (EDT)
- 12.1.10. Verificar y controlar el alcance

12.2. El desarrollo del cronograma
- 12.2.1. La gestión del tiempo y su plan
- 12.2.2. Definir las actividades
- 12.2.3. Establecimiento de la secuencia de las actividades
- 12.2.4. Estimación de recursos de las actividades
- 12.2.5. Estimación de la duración de las actividades
- 12.2.6. Desarrollo del cronograma y cálculo del camino crítico
- 12.2.7. Control del cronograma

12.3. El desarrollo del presupuesto y la respuesta a los riesgos
- 12.3.1. Estimar los costes
- 12.3.2. Desarrollar el presupuesto y la curva S
- 12.3.3. Control de costes y método del valor ganado
- 12.3.4. Los conceptos de riesgo
- 12.3.5. Cómo hacer un análisis de riesgos
- 12.3.6. El desarrollo del plan de respuesta

12.4. La comunicación y los recursos humanos
- 12.4.1. Planificar la gestión de las comunicaciones
- 12.4.2. Análisis de requisitos de comunicaciones
- 12.4.3. Tecnología de las comunicaciones
- 12.4.4. Modelos de comunicación
- 12.4.5. Métodos de comunicación
- 12.4.6. Plan de gestión de las comunicaciones
- 12.4.7. Gestionar las comunicaciones
- 12.4.8. La gestión de los recursos humanos
- 12.4.9. Principales actores y sus roles en los proyectos
- 12.4.10. Tipos de organizaciones
- 12.4.11. Organización del proyecto
- 12.4.12. El equipo de trabajo

12.5. El aprovisionamiento
- 12.5.1. El proceso de adquisiciones
- 12.5.2. Planificación
- 12.5.3. Búsqueda de suministradores y solicitud de ofertas
- 12.5.4. Adjudicación del contrato
- 12.5.5. Administración del contrato
- 12.5.6. Los contratos
- 12.5.7. Tipos de contratos
- 12.5.8. Negociación del contrato

12.6. Ejecución, monitorización y control y cierre
- 12.6.1. Los grupos de procesos
- 12.6.2. La ejecución del proyecto
- 12.6.3. La monitorización y control del proyecto
- 12.6.4. El cierre del proyecto

12.7. Responsabilidad profesional
- 12.7.1. Responsabilidad profesional
- 12.7.2. Características de la responsabilidad social y profesional
- 12.7.3. Código deontológico del líder de proyectos
- 12.7.4. Responsabilidad vs. PMP®
- 12.7.5. Ejemplos de responsabilidad
- 12.7.6. Beneficios de la profesionalización

Módulo 13. Plataformas de desarrollo del Software

13.1. Introducción al desarrollo de aplicaciones
- 13.1.1. Aplicaciones de escritorio
- 13.1.2. Lenguaje de programación
- 13.1.3. Entornos de desarrollo integrado
- 13.1.4. Aplicaciones web
- 13.1.5. Aplicaciones móviles
- 13.1.6. Aplicaciones en la nube

13.2. Desarrollo de aplicaciones e interfaz gráfica en Java
- 13.2.1. Entornos de desarrollo integrados para Java
- 13.2.2. Principales IDE para Java
- 13.2.3. Introducción a la plataforma de desarrollo Eclipse
- 13.2.4. Introducción a la plataforma de desarrollo NetBeans
- 13.2.5. Modelo Vista Controlador para las interfaces gráficas de usuario
- 13.2.6. Diseñar una interfaz gráfica en Eclipse
- 13.2.7. Diseñar una interfaz gráfica en NetBeans

13.3. Depuración y pruebas en Java
- 13.3.1. Pruebas y depuración de programas en Java
- 13.3.2. Depuración en Eclipse
- 13.3.3. Depuración en NetBeans

13.4. Desarrollo de aplicaciones e interfaz gráfica en .NET
- 13.4.1. Net Framework
- 13.4.2. Componentes de la plataforma de desarrollo .NET
- 13.4.3. Visual Studio .NET
- 13.4.4. Herramientas de .NET para GUI
- 13.4.5. La GUI con Windows Presentation Foundation
- 13.4.6. Depurar y compilar una aplicación de WPF

13.5. Programación para redes .NET
- 13.5.1. Introducción a la programación para redes en .NET
- 13.5.2. Peticiones y respuestas en .NET
- 13.5.3. Uso de protocolos de aplicación en .NET
- 13.5.4. Seguridad en la programación para redes en .NET

13.6. Entornos de desarrollo de aplicaciones móviles
- 13.6.1. Aplicaciones móviles
- 13.6.2. Aplicaciones móviles Android
- 13.6.3. Pasos para el desarrollo en Android
- 13.6.4. El IDE Android Studio

13.7. Desarrollo de aplicaciones en el entorno Android Studio
- 13.7.1. Instalar e iniciar Android Studio
- 13.7.2. Ejecución de una aplicación Android
- 13.7.3. Desarrollo de la interfaz gráfica en Android Studio
- 13.7.4. Iniciando actividades en Android Studio

13.8. Depuración y publicación de aplicaciones Android
- 13.8.1. Depuración de una aplicación en Android Studio
- 13.8.2. Memorizar aplicaciones en Android Studio
- 13.8.3. Publicación de una aplicación en Google Play

13.9. Desarrollo de aplicaciones para la nube
- 13.9.1. *Cloud computing*
- 13.9.2. Niveles de *cloud*: SaaS, PaaS, IaaS
- 13.9.3. Principales plataformas de desarrollo en la nube
- 13.9.4. Referencias bibliográficas

13.10. Introducción a Google Cloud Platform
- 13.10.1. Conceptos básicos de Google Cloud Platform
- 13.10.2. Servicios de Google Cloud Platform
- 13.10.3. Herramientas de Google Cloud Platform

Módulo 14. Computación en el cliente web

14.1. Introducción a HTML
- 14.1.1. Estructura de un documento
- 14.1.2. Color
- 14.1.3. Texto
- 14.1.4. Enlaces de hipertexto
- 14.1.5. Imágenes
- 14.1.6. Listas
- 14.1.7. Tablas
- 14.1.8. Marcos (*frames*)
- 14.1.9. Formularios
- 14.1.10. Elementos específicos para tecnologías móviles
- 14.1.11. Elementos en desuso

14.2. Hojas de estilo web (CSS)
- 14.2.1. Elementos y estructura de una hoja de estilos
  - 14.2.1.1. Creación de hojas de estilo
  - 14.2.1.2. Aplicación de estilos. Selectores
  - 14.2.1.3. Herencia de estilos y aplicación en cascada
  - 14.2.1.4. Formateado de páginas mediante estilos
  - 14.2.1.5. Estructura de páginas mediante estilos. El modelo de cajas
- 14.2.2. Diseño de estilos para diferentes dispositivos
- 14.2.3. Tipos de hojas de estilos: estáticas y dinámicas. Las Pseudoclases
- 14.2.4. Buenas prácticas en el uso de hojas de estilo

14.3. Introducción e historia de JavaScript
- 14.3.1. Introducción
- 14.3.2. Historia de JavaScript
- 14.3.3. Entorno de desarrollo que vamos a usar

14.4. Nociones básicas de programación web
- 14.4.1. Sintaxis básica de JavaScript
- 14.4.2. Tipos de datos primitivos y operadores
- 14.4.3. Variables y ámbitos
- 14.4.4. Cadenas de texto y *template literals*
- 14.4.5. Números y booleanos
- 14.4.6. Comparaciones

14.5. Estructuras complejas en JavaScript
- 14.5.1. Vectores o *arrays* y objetos
- 14.5.2. Conjuntos
- 14.5.3. Mapas
- 14.5.4. Disyuntivas
- 14.5.5. Bucles

14.6. Funciones y objetos
- 14.6.1. Definición e invocación de funciones
- 14.6.2. Argumentos
- 14.6.3. Funciones flecha
- 14.6.4. Funciones de retrollamada o *callback*
- 14.6.5. Funciones de orden superior
- 14.6.6. Objetos literales
- 14.6.7. El objeto *this*
- 14.6.8. Objetos como espacios de nombres: el objeto Math y el objeto Date

14.7. El modelo de objetos del documento (DOM)
- 14.7.1. ¿Qué es el DOM?
- 14.7.2. Un poco de historia
- 14.7.3. Navegación y obtención de elementos
- 14.7.4. Un DOM virtual con JSDOM
- 14.7.5. Selectores de consulta o *query selectors*
- 14.7.6. Navegación mediante propiedades
- 14.7.7. Asignación de atributos a los elementos
- 14.7.8. Creación y modificación de nodos
- 14.7.9. Actualización del estilo de los elementos del DOM

14.8. Desarrollo web moderno
- 14.8.1. Flujo basado en eventos y *listeners*
- 14.8.2. *Toolkits* web modernos y sistemas de alineamiento
- 14.8.3. Modo estricto de JavaScript
- 14.8.4. Algo más sobre funciones
- 14.8.5. Promesas y funciones asíncronas
- 14.8.6. *Closures*
- 14.8.7. Programación funcional
- 14.8.8. POO en JavaScript

14.9. Usabilidad web
- 14.9.1. Introducción a la usabilidad
- 14.9.2. Definición de usabilidad
- 14.9.3. Importancia del diseño web centrado en el usuario
- 14.9.4. Diferencias entre accesibilidad y usabilidad
- 14.9.5. Ventajas y problemas en la combinación de accesibilidad y usabilidad
- 14.9.6. Ventajas y dificultades en la implantación de sitios web usables
- 14.9.7. Métodos de usabilidad
- 14.9.8. Análisis de requerimiento de usuario
- 14.9.9. Principios del diseño conceptual. Creación de prototipos orientados al usuario
- 14.9.10. Pautas para la creación de sitios web usables
  - 14.9.10.1. Pautas de usabilidad de Jakob Nielsen
  - 14.9.10.2. Pautas de usabilidad de Bruce Tognazzini
- 14.9.11. Evaluación de la usabilidad

14.10. Accesibilidad web
- 14.10.1. Introducción
- 14.10.2. Definición de accesibilidad web
- 14.10.3. Tipos de discapacidades
  - 14.10.3.1. Discapacidades temporales o permanentes
  - 14.10.3.2. Discapacidades visuales
  - 14.10.3.3. Discapacidades auditivas
  - 14.10.3.4. Discapacidades motrices
  - 14.10.3.5. Discapacidad neurológicas o cognitivas
  - 14.10.3.6. Dificultades derivadas del envejecimiento
  - 14.10.3.7. Limitaciones derivadas del entorno
  - 14.10.3.8. Barreras que impiden el acceso a la web
- 14.10.4. Ayudas técnicas y productos de apoyo para superar las barreras
  - 14.10.4.1. Ayudas para personas ciegas
  - 14.10.4.2. Ayudas para persona con baja visión
  - 14.10.4.3. Ayudas para personas con daltonismo
  - 14.10.4.4. Ayudas para personas con discapacidad auditiva
  - 14.10.4.5. Ayudas para personas con discapacidad motriz
  - 14.10.4.6. Ayudas para personas con discapacidad cognitiva y neurológica
- 14.10.5. Ventajas y dificultades en la implantación de la accesibilidad web
- 14.10.6. Normativa y estándares sobre accesibilidad web
- 14.10.7. Organismos regulatorios de la accesibilidad web
- 14.10.8. Comparativa de normas y estándares
- 14.10.9. Guías para el cumplimiento de normativas y estándares
  - 14.10.9.1. Descripción de las pautas principales (imágenes, enlaces videos, etc.)
  - 14.10.9.2. Pautas para una navegación accesible
    - 14.10.9.2.1. Perceptibilidad
    - 14.10.9.2.2. Operatividad
    - 14.10.9.2.3. Comprensibilidad
    - 14.10.9.2.4. Robustez
- 14.10.10. Descripción del proceso de la conformidad en accesibilidad web
- 14.10.11. Niveles de conformidad
- 14.10.12. Criterios de conformidad
- 14.10.13. Requisitos de conformidad
- 14.10.14. Metodología de evaluación de la accesibilidad en sitios web

Módulo 15. Computación en Servidor Web

15.1. Introducción a la programación en el servidor: PHP
- 15.1.1. Conceptos básicos de programación en el servidor
- 15.1.2. Sintaxis básica de PHP
- 15.1.3. Generación de contenido HTML con PHP
- 15.1.4. Entornos de desarrollo y pruebas: XAMPP

15.2. PHP avanzado
- 15.2.1. Estructuras de control con PHP
- 15.2.2. Funciones en PHP
- 15.2.3. Manejo de *arrays* en PHP
- 15.2.4. Manejo de cadenas con PHP
- 15.2.5. Orientación a objetos en PHP

15.3. Modelos de datos
- 15.3.1. Concepto de dato. Ciclo de vida de los datos
- 15.3.2. Tipos de dato
  - 15.3.2.1. Básicos
  - 15.3.2.2. Registros
  - 15.3.2.3. Dinámicos

15.4. El modelo relacional
- 15.4.1. Descripción
- 15.4.2. Entidades y tipos de entidades
- 15.4.3. Elementos de datos. Atributos
- 15.4.4. Relaciones: tipos, subtipos, cardinalidad
- 15.4.5. Claves. Tipos de claves
- 15.4.6. Normalización. Formas normales

15.5. Construcción del modelo lógico de datos
- 15.5.1. Especificación de tablas
- 15.5.2. Definición de columnas
- 15.5.3. Especificación de claves
- 15.5.4. Conversión a formas normales. Dependencias

15.6. El modelo físico de datos. Ficheros de datos
- 15.6.1. Descripción de los ficheros de datos
- 15.6.2. Tipos de ficheros
- 15.6.3. Modos de acceso
- 15.6.4. Organización de ficheros

15.7. Acceso a bases de datos desde PHP
- 15.7.1. Introducción a MariaDB
- 15.7.2. Trabajar con una base de datos MariaDB: el lenguaje SQL
- 15.7.3. Acceder a la base de datos MariaDB desde PHP
- 15.7.4. Introducción a MySql
- 15.7.5. Trabajar con una base de datos MySql: el lenguaje SQL
- 15.7.6. Acceder a la base de datos MySql desde PHP

15.8. Interacción con el cliente desde PHP
- 15.8.1. Formularios PHP
- 15.8.2. Cookies
- 15.8.3. Manejo de sesiones

15.9. Arquitectura de aplicaciones web
- 15.9.1. El patrón Modelo Vista Controlador
- 15.9.2. Controlador
- 15.9.3. Modelo
- 15.9.4. Vista

15.10. Introducción a los servicios web
- 15.10.1. Introducción a XML
- 15.10.2. Arquitecturas orientas a servicios (SOA): servicios web
- 15.10.3. Creación de servicios web SOAP y REST
- 15.10.4. El protocolo SOAP
- 15.10.5. El protocolo REST

Módulo 16. Gestión de la seguridad

16.1. La seguridad de la información
- 16.1.1. Introducción
- 16.1.2. La seguridad de la información implica la confidencialidad, integridad y disponibilidad
- 16.1.3. La seguridad es un asunto económico
- 16.1.4. La seguridad es un proceso
- 16.1.5. La clasificación de la información
- 16.1.6. La seguridad en la información implica la gestión de los riesgos
- 16.1.7. La seguridad se articula con controles de seguridad
- 16.1.8. La seguridad es tanto física como lógica
- 16.1.9. La seguridad implica a las personas

16.2. El profesional de la seguridad de la información
- 16.2.1. Introducción
- 16.2.2. La seguridad de la información como profesión
- 16.2.3. Las certificaciones (ISC)2
- 16.2.4. El estándar ISO 27001
- 16.2.5. Buenas prácticas de seguridad en la gestión de servicios TI
- 16.2.6. Modelos de madurez para la seguridad de la información
- 16.2.7. Otras certificaciones, estándares y recursos profesionales

16.3. Control de accesos
- 16.3.1. Introducción
- 16.3.2. Requisitos del control de accesos
- 16.3.3. Mecanismos de autenticación
- 16.3.4. Métodos de autorización
- 16.3.5. Contabilidad y auditoría de accesos
- 16.3.6. Tecnologías Triple A

16.4. Programas, procesos y políticas de seguridad de la información
- 16.4.1. Introducción
- 16.4.2. Programas de gestión de la seguridad
- 16.4.3. La gestión de riesgos
- 16.4.4. Diseño de políticas de seguridad

16.5. Planes de continuidad de negocio
- 16.5.1. Introducción a los PCN
- 16.5.2. Fase I y II
- 16.5.3. Fase III y IV
- 16.5.4. Mantenimiento del PCN

16.6. Procedimientos para a correcta protección de la empresa
- 16.6.1. Redes DMZ
- 16.6.2. Sistemas de detección de intrusos
- 16.6.3. Listas de control de accesos
- 16.6.4. Aprender del atacante: Honeypot

16.7. Arquitectura de seguridad. Prevención
- 16.7.1. Visión general. Actividades y modelo de capas
- 16.7.2. Defensa perimetral (firewalls, WAFs, IPS, etc.)
- 16.7.3. Defensa del punto final (equipos, servidores y servicios)

16.8. Arquitectura de seguridad. Detección
- 16.8.1. Visión general detección y supervisión
- 16.8.2. Logs, ruptura de tráfico cifrado, grabación y Siems
- 16.8.3. Alertas e inteligencia

16.9. Arquitectura de seguridad. Reacción
- 16.9.1. Reacción. Productos, servicios y recursos
- 16.9.2. Gestión de Incidentes
- 16.9.3. CERTS y CSIRTs

16.10. Arquitectura de seguridad. Recuperación
- 16.10.1. Resiliencia, Conceptos, Requerimientos de Negocio y Normativa
- 16.10.2. Soluciones IT de Resiliencia
- 16.10.3. Gestión y Gobierno de las Crisis

Módulo 17. Seguridad en el Software

17.1. Problemas de la seguridad en el Software
- 17.1.1. Introducción al problema de la seguridad en el Software
- 17.1.2. Vulnerabilidades y su clasificación
- 17.1.3. Propiedades Software seguro
- 17.1.4. Referencias

17.2. Principios de diseño seguridad del Software
- 17.2.1. Introducción
- 17.2.2. Principios de diseño seguridad del Software
- 17.2.3. Tipos de S-SDLC
- 17.2.4. Seguridad del Software en las fases del S-SDLC
- 17.2.5. Metodologías y estándares
- 17.2.6. Referencias

17.3. Seguridad en el ciclo de vida del Software en las fases de requisitos y diseño
- 17.3.1. Introducción
- 17.3.2. Modelado de ataques
- 17.3.3. Casos de abuso
- 17.3.4. Ingeniería de requisitos de seguridad
- 17.3.5. Análisis de riesgo. Arquitectónico
- 17.3.6. Patrones de diseño
- 17.3.7. Referencias

17.4. Seguridad en el ciclo de vida del Software en las fases de codificación, pruebas y operación
- 17.4.1. Introducción
- 17.4.2. Pruebas de seguridad basadas en riesgo
- 17.4.3. Revisión de código
- 17.4.4. Test de penetración
- 17.4.5. Operaciones de seguridad
- 17.4.6. Revisión externa
- 17.4.7. Referencias

17.5. Codificación segura aplicaciones I
- 17.5.1. Introducción
- 17.5.2. Prácticas de codificación segura
- 17.5.3. Manipulación y validación de entradas
- 17.5.4. Desbordamiento de memoria
- 17.5.5. Referencias

17.6. Codificación segura aplicaciones II
- 17.6.1. Introducción
- 17.6.2. Integers overflows, errores de truncado y problemas con conversiones de tipo entre números enteros
- 17.6.3. Errores y excepciones
- 17.6.4. Privacidad y confidencialidad
- 17.6.5. Programas privilegiados
- 17.6.6. Referencias

17.7. Seguridad en el desarrollo y en la nube
- 17.7.1. Seguridad en el desarrollo; metodología y práctica
- 17.7.2. Modelos PaaS, IaaS,CaaS y SaaS
- 17.7.3. Seguridad en la nube y para servicios en la Nube

17.8. Cifrado
- 17.8.1. Fundamentos de la Criptología
- 17.8.2. Cifrado Simétrico y asimétrico
- 17.8.3. Cifrado en reposo y en tránsito

17.9. Automatización y orquestación de seguridad (SOAR)
- 17.9.1. Complejidad del tratamiento manual; necesidad de automatizar las tareas
- 17.9.2. Productos y servicios
- 17.9.3. Arquitectura SOAR

17.10. Seguridad en el Teletrabajo
- 17.10.1. Necesidad y escenarios
- 17.10.2. Productos y servicios
- 17.10.3. Seguridad en el teletrabajo

Módulo 18. Administración de Servidores Web

18.1. Introducción a servidores web
- 18.1.1. ¿Qué es un servidor web?
- 18.1.2. Arquitectura y funcionamiento de un servidor web
- 18.1.3. Recursos y contenidos en un servidor web
- 18.1.4. Servidores de aplicaciones
- 18.1.5. Servidores Proxy
- 18.1.6. Principales servidores web del mercado
- 18.1.7. Estadística de uso servidores web
- 18.1.8. Seguridad en servidores web
- 18.1.9. Balanceo de carga en servidores web
- 18.1.10. Referencias

18.2. Manejo del protocolo HTTP
- 18.2.1. Funcionamiento y estructura.
- 18.2.2. Descripción de peticiones o request methods
- 18.2.3. Códigos de estado
- 18.2.4. Cabeceras
- 18.2.5. Codificación del contenido. Páginas de códigos
- 18.2.6. Realización de peticiones HTTP en Internet mediante un proxy, livehttpheaders o método similar, analizando el protocolo utilizado

18.3. Descripción de arquitecturas distribuidas en múltiples servidores
- 18.3.1. Modelo de 3 capas
- 18.3.2. Tolerancia a fallos
- 18.3.3. Reparto de carga
- 18.3.4. Almacenes de estado de sesión
- 18.3.5. Almacenes de caché

18.4. Internet Information Services (IIS)
- 18.4.1. ¿Que es IIS?
- 18.4.2. Historia y evolución de IIS
- 18.4.3. Principales ventajas y características de IIS7 y posteriores
- 18.4.4. Arquitectura IIS7 y posteriores

18.5. Instalación, administración y configuración de IIS
- 18.5.1. Preámbulo
- 18.5.2. Instalación de Internet Information Services (IIS)
- 18.5.3. Herramientas de administración de IIS
- 18.5.4. Creación, configuración y administración de sitios web
- 18.5.5. Instalación y manejo de extensiones en IIS

18.6. Seguridad avanzada en IIS
- 18.6.1. Preámbulo
- 18.6.2. Autentificación, Autorización, y Control de Acceso en IIS
- 18.6.3. Configuración de un sitio web seguro en IIS con SSL
- 18.6.4. Políticas de seguridad implementada en IIS 8.x

18.7. Introducción a Apache
- 18.7.1. ¿Qué es Apache?
- 18.7.2. Principales ventajas de Apache
- 18.7.3. Características principales de Apache
- 18.7.4. Arquitectura

18.8. Instalación y configuración de Apache
- 18.8.1. Instalación inicial de apache
- 18.8.2. Configuración de apache

18.9. Instalación y configuración de los diferentes módulos en Apache
- 18.9.1. Instalación de módulos en Apache
- 18.9.2. Tipos de módulos
- 18.9.3. Configuración segura de Apache

18.10. Seguridad avanzada
- 18.10.1. Autenticación, autorización y control de acceso
- 18.10.2. Métodos de autenticación
- 18.10.3. Configuración segura de Apache con SSL

Módulo 19. Auditoría de Seguridad

19.1. Introducción a los sistemas de información y su auditoría
- 19.1.1. Introducción a los sistemas de información y el rol de la auditoría informática
- 19.1.2. Definiciones de auditoría informática y de control interno informático
- 19.1.3. Funciones y objetivos de la auditoría informática
- 19.1.4. Diferencias entre control interno y auditoría informática

19.2. Controles internos de los Sistemas de Información
- 19.2.1. Organigrama funcional de un centro de proceso de datos
- 19.2.2. Clasificación de los controles de los sistemas de información
- 19.2.3. La regla de oro

19.3. El proceso y las fases de la auditoría de Sistemas de Información
- 19.3.1. Evaluación de riesgos (EDR) y otras metodologías de auditoría informática
- 19.3.2. Ejecución de una auditoría de Sistemas de Información. Fases de auditoría
- 19.3.3. Habilidades fundamentales del auditor de Sistemas de Información

19.4. Auditoría técnica de seguridad en sistemas y redes
- 19.4.1. Auditorías técnicas de seguridad. Test de intrusión. Conceptos previos
- 19.4.2. Auditorías de seguridad en sistemas. Herramientas de apoyo
- 19.4.3. Auditorías de seguridad en redes. Herramientas de apoyo

19.5. Auditoría técnica de seguridad en internet y dispositivos móviles
- 19.5.1. Auditoría de seguridad en Internet. Herramientas de apoyo
- 19.5.2. Auditoría de seguridad en dispositivos móviles. Herramientas de apoyo
- 19.5.3. Anexo 1. Estructura de informe ejecutivo e informe técnico
- 19.5.4. Anexo 2. Inventario de herramientas
- 19.5.5. Anexo 3. Metodologías

19.6. Sistema de gestión de seguridad de la información
- 19.6.1. Seguridad de los SI: propiedades y factores de influencia
- 19.6.2. Riesgos empresariales y gestión de riesgos: implantación de controles
- 19.6.3. SG de la Seguridad de la Información (SGSI): concepto y factores críticos para el éxito
- 19.6.4. SGSI-Modelo PDCA
- 19.6.5. SGSI ISO-IEC 27001: contexto de la organización
- 19.6.6. Contexto de la organización
- 19.6.7. Liderazgo
- 19.6.8. Planificación
- 19.6.9. Soporte
- 19.6.10. Operación
- 19.6.11. Evaluación del desempeño
- 19.6.12. Mejora
- 19.6.13. Anexo a ISO 27001/ISO-IEC 27002: objetivos y controles
- 19.6.14. Auditoría del SGSI

19.7. Realización de la auditoría
- 19.7.1. Procedimientos
- 19.7.2. Técnicas

19.8. Trazabilidad
- 19.8.1. Metodologías
- 19.8.2. Análisis

19.9. Custodia
- 19.9.1. Técnicas
- 19.9.2. Resultados

19.10. Reportes y presentación de pruebas
- 19.10.1. Tipos de reportes
- 19.10.2. Análisis de los datos
- 19.10.3. Presentación de pruebas

Módulo 20. Seguridad en Aplicaciones Online

20.1. Vulnerabilidades y problemas de seguridad en las aplicaciones online
- 20.1.1. Introducción a la seguridad en las aplicaciones online
- 20.1.2. Vulnerabilidades de seguridad en el diseño de las aplicaciones web
- 20.1.3. Vulnerabilidades de seguridad en la implementación de las aplicaciones web
- 20.1.4. Vulnerabilidades de seguridad en el despliegue de las aplicaciones web
- 20.1.5. Listas oficiales de vulnerabilidades de seguridad

20.2. Políticas y estándares para la seguridad de las aplicaciones online
- 20.2.1. Pilares para la seguridad de las aplicaciones online
- 20.2.2. Política de seguridad
- 20.2.3. Sistema de gestión de seguridad de la información
- 20.2.4. Ciclo de vida de desarrollo seguro de Software
- 20.2.5. Estándares para la seguridad de las aplicaciones

20.3. Seguridad en el diseño de las aplicaciones web
- 20.3.1. Introducción a la seguridad de las aplicaciones web
- 20.3.2. Seguridad en el diseño de las aplicaciones web

20.4. Test de la seguridad y protección online de las aplicaciones web
- 20.4.1. Análisis y test de la seguridad de las aplicaciones web
- 20.4.2. Seguridad en el despliegue y producción de las aplicaciones web

20.5. Seguridad de los servicios web
- 20.5.1. Introducción a la seguridad de los servicios web
- 20.5.2. Funciones y tecnologías de la seguridad de los servicios web

20.6. Test de la seguridad y protección online de los servicios web
- 20.6.1. Evaluación de la seguridad de los servicios web
- 20.6.2. Protección online. Firewalls y gateways XML

20.7. Hacking ético, malware y forensic
- 20.7.1. Hacking ético
- 20.7.2. Análisis de Malware
- 20.7.3. Análisis Forense

20.8. Buenas prácticas para garantizar la seguridad en las aplicaciones
- 20.8.1. Manual de buenas prácticas en el desarrollo de las aplicaciones online
- 20.8.2. Manual de buenas prácticas en la implementación de las aplicaciones online

20.9. Errores comunes que perjudican la seguridad de las aplicaciones
- 20.9.1. Errores comunes en el desarrollo
- 20.9.2. Errores comunes en el hospedaje
- 20.9.3. Errores comunes en la producción

Un temario completo que te llevará a dominar el área del Big Data y convertirte en un arquitecto de estrategias empresariales exitoso”

04

Objetivos docentes

El Grand Master de Ingeniería y Calidad del Software tiene como objetivo formar a profesionales altamente capacitados en el diseño, desarrollo y gestión de sistemas de Software de alta calidad. Con un enfoque particular en la calidad y la gestión estratégica de proyectos tecnológicos, el programa también promueve la capacidad de adaptación a los rápidos avances en el sector. De este modo, se garantiza que los alumnos no solo estén preparados para enfrentar los desafíos del futuro, sino que también sean capaces de liderar la innovación en el campo del Software.

Convierte las ideas en soluciones tecnológicas efectivas con el arte de la ingeniería de Software”

Objetivos generales

- Desarrollar habilidades avanzadas en el diseño, desarrollo y mantenimiento de sistemas de Software complejos y escalables, aplicando las mejores prácticas y metodologías de ingeniería del Software
- Capacitar a los estudiantes en el aseguramiento de la calidad del Software, proporcionándoles herramientas y técnicas para garantizar la fiabilidad, seguridad y rendimiento de las soluciones tecnológicas
- Fomentar el liderazgo en la gestión de proyectos tecnológicos, desarrollando competencias en la dirección de equipos multidisciplinarios, la planificación estratégica y la toma de decisiones en entornos dinámicos
- Promover la capacidad de adaptación a los rápidos avances tecnológicos, mediante la especialización en nuevas herramientas, técnicas y tendencias que permiten mantenerse a la vanguardia de la ingeniería del Software
- Desarrollar competencias en la gestión de la calidad en todo el ciclo de vida del Software, desde la planificación inicial hasta el mantenimiento y la mejora continua de los sistemas
- Fortalecer las habilidades de comunicación y trabajo en equipo, esenciales para colaborar eficazmente con diferentes stakeholders, gestionar expectativas y garantizar el éxito de proyectos tecnológicos.

Mejora tus habilidades y que te conviértete en un líder en la creación de soluciones tecnológicas de vanguardia”

Objetivos específicos

Módulo 1. Calidad del Software. Niveles de desarrollo TRL

- Comprender los diferentes niveles de madurez tecnológica y su relación con la calidad del Software
- Evaluar el desarrollo del Software en cada etapa del TRL y cómo impacta en la calidad final del producto

Módulo 2. Desarrollo de proyectos Software. Documentación funcional y técnica

- Desarrollar habilidades para crear documentación funcional y técnica clara y detallada en proyectos de Software
- Analizar la importancia de una documentación precisa para la gestión de proyectos y la calidad del Software

Módulo 3. *Testing* de Software. Automatización de pruebas

- Desarrollar habilidades para diseñar y ejecutar pruebas automatizadas en aplicaciones de Software
- Implementar soluciones de *testing* eficientes utilizando herramientas de automatización de pruebas

Módulo 4. Metodologías de gestión de proyectos Software. Metodologías Waterfall frente a Metodologías Ágiles

- Analizar las diferencias entre las metodologías Waterfall y Ágiles en la gestión de proyectos de Software
- Evaluar los beneficios y limitaciones de cada metodología en función del tipo de proyecto

Módulo 5.TDD (*Test Driven Developement*). Diseño de Software guiado por las pruebas

- Desarrollar habilidades para escribir pruebas unitarias antes de escribir el código de producción
- Mejorar la calidad del Software mediante la implementación de TDD en el proceso de desarrollo

Módulo 6. DevOps. Gestión de Calidad del Software

- Explorar el concepto de DevOps y su impacto en la mejora continua de la calidad del Software
- Aprender a integrar las prácticas de desarrollo y operaciones para lograr un ciclo de vida de Software más ágil y eficiente

Módulo 7. DevOps e integración continua. Soluciones prácticas avanzadas en desarrollo de Software

- Profundizar en las técnicas avanzadas de integración continua dentro del marco de DevOps
- Implementar soluciones prácticas de integración continua para automatizar el proceso de desarrollo y despliegue de Software

Módulo 8. Diseño de Bases de Datos (BD). Normalización y rendimiento. Calidad del Software

- Analizar los principios de diseño de bases de datos, incluyendo normalización y optimización del rendimiento
- Comprender cómo el diseño adecuado de bases de datos contribuye a la calidad del Software

Módulo 9. Diseño de arquitecturas escalables. La arquitectura en el ciclo de vida del Software

- Ahondar en los principios de diseño de arquitecturas escalables y su impacto en la calidad y el rendimiento del Software
- Evaluar diferentes patrones de arquitectura para aplicaciones de Software escalables

Módulo 10. Criterios de Calidad ISO, IEC 9126. Métrica de Calidad del Software

- Comprender los criterios de Calidad del Software según estos estándares y cómo aplicarlos
- Implementar métricas de calidad para evaluar y mejorar continuamente las aplicaciones de Software

Módulo 11. Metodologías, desarrollo y calidad en la Ingeniería de Software

- Profundizar en las metodologías más utilizadas en la Ingeniería de Software y su relación con la Calidad
- Desarrollar un enfoque integral que combine desarrollo, pruebas y calidad en proyectos de Software

Módulo 12. Gestión de proyectos de Software

- Desarrollar habilidades en la gestión de proyectos de Software, desde la planificación hasta la ejecución
- Gestionar los recursos, tiempos y riesgos asociados a los proyectos de desarrollo de Software

Módulo 13. Plataformas de desarrollo del Software

- Explorar las diferentes plataformas de desarrollo de software y sus características
- Evaluar plataformas de desarrollo en función de sus capacidades, flexibilidad y compatibilidad con distintos proyectos

Módulo 14. Computación en el cliente web

- Analizar cómo se realiza la computación en el lado del cliente en el desarrollo de aplicaciones web
- Desarrollar aplicaciones que aprovechen la computación en el cliente para mejorar la interacción y el rendimiento

Módulo 15. Computación en Servidor Web

- Explorar las tecnologías y técnicas utilizadas para la computación en el servidor web
- Comprender el manejo de datos, la lógica de negocio y la gestión de usuarios en el servidor

Módulo 16. Gestión de la seguridad

- Evaluar los riesgos de seguridad en las aplicaciones y aplicar medidas preventivas
- Implementar controles de seguridad en todas las etapas del ciclo de vida del Software

Módulo 17. Seguridad en el Software

- Explorar las mejores prácticas de seguridad en el desarrollo de Software
- Analizar las vulnerabilidades más comunes en el Software y aprender a mitigarlas

Módulo 18. Administración de Servidores Web

- Comprender el rol de los servidores web en el desarrollo y despliegue de aplicaciones
- Desarrollar habilidades en la administración y mantenimiento de servidores web

Módulo 19. Auditoría de Seguridad

- Evaluar la seguridad de los sistemas mediante auditorías y pruebas de penetración
- Implementar procesos de auditoría continua para mejorar la seguridad del Software

Módulo 20. Seguridad en Aplicaciones Online

- Implementar soluciones para proteger aplicaciones online contra amenazas externas e internas
- Establecer políticas de seguridad y auditoría para garantizar la integridad de las aplicaciones online

05

Salidas profesionales

El egresado estará altamente capacitado para asumir roles de liderazgo en el desarrollo, implementación y gestión de Software de alta calidad. Gracias a su especialización avanzada en áreas clave como arquitectura de Software, seguridad en aplicaciones online, gestión de proyectos tecnológicos y metodologías ágiles, podrá dirigir equipos de desarrollo. Además, su preparación le permitirá ocupar cargos de dirección en proyectos de Software, mientras que su capacidad para innovar y liderar equipos multidisciplinarios lo hará capaz de enfrentar los desafíos del entorno digital y contribuir significativamente al éxito de cualquier organización.

TECH te brinda la oportunidad de cumplir tus sueños en la disciplina más apasionante, que convierte ideas en productos tangibles capaces de mejorar la vida de las personas”

Perfil del egresado

El perfil del egresado del Grand Master de Ingeniería y Calidad del Software está orientado a instruir a profesionales altamente cualificados, capaces de liderar y gestionar proyectos tecnológicos de alto impacto. Asegurando la calidad, seguridad y eficiencia en todas las fases del desarrollo de Software, dominará tanto las metodologías ágiles como las tradicionales. Además, estará capacitado para diseñar y desarrollar sistemas de Software escalables, eficientes y seguros, aplicando estándares internacionales de calidad y metodologías avanzadas como DevOps e integración continua.

Conviértete en un experto que garantice ese éxito de las empresas, en la universidad digital más grande del mundo.

- **Seguridad en Software y sistemas:** Competencia en la implementación de prácticas avanzadas de seguridad, incluida la protección de datos y la gestión de vulnerabilidades en aplicaciones online
- **Aseguramiento de la calidad del Software:** Habilidad para aplicar estándares internacionales (ISO, IEC 9126) y herramientas de testing automatizado para garantizar la fiabilidad y rendimiento del Software
- **Desarrollo de arquitecturas escalables:** Capacidad para diseñar y construir sistemas de Software que puedan crecer y adaptarse a las demandas del mercado sin comprometer su calidad ni seguridad
- **Integración continua y DevOps:** Habilidad para implementar y gestionar procesos de integración continua, garantizando una entrega eficiente y sin interrupciones de nuevas funcionalidades de Software

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

1. **Director de Tecnología (CTO):** Responsable de la dirección estratégica de la tecnología en una empresa, liderando equipos de desarrollo y supervisando la implementación de soluciones tecnológicas innovadoras.
2. **Gerente de Calidad del Software:** Encargado de supervisar y asegurar que los procesos y productos de Software cumplan con los estándares de calidad establecidos, liderando las iniciativas de mejora continua y pruebas de Software
3. **Arquitecto de Software:** Diseñador principal de la estructura y la arquitectura de sistemas de Software complejos, asegurando que sean escalables, seguros y eficientes
4. **Líder de Proyectos de Software:** Responsable de la planificación, ejecución y entrega de proyectos de Software, gestionando equipos multidisciplinarios y asegurando que los proyectos se completen dentro del plazo, presupuesto y con los estándares de calidad adecuados
5. **Especialista en Seguridad Informática:** Encargado de la protección de aplicaciones, infraestructuras y datos ante amenazas cibernéticas, implementando estrategias y políticas de seguridad efectivas
6. **Auditor de Seguridad en Software:** Realiza auditorías exhaustivas para identificar vulnerabilidades en las aplicaciones y sistemas, proponiendo mejoras y soluciones para garantizar la seguridad del Software

Si quieres marcar la diferencia en el mundo digital, elige este camino que te especializa como experto en la creación de Software de calidad"

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.

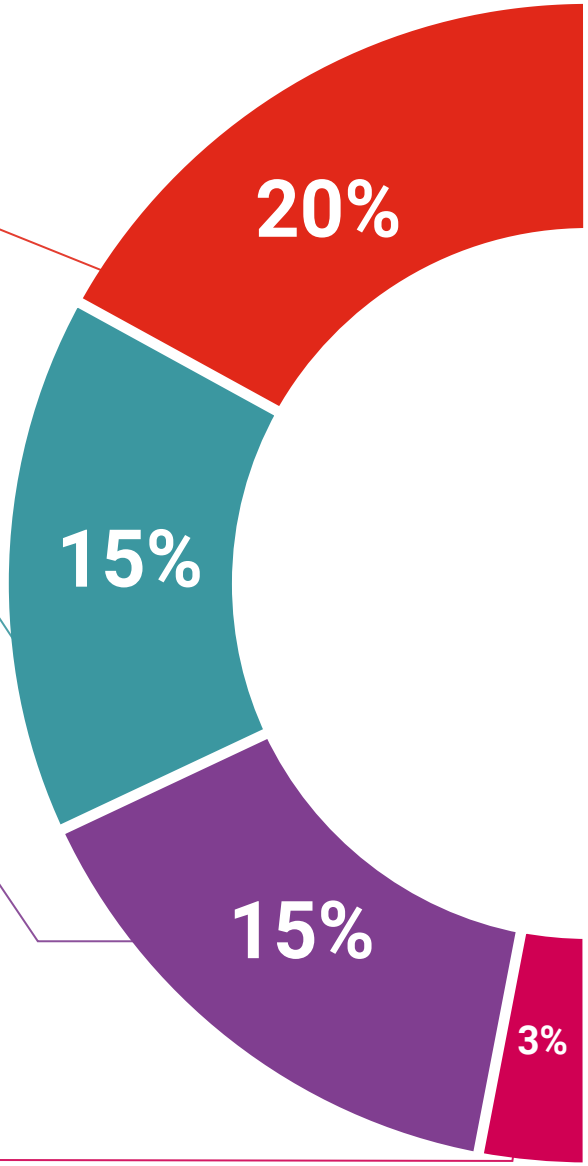
Resúmenes interactivos

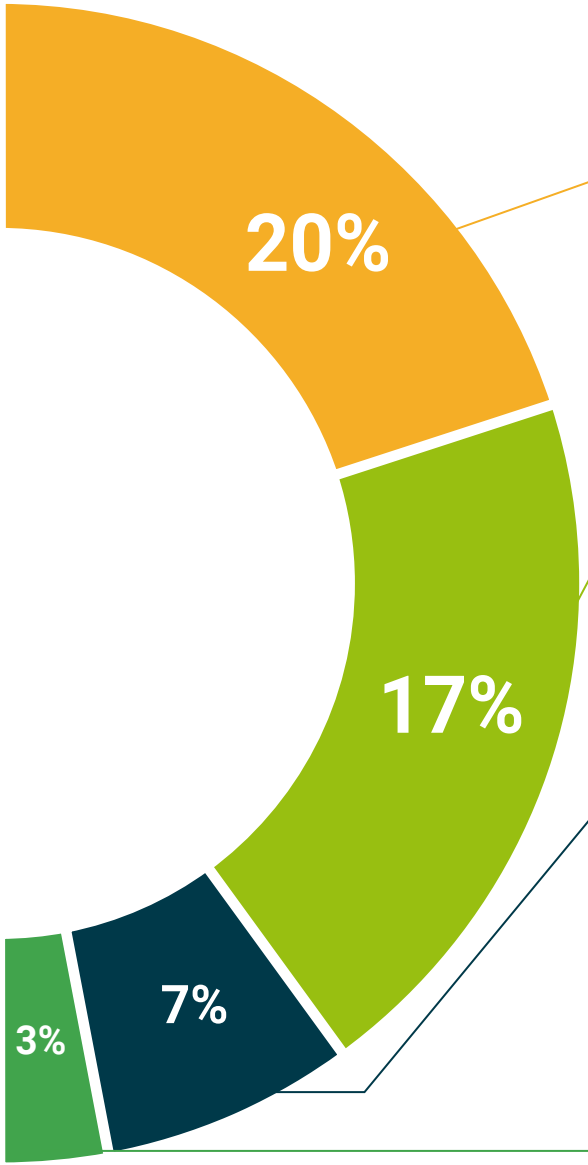
Presentamos los contenidos de manera atractiva y dinámica en píldoras multimedia que incluyen audio, vídeos, imágenes, esquemas y mapas conceptuales con el fin de afianzar el conocimiento.

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.





Case Studies

Completarás una selección de los mejores *case studies* de la materia. Casos presentados, analizados y tutorizados por los mejores especialistas del panorama internacional.

Testing & Retesting

Evaluamos y reevaluamos periódicamente tu conocimiento a lo largo del programa. Lo hacemos sobre 3 de los 4 niveles de la Pirámide de Miller.

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 nuestras futuras decisiones difíciles.

Guías rápidas de actuación

TECH ofrece los contenidos más relevantes del curso en forma de fichas o guías rápidas de actuación. Una manera sintética, práctica y eficaz de ayudar al estudiante a progresar en su aprendizaje.

07

Cuadro docente

La dirección y la docencia de este Grand Master de Ingeniería y Calidad del Software corre a cargo de un equipo de expertos en ingeniería con muchos años de experiencia en la gestión y en el desarrollo de proyectos técnicos y especializados. Su bagaje profesional aporta a esta titulación un plus de calidad que se verá reflejado en una mejor contextualización del contenido por parte del egresado, así como la implementación a la experiencia académica de casos prácticos reales y simulados, pero siempre dirigidos a ofrecer un programa 100% online dinámico, vanguardista y basado en la realidad inmediata del sector.

Únete a la carrera con mayor demanda de expertos en ingeniería y calidad del Software y aprovecha esta oportunidad para convertirte en uno de ellos”

Director Invitado 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"

Dirección

D. Molina Molina, Jerónimo

- Responsable de Inteligencia Artificial en Helphone
- AI Engineer & Software Architect en NASSAT, Internet Satélite en Movimiento
- Consultor Senior en Hexa Ingeniero
- Introductor de Inteligencia Artificial (ML y CV)
- Experto en Soluciones Basadas en Inteligencia Artificial en los campos de Computer Vision, ML/DL y NLP
- Experto Universitario en Creación y Desarrollo de Empresas en Bancaixa y Fundeun
- Ingeniero en Informática por la Universidad de Alicante
- Máster en Inteligencia Artificial por la Universidad Católica de Ávila
- MBA Executive en el Foro Europeo Campus Empresarial

Profesores

Dña. Rodríguez Míguez, Cándida

- Junior Application Developer en Getronics
- Cofundadora y City Leader de la red Galicia AI
- Ingeniera de Software Junior en Indra
- Desarrolladora web en EDISA
- Graduada en Ingeniería Informática por la Universidad de Vigo
- Máster Universitario en Ingeniería Informática por la Universidad de Vigo

D. Pi Morell, Oriol

- Analista Funcional en Fihoca
- Product Owner de Hosting y correo en CDmon
- Analista Funcional y Software Engineer en Atmira y Capgemini
- Docente en Capgemini, Forms Capgemini y en Atmira
- Licenciado en Ingeniería Técnica de Informática de Gestión por la Universidad Autónoma de Barcelona
- Máster en Inteligencia Artificial por la Universidad Católica de Ávila
- MBA en Dirección y Administración de Empresas por la IMF Smart Education
- Máster en Dirección de Sistemas de Información por la IMF Smart Education
- Postgrado en Patrones de Diseño por la Universitat Oberta de Catalunya

D. Martínez Calvo, Francisco Javier

- Ingeniero Técnico Industrial Especializado en Electricidad y Electrónica
- Técnico de Software en HEXA Ingenieros
- Desarrollador Sénior .Net/Arquitecto de soluciones .Net en Everis
- Analista/Arquitecto *Software* en LaLiga
- Ingeniero On-site de Microsoft en BBVA
- Consultor Técnico-Informático *Freelance*
- Formador en Visual Studio, SqlServer, CCNA (Routers y Switch Cisco), Programación Web PHP y .Net en varios centros (Salesianos, Maforem, Dreamsoft)
- Ingeniero Técnico Industrial con Especialización en Electricidad, Electrónica Industrial
- Máster Cibernos en .NET, MCAD
- Máster Eidos en Programación Avanzada, Nivel Experto
- Máster Web con Certificaciones Dreamweaver, Fireworks, Flash y ActionScript, Versiones MX

D. Tenrero Morán, Marcos

- Ingeniero DevOps en Allot Communications
- Manager de Gestión del Ciclo de Vida de las Aplicaciones en Cegid Meta4
- Ingeniero de Automatización QA en Cegid Meta4
- Máster en Desarrollo de Aplicaciones Profesionales para Android por la Universidad Galileo. Guatemala
- Máster en Desarrollo de Servicios en la Nube, Node.Js, JavaScript, HTML5 por la Universidad Politécnica de Madrid
- Desarrollo Web con Angular-CLI (4), Ionic y Node.Js, Meta4 por la Universidad Rey Juan Carlos
- Graduado en Ingeniería de Computadores por la Universidad Rey Juan Carlos

Dña. Acebes Tamargo, Patricia

- Consultora Especializada en Big Data
- Departamento de Operaciones, trabaja con Elasticsearch y Kivana en Sirt
- Investigadora en Línea *Human Factor* y AI *Aplications* en CTIC Centro Tecnológico
- Investigadora en Línea Unidad Negocio en CTIC Centro Tecnológico
- Departamento de Salud Digital y Envejecimiento Activo en CTIC Centro Tecnológico
- Departamento de Ciencia de Datos en CTIC Centro Tecnológico
- Doctor en Informática en Inteligencia Artificial por la Universidad Politécnica de Valencia
- Licenciada en Economía por la Universidad de Oviedo
- Máster en Análisis de Datos UCJC
- Máster de Investigación en Inteligencia Artificial por la UNED
- Máster en *Blockchain, Smart Contracts* y Criptomonedas por la Universidad de Alcalá
- Posgrado en Ingeniería de *Blockchain* por EADA
- Máster Universitario en Economía, Instrumentos, Análisis Económico por la Universidad de Oviedo
- Máster en Fiscalidad por el Colegio de Economistas

08
Titulación

El Grand Master en Ingeniería y Calidad del Software garantiza, además de la capacitación más rigurosa y actualizada, el acceso a un título de Grand Master 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 **Grand Master en Ingeniería y Calidad del Software** 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: **Grand Master en Ingeniería y Calidad del Software**

Modalidad: **online**

Duración: **2 años**

Acreditación: **120 ECTS**

tech global university

D/Dña _____, con documento de identificación _____ ha superado con éxito y obtenido el título de:

Grand Master en Ingeniería y Calidad del Software

Se trata de un título propio de 3.600 horas de duración equivalente a 120 ECTS, con fecha de inicio dd/mm/aaaa y fecha de finalización dd/mm/aaaa.

TECH Global University es una universidad reconocida oficialmente por el Gobierno de Andorra el 31 de enero de 2024, que pertenece al Espacio Europeo de Educación Superior (EEES).

En Andorra la Vella, a 28 de febrero de 2024

Dr. Pedro Navarro Illana
Rector

código único TECH: AFWOR23S techtitute.com/titulos

Grand Master en Ingeniería y Calidad del Software

Distribución General del Plan de Estudios

Curso	Materia	ECTS	Carácter
1º	Calidad del Software. Niveles de desarrollo TRL	6	OB
1º	Desarrollo de proyectos Software. Documentación funcional y técnica	6	OB
1º	*Testing* de Software. Automatización de Pruebas	6	OB
1º	Metodologías de gestión de proyectos Software. Metodologías Waterfall frente a Metodologías Agiles	6	OB
1º	TDD (*Test Driven Developement*). Diseño de Software guiado por las pruebas	6	OB
1º	DevOps. Gestión de Calidad del Software	6	OB
1º	DevOps e integración continua. Soluciones prácticas avanzadas en desarrollo de Software	6	OB
1º	Diseño de Bases de Datos (BD). Normalización y rendimiento. Calidad del Software	6	OB
1º	Diseño de arquitecturas escalables. La arquitectura en el ciclo de vida del Software	6	OB

Curso	Materia	ECTS	Carácter
1º	Criterios de Ccalidad ISO, IEC 9126. Métrica de Calidad del Software	6	OB
2º	Metodologías, desarrollo y calidad en la Ingeniería de Software	6	OB
2º	Gestión de proyectos de Software	6	OB
2º	Plataformas de desarrollo del Software	6	OB
2º	Computación en el cliente web	6	OB
2º	Computación en Servidor Web	6	OB
2º	Gestión de la seguridad	6	OB
2º	Seguridad en el Software	6	OB
2º	Administración de Servidores Web	6	OB
2º	Auditoría de Seguridad	6	OB
2º	Seguridad en Aplicaciones Online	6	OB

Dr. Pedro Navarro Illana
Rector

tech global university

*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.

tech global university

Grand Master
Ingeniería y Calidad del Software

- Modalidad: **online**
- Duración: **2 años**
- Titulación: **TECH Global University**
- Acreditación: **120 ECTS**
- Horario: **a tu ritmo**
- Exámenes: **online**

Grand Master

Ingeniería y Calidad del Software

tech global university