Máster Título Propio
Calidad del Software

tech universidad tecnológica

tech universidad tecnológica

Máster Título Propio
Calidad del Software

- Modalidad: **online**
- Duración: **12 meses**
- Titulación: **TECH Universidad Tecnológica**
- Horario: **a tu ritmo**
- Exámenes: **online**

Acceso web: **www.techtitute.com/informatica/master/master-calidad-software**

Índice

01

Presentación

El vertiginoso crecimiento de la industria y las demandas actuales del mercado, han hecho que exista un elevado nivel de deuda técnica en los proyectos de software. Debido a la imperiosa necesidad de reflejar respuestas rápidas a los requerimientos del cliente o empresa, sin evaluar ni precisar en los detalles propios de la calidad del sistema. Es allí donde se refleja la necesidad de tener en cuenta la escalabilidad del proyecto a lo largo de su ciclo de vida, lo que exige de conocimientos informáticos enfocados en la calidad desde un enfoque *top-down*. Este programa desarrolla los criterios, tareas y metodologías avanzadas para comprender la relevancia de un trabajo orientado a la necesidad de implantar políticas de calidad en las *Software Factories*. Su estudio será completamente online y con una duración de 12 meses, ajustado a la metodología implementada por la mayor universidad digital del mundo.

Especialízate en Calidad del Software desde la perspectiva técnica y de gestión; titulándote en 12 meses, y marca la diferencia en tu entorno profesional"

El concepto de Deuda Técnica aplicándose en la actualidad, por un gran número de corporaciones y administraciones con sus proveedores, refleja la forma improvisada en la que se han desarrollado los proyectos. Generando un nuevo coste implícito al tener que rehacer un proyecto por haber adoptado una solución rápida y sencilla frente a lo que debe ser un planteamiento escalable en la evolución del proyecto.

Y es que de unos años hasta ahora se han desarrollado los proyectos de forma muy rápida, con el objetivo de cerrarlos con el cliente bajo criterios de precio y plazos; en lugar de plantear un enfoque de calidad. Ahora esas decisiones, están pasando factura a muchos proveedores y clientes.

Este Máster Título Propio capacitará al profesional informático para analizar los criterios subyacentes en la Calidad del Software, en todos los niveles. Criterios como la normalización de las bases de datos, el desacoplamiento entre componentes de un sistema de información, las arquitecturas escalables, las métricas, la documentación, tanto funcional como técnica. Además de las metodologías en la gestión y desarrollo de proyectos y otros métodos para asegurar la calidad, como las técnicas de trabajo colaborativo; incluso la denominada *Pair Programming,* que permite que el conocimiento resida en la empresa y no en las personas.

La inmensa mayoría de los másteres de este tipo se centran en una tecnología, en un lenguaje o en una herramienta. Este programa se hace exclusivo en la manera que conciencia al profesional en la importancia de la calidad del software, reducir la deuda técnica de los proyectos con uno de calidad en lugar de un enfoque basado en la economía y los plazos cortos; dota al alumno de conocimientos especializados, de tal modo que pueda justificarse la presupuestación de proyectos.

Para hacer esto posible TECH Universidad Tecnológica ha reunido a un grupo de expertos en el área que transmitirán los conocimientos y experiencias más actualizados. A través de un moderno campus virtual con contenido teórico y práctico, distribuido en diferentes formatos. Serán 10 Módulos divididos en diversos temas y subtemas que harán posible el aprendizaje en 12 meses atendiendo a la metodología *Relearning,* que facilita la memorización y aprendizaje de forma ágil y eficiente.

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

- El desarrollo de casos prácticos presentados por expertos en Desarrollo de Software
- Los contenidos gráficos, esquemáticos y eminentemente prácticos con los que está concebido 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
- Las lecciones teóricas, preguntas al experto y trabajos de reflexión individual
- La disponibilidad de acceso a los contenidos desde cualquier dispositivo fijo o portátil con conexión a internet

El Máster Título Propio en Calidad del Software analiza los criterios subyacentes al tema en todos los niveles. Amplía tu nivel de experiencia. Matricúlate ahora”

Desarrolla los criterios, tareas y metodologías avanzadas para comprender la relevancia de un trabajo orientado a la calidad, y proveer soluciones efectivas a tu empresa o cliente"

Un programa centrado en la concienciación sobre la importancia de la calidad del software y la necesidad de implantar políticas de calidad en los software Factories.

Aprende de una manera práctica y flexible. Compartiendo tu día a día con esta capacitación 100% online exclusiva de TECH Universidad Tecnológica.

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

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

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

02

Objetivos

El Máster Título Propio en Calidad del Software proporciona al alumno una visión clara y especializada de la importancia de la calidad en los procesos de desarrollo de software. Así como de las herramientas más avanzadas para implantar procesos de DevOps y de sistemas para el aseguramiento de la calidad. En definitiva, brindará un amplio y especializado conocimiento teórico-práctico para que entiendan el desarrollo de proyectos desde una perspectiva moderna y eficiente.

Podrás acceder fácilmente a todos los contenidos cada vez que desees. Desde tu ordenador o dispositivo favorito. Además de descargarlos para tu próxima consulta”

Objetivos generales

- Desarrollar los criterios, tareas y metodologías avanzadas para comprender la relevancia de un trabajo orientado a la calidad
- Analizar los factores clave en la calidad de un proyecto software
- Desarrollar los aspectos normativos relevantes
- Implantar procesos de DevOps y de sistemas para el aseguramiento de la calidad
- Reducir la deuda técnica de los proyectos con un enfoque de calidad en lugar de un enfoque basado en la economía y los plazos cortos
- Dotar al alumno de conocimientos especializados para poder medir y cuantificar la calidad de un proyecto software
- Defender las propuestas económicas de proyectos desde la base de la calidad

Objetivos específicos

Módulo 1. Calidad del Software. Niveles de desarrollo TRL

- Desarrollar de forma clara y concisa los elementos que engloban la calidad del software
- Aplicar los modelos y estándares en función de sistema, producto y proceso software
- Profundizar en las normas ISO de calidad aplicadas tanto de forma general como en partes específicas
- Aplicar las normas en función del ámbito del entorno (local, nacional, internacional)
- Examinar los niveles de madurez TRL y adaptarlos a las diferentes partes del proyecto software a tratar
- Adquirir capacidad de abstracción para aplicar uno o varios criterios de elementos y niveles de la calidad del software
- Distinguir los casos de aplicación de las normativas y niveles de madurez en un proyecto simulado de caso real

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

- Determinar la influencia de la gestión de proyectos en la calidad
- Desarrollar las diferentes fases de un proyecto
- Diferenciar los conceptos de calidad inherentes a la documentación funcional y técnica
- Analizar la fase de toma requisitos, la fase de análisis la gestión del equipo y la fase de construcción
- Establecer las diferentes metodologías de gestión de proyectos software
- Generar criterio para decidir cuál es la metodología más adecuada en función del tipo de proyecto

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

- Establecer las diferencias entre calidad de producto, de proceso y calidad de uso
- Conocer la normativa ISO/IEC 15504
- Determinar los detalles de CMMI
- Aprender las claves de la integración continua, los repositorios y las repercusiones que tienen en un equipo de desarrollo de software
- Establecer la relevancia de incorporar repositorios por proyectos de software. Aprender a crearlos con TFS
- Asimilar la importancia de la escalabilidad del software en diseño y desarrollo de sistemas de información

Módulo 4. Metodologías de Gestión de Proyectos Software. Metodologías *Waterfall* frente a metodologías ágiles

- Determinar en qué consiste la metodología Waterfall
- Profundizar en la Metodología Scrum
- Establecer las Diferencias entre Waterfall y Scrum
- Concretar las diferencias entre las metodologías Waterfall y Scrum y cómo lo ve el cliente
- Examinar el Panel Kanban
- Plantear un mismo proyecto con Waterfall y Scrum
- Montar un proyecto híbrido

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

- Conocer la aplicación práctica del TDD y sus posibilidades, para la realización de pruebas de un proyecto software en un futuro
- Completar casos de simulación real propuestos, como aprendizaje continuo de este concepto TDD
- Analizar, en los casos de simulación, hasta qué punto pueden acertar o fallar las pruebas, desde un punto de vista constructivo
- Determinar las alternativas a TDD, realizando un análisis comparativo entre ellas

Módulo 6. DevOps. Gestión de Calidad del Software

- Analizar las deficiencias de un proceso tradicional
- Evaluar las posibles soluciones y elegir la más idónea
- Comprender las necesidades de negocio y sus impactos en la implantación
- Evaluar los costes de las mejoras a implementar
- Desarrollar un ciclo de vida de software evolucionable, adaptado a las necesidades reales
- Anticipar posibles errores y evitarlos desde el proceso de diseño
- Fundamentar el uso de los distintos modelos de implantación

Módulo 7. DevOps e Integración Continua. Soluciones prácticas avanzadas en Desarrollo de Software

- Identificar las etapas del ciclo de desarrollo y entrega de software adaptados a los casos particulares
- Diseñar un proceso de entrega de software mediante integración continua
- Construir e implementar integración y despliegue continuo basado en su diseño previo
- Establecer puntos de control de calidad automáticos en cada entrega de software
- Mantener un proceso de entrega de software automático y robusto
- Adaptar las necesidades futuras al proceso de integración y despliegue continuo
- Analizar y anticipar vulnerabilidades de seguridad durante el proceso de entrega de software y tras su entrega

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

- Valorar el uso del modelo entidad-relación para el diseño previo de una base de datos
- Aplicar una entidad, un atributo, una clave, etc, para la mejor integridad de los datos
- Evaluar las dependencias, formas y reglas de la normalización de bases de datos
- Especializarse en el funcionamiento de un sistema de almacén de datos OLAP, elaborando y usando tanto la tabla de hechos, como la tabla de dimensiones
- Determinar los puntos clave para el rendimiento de la base de datos
- Completar casos de simulación real propuestos, como aprendizaje continuo de diseño, normalización y rendimiento de la base de datos
- Establecer en los casos de simulación, las opciones a resolver en la creación de la base de datos desde un punto de vista constructivo

Módulo 9. Diseño de Arquitecturas Escalables. La Arquitectura en el Ciclo de Vida del Software

- Desarrollar el concepto de arquitectura del software y sus características
- Determinar los diferentes tipos de escalabilidad en la arquitectura del software
- Analizar los diferentes niveles que pueden darse en una escalabilidad Web
- Adquirir conocimiento especializado sobre el concepto de ciclo de vida del software, etapas y modelos
- Determinar el impacto de una arquitectura en el ciclo de vida de software, con sus ventajas, limitaciones y herramientas de ayuda
- Completar casos de simulación real propuestos, como aprendizaje continuo de la arquitectura y ciclo de vida del software
- Valorar, en los casos de simulación, hasta qué punto pueden dar factible o innecesario el diseño de la arquitectura

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

- Desarrollar el concepto de criterios de calidad y aspectos relevantes
- Examinar la norma ISO/IEC 9126, aspectos principales e indicadores
- Analizar las diferentes mediciones para que un proyecto software cumpla las evaluaciones acordadas
- Examinar los atributos internos y externos a tratar en la calidad de un proyecto software
- Distinguir las métricas en función del tipo de programación (estructurado, orientación a objetos, por capas, etc.)
- Completar casos de simulación real, como aprendizaje continuo de la medición de la calidad
- Ver en los casos de simulación hasta qué punto es factible o innecesario; es decir, desde un punto de vista constructivo de las autoras

Destaca tu perfil profesional con esta capacitación exclusiva. Titúlate en 12 meses y de forma práctica con la metodología que solo te brinda TECH Universidad Tecnológica"

03

Competencias

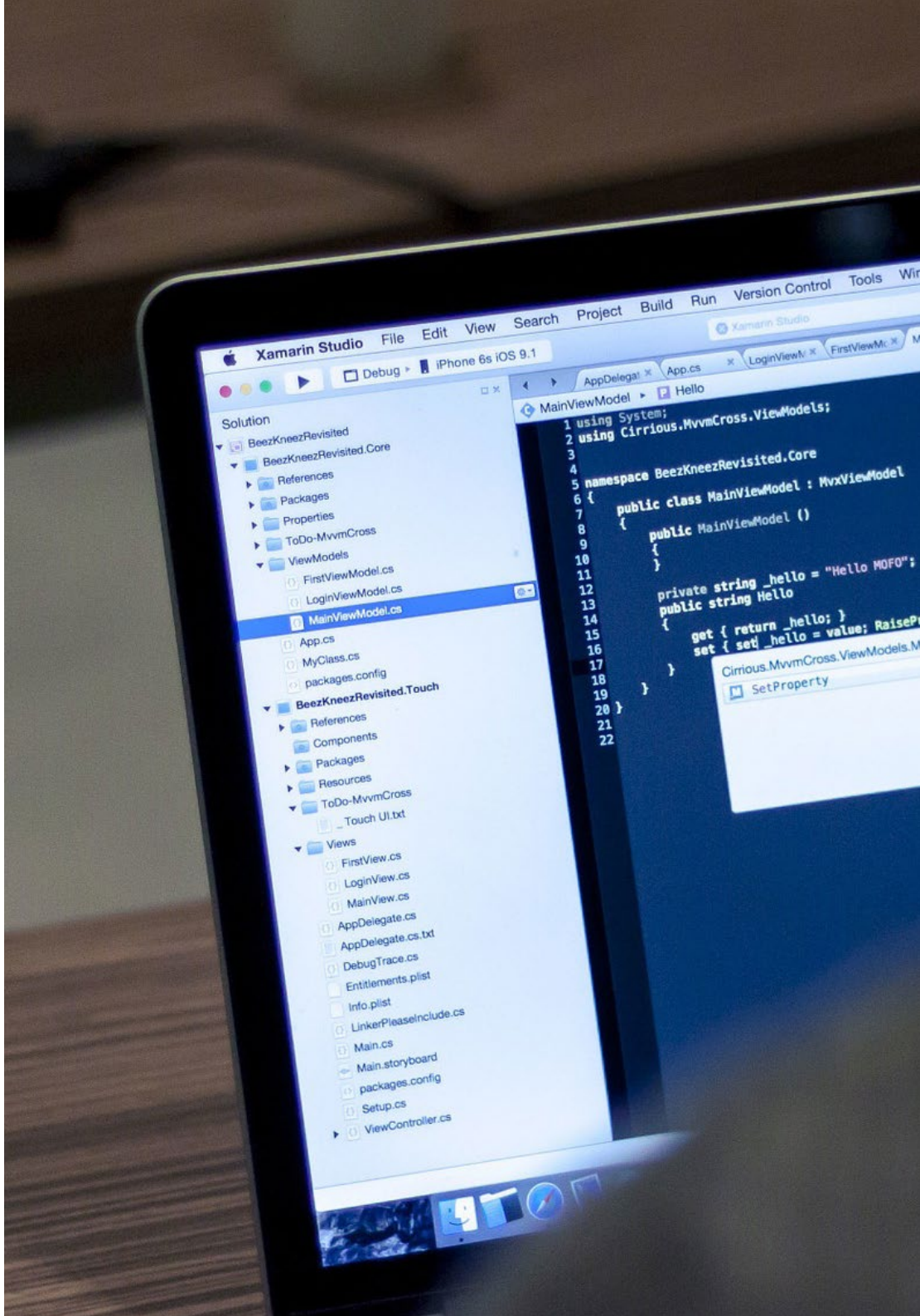
El egresado de este Máster Título Propio en Calidad del Software dominará el tema desde la perspectiva tanto técnica como de gestión. Pudiendo desarrollar el planteamiento de un proyecto, así como la ejecución del mismo y elaborar una arquitectura sostenible, eficaz y de calidad en los proyectos de software que se le presente. Para ello el personal docente ha volcado toda su experiencia personal en la elaboración de multitud de casos prácticos, que servirán de contextualización y evolución ante esa “deuda técnica” que ya, no se hará presente.

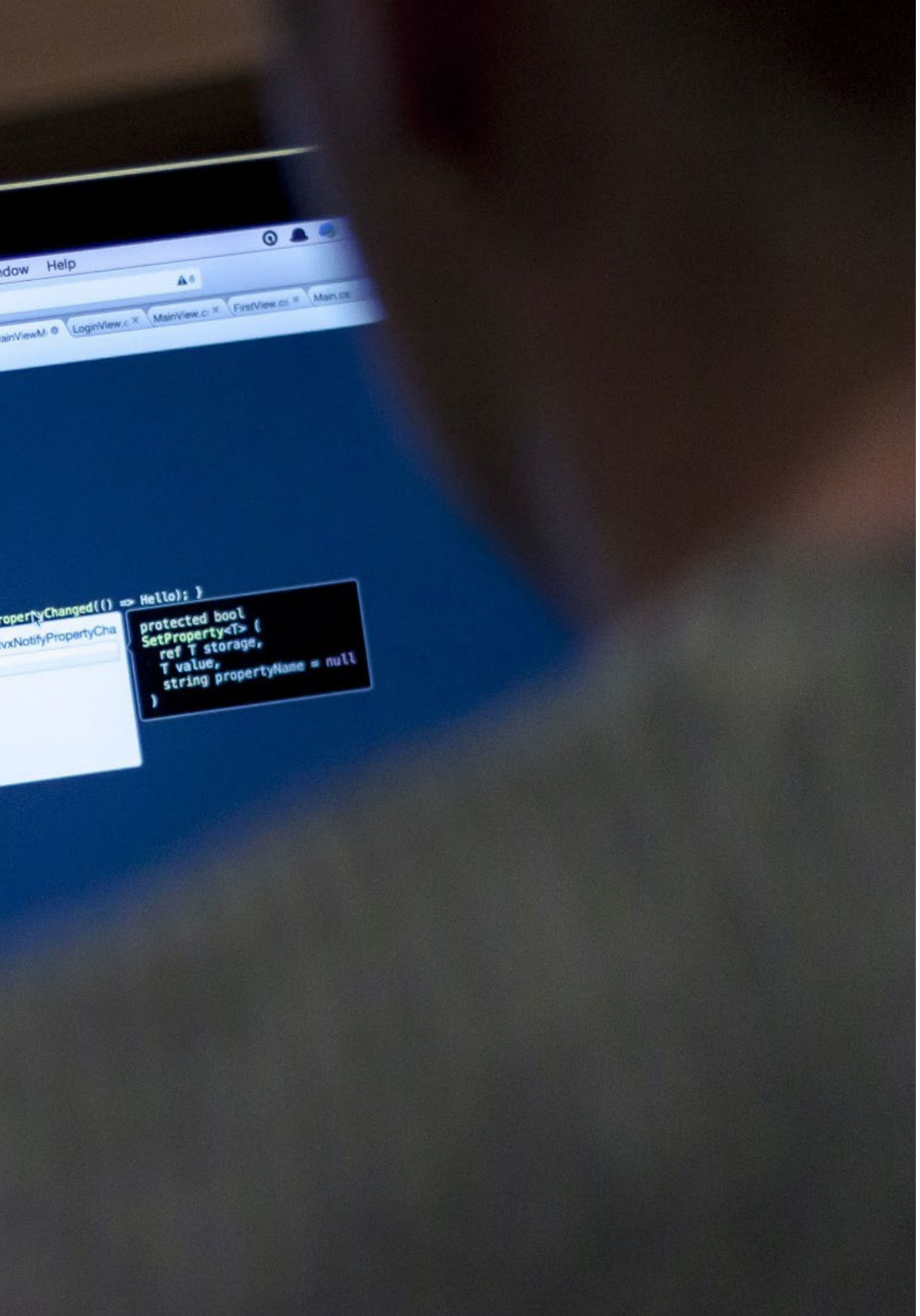
Un profesional informático enfocado en la calidad es un valor en alza para las consultoras de software y las grandes corporaciones. Matricúlate ahora en este Máster Título Propio en calidad del Software"

Competencias generales

- Reducir la deuda técnica de los proyectos con un enfoque de calidad en lugar de un enfoque basado en la economía y los plazos cortos
- Medir y cuantificar la calidad de un proyecto software
- Realizar el TDD de forma correcta, para que elevar los estándares de calidad del software
- Justificar la presupuestación de proyectos orientados a la calidad
- Desarrollar las normas, modelos y estándares de la calidad
- Examina las diferentes evaluaciones de madurez en la tecnología
- Reducir riesgo y asegurar el mantenimiento y control de las versiones posteriores
- Dominar las fases en las que se descompone un proyecto

Potencia tus habilidades y descubre las infinitas posibilidades de crecimiento profesional que se abren ante esta nueva experiencia"





Competencias específicas

- Evaluar un sistema software en cuanto al grado de avance en el proceso del proyecto
- Abordar estos puntos de fiabilidad, métrica y garantía en los proyectos software de manera correcta y estratégica
- Abordar el proceso de decisión de la metodología a utilizar en el proyecto
- Dominar los aspectos normativos imprescindibles para la creación de software
- Desarrollar el *Testing* de forma automática
- Establecer una comunicación adecuada con el cliente, entendiendo la forma en qué percibe el proyecto según la metodología aplicada
- Elaborar la lista de requisitos de las pruebas
- Realizar la abstracción, división en pruebas más unitarias y eliminar lo que no aplique a la buena realización de las pruebas del proyecto software a realizar
- Actualizar la lista de requisitos de las pruebas de forma mesurada y correcta
- Adaptar la cultura DevOps a las necesidades de negocio
- Desarrollar las últimas prácticas y herramientas en la integración y despliegue continuo
- Refactorizar y afrontar la gestión y coordinación de los datos

04

Dirección del curso

Expertos docentes con un dilatado currículum en el área de soluciones informáticas y desarrollo de software e investigación, dirigen este Máster Título Propio para brindar las herramientas y conocimientos necesarios al futuro egresado enfocado en calidad en los procesos de desarrollo de software y de las herramientas más avanzadas para implantar procesos de DevOps y de sistemas para el aseguramiento de la calidad. Este equipo de profesionales guiará en todo momento al alumno, para la consecución de los objetivos de forma remota por ser un programa netamente online y siguiendo la metodología más vanguardista implementada por TECH.

Docentes especializados están comprometidos en brindarte el mejor contenido y hacer de tu proceso de aprendizaje, una experiencia ágil y dinámica. Aclarando tus dudas y acompañándote en todo el recorrido”

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

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. Martínez Cerrato, Yésica

- Experta en Analítica de Negocio y Gestión de los Sistemas de Información
- Product Manager en Seguridad Electrónica en Securitas Direct
- Gestora de Proyectos del Área de Integración de Grandes Cuentas en Correos
- Analista de Inteligencia Empresarial en Ricopia Technologies
- Docente en estudios universitarios y postuniversitarios
- Graduada en Ingeniería de Telecomunicaciones por la Universidad de Alcalá

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

Dr. Peralta Martín-Palomino, Arturo

- CEO y CTO en Prometeus Global Solutions
- CTO en Korporate Technologies
- CTO en AI Shepherds GmbH
- Consultor y Asesor Estratégico Empresarial en Alliance Medical
- Director de Diseño y Desarrollo en DocPath
- Doctor en Ingeniería Informática por la Universidad de Castilla-La Mancha
- Doctor en Economía, Empresas y Finanzas por la Universidad Camilo José Cela
- Doctor en Psicología por la Universidad de Castilla-La Mancha
- Máster en Executive MBA por la Universidad Isabel I
- Máster en Dirección Comercial y Marketing por la Universidad Isabel I
- Máster Experto en Big Data por Formación Hadoop
- Máster en Tecnologías Informáticas Avanzadas por la Universidad de Castilla-La Mancha
- Miembro: Grupo de Investigación SMILE

D. Soto Jiménez, Manuel

- Lynx Financial Crime Tech en el grupo Santander
- Grado en Ingeniería Informática por la Universidad Autónoma de Madrid
- Grado en Matemáticas por la Universidad Autónoma de Madrid
- Curso en Quantum 101: Quantum Computing & Quantum Internet Professional Certificate por la Universidad Técnica de Delft
- Curso en Deep Learning con TensorFlow por IBM
- Lenguajes de programación: Python, R, C, SQL, MongoDB, Matlab, Sage, Cypher, VHDL, Prolog, Javascript, CSS. Lenguajes de Marcado: Markdown, HTML, Latex

Nuestro equipo docente te brindará todos sus conocimientos para que estés al día de la información más actualizada en la materia"

05

Estructura y contenido

La estructura y contenidos de este Máster Título Propio han sido desarrollados para abarcar los temas más importantes para el desarrollo de un Software con Calidad. Compuesto por 10 módulos de enseñanza, que van desde desarrollo de proyectos software, la documentación funcional y técnica, el *test Driven Developement* y las diferentes metodologías, hasta la implementación de soluciones prácticas avanzadas con DevOps e integración continua, todo fundamentado en alcanzar la calidad del software. El amplio contenido multimedia, seleccionado con rigor por los docentes expertos, será de gran apoyo para aliviar la carga lectiva y servir de material de referencia para futuras consultas.

Los casos prácticos, basados en la realidad, te servirán para afianzar y contextualizar toda la teoría aprendida durante el programa"

Módulo 1. Calidad del Software. Niveles de desarrollo TRL

1.1. Elementos que influyen en la calidad del 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 del 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 de proyectos´
- 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
- 2.6.2. Riesgos en el proyecto
- 2.6.3. Gestión del riesgo

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 stress 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 stress
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 del 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 del 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 secretos

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
- 7.6.3. Despliegue continuo de aplicaciones móviles

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 de 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 del proyecto real para diseño BD (I)
8.8.1. Descripción general del proyecto (Empresa A)
8.8.2. Aplicación del diseño de bases 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). Beneficios, 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 el diseño de la arquitectura del 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 el diseño de la arquitectura del 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
- 10.2.7. Seguridad

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*

Accederás a un contenido único y especializado. Seleccionado por docentes expertos para una titulación que hará trascender tu perfil profesional”

06

Metodología

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

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

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

Estudio de Caso para contextualizar todo el contenido

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

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

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

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

Un método de aprendizaje innovador y diferente

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

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

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

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

Relearning Methodology

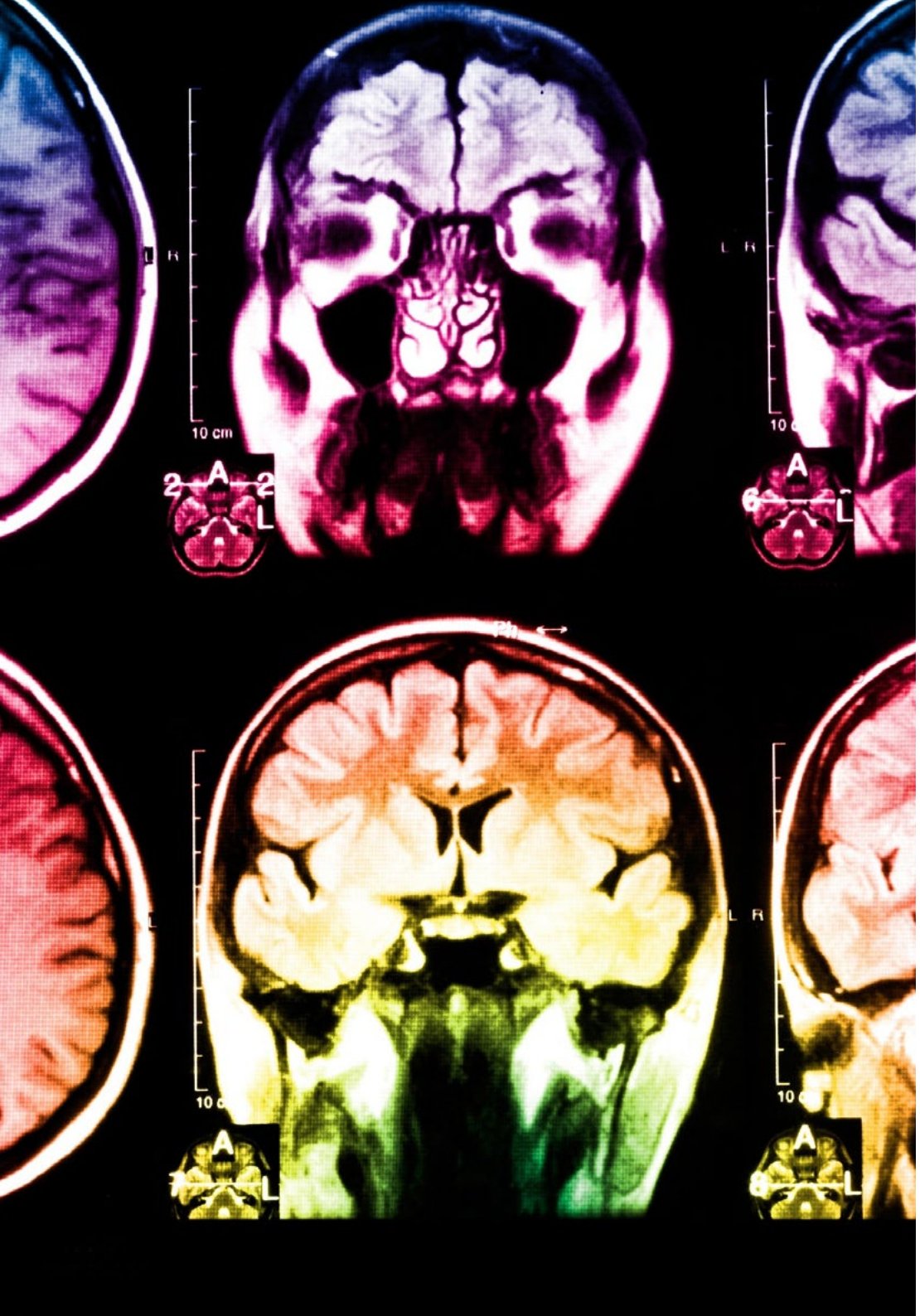
TECH aúna de forma eficaz la metodología del Estudio de Caso con un sistema de aprendizaje 100% online basado en la reiteración, que combina elementos didácticos diferentes en cada lección.

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

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

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

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



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

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

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

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

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

Material de estudio

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

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

Clases magistrales

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

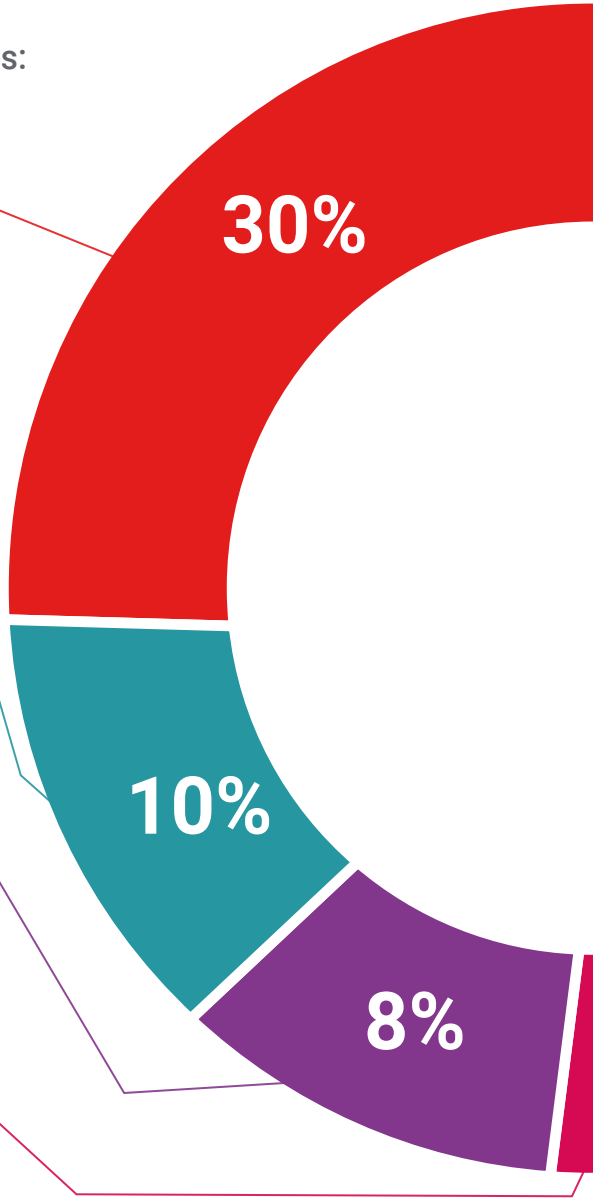
El denominado Learning from an Expert afianza el conocimiento y el recuerdo, y genera seguridad en las futuras decisiones difíciles.

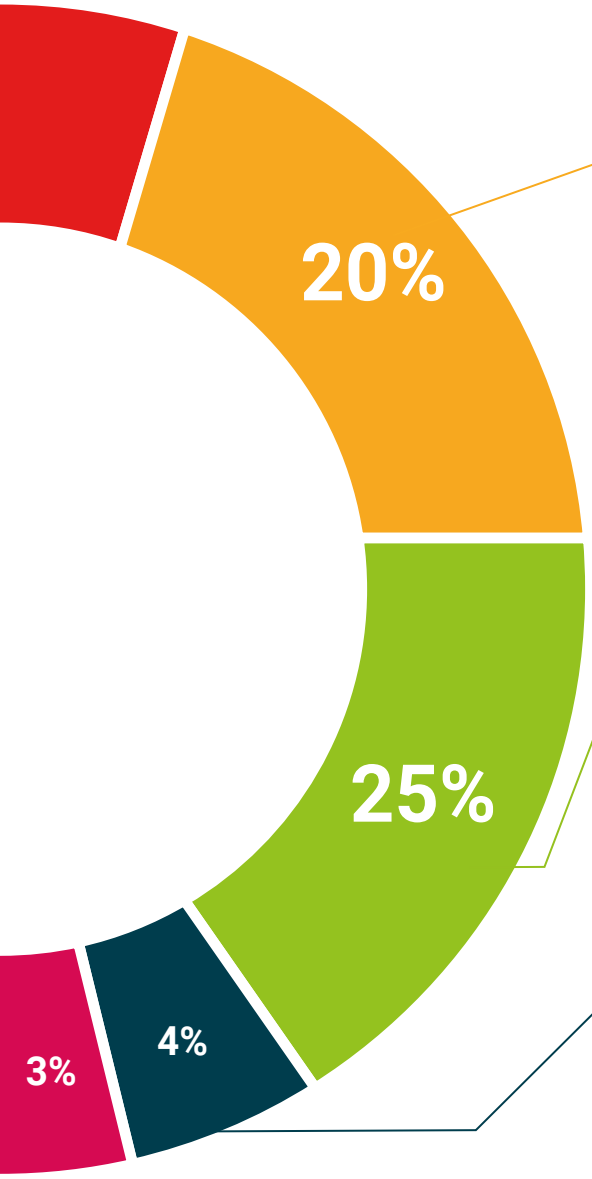
Prácticas de habilidades y competencias

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

Lecturas complementarias

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





Case studies

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

Resúmenes interactivos

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

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

Testing & Retesting

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

07

Titulación

El Máster Título Propio en Calidad del Software garantiza, además de la capacitación más rigurosa y actualizada, el acceso a un título de Máster Propio expedido por TECH Universidad Tecnológica.

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

Este **Máster Título Propio en Calidad del Software** contiene el programa más completo y actualizado del mercado.

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

El título expedido por **TECH Universidad Tecnológica** expresará la calificación que haya obtenido en el Máster Título Propio y reunirá los requisitos comúnmente exigidos por las bolsas de trabajo, oposiciones y comités evaluadores de carreras profesionales.

Título: **Máster Título Propio en Calidad del Software**

N.º Horas Oficiales: **1.500 h.**


tech universidad tecnológica

Otorga la presente

CONSTANCIA

a

C. _____, con documento de identificación nº_____

Por haber superado con éxito y acreditado el programa de

MÁSTER TÍTULO PROPIO

en

Calidad del Software

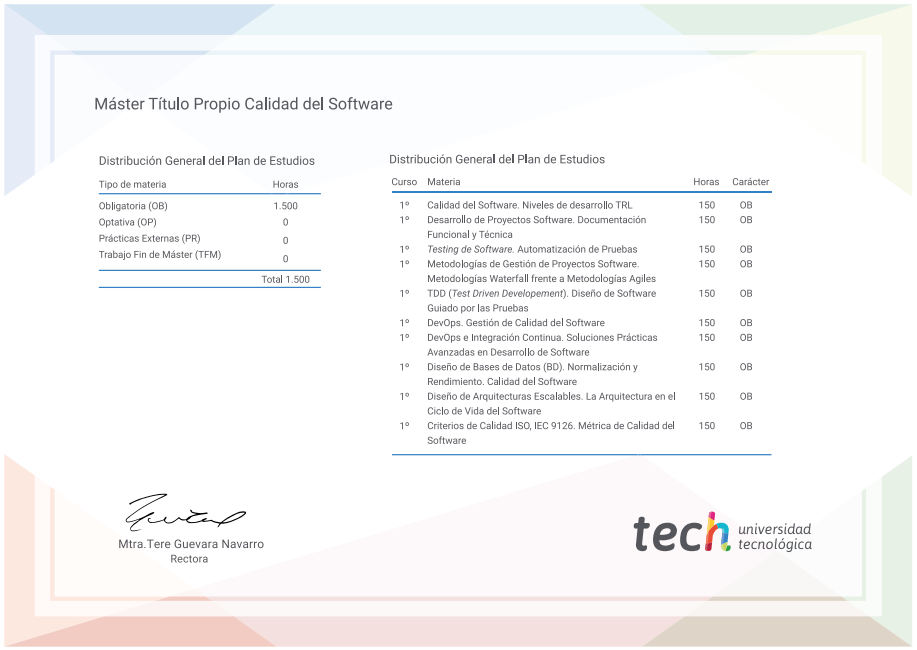
Se trata de un título propio de esta Universidad con una duración de 1.500 horas, con fecha de inicio dd/mm/aaaa y fecha de finalización dd/mm/aaaa.

TECH es una Institución Particular de Educación Superior reconocida por la Secretaría de Educación Pública a partir del 28 de junio de 2018.

A 17 de junio de 2020

Mtra.Tere Guevara Navarro
Rectora

Este título propio se deberá acompañar siempre del título universitario habilitante expedido por la autoridad competente para ejercer profesionalmente en cada país. código único TECH: AFWOR23S techtitute.com/titulos


Máster Título Propio Calidad del Software

Distribución General del Plan de Estudios

Tipo de materia	Horas
Obligatoria (OB)	1.500
Optativa (OP)	0
Prácticas Externas (PR)	0
Trabajo Fin de Máster (TFM)	0
	Total 1.500

Distribución General del Plan de Estudios

Curso	Materia	Horas	Carácter
1º	Calidad del Software. Niveles de desarrollo TRL	150	OB
1º	Desarrollo de Proyectos Software. Documentación Funcional y Técnica	150	OB
1º	*Testing de Software*. Automatización de Pruebas	150	OB
1º	Metodologías de Gestión de Proyectos Software. Metodologías Waterfall frente a Metodologías Agiles	150	OB
1º	TDD (*Test Driven Developement*). Diseño de Software Guiado por las Pruebas	150	OB
1º	DevOps. Gestión de Calidad del Software	150	OB
1º	DevOps e Integración Continua. Soluciones Prácticas Avanzadas en Desarrollo de Software	150	OB
1º	Diseño de Bases de Datos (BD). Normalización y Rendimiento. Calidad del Software	150	OB
1º	Diseño de Arquitecturas Escalables. La Arquitectura en el Ciclo de Vida del Software	150	OB
1º	Criterios de Calidad ISO, IEC 9126. Métrica de Calidad del Software	150	OB

Mtra.Tere Guevara Navarro
Rectora

tech universidad tecnológica

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

tech universidad tecnológica

Máster Título Propio

Calidad del Software

- Modalidad: **online**
- Duración: **12 meses**
- Titulación: **TECH Universidad Tecnológica**
- Horario: **a tu ritmo**
- Exámenes: **online**

Máster Título Propio
Calidad del Software

tech universidad tecnológica