

Máster Título Propio

Calidad del Software





Máster Título Propio Calidad del Software

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

Acceso web: www.techtute.com/informatica/master/master-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. 24

05

Salidas profesionales

pág. 30

06

Licencias de software incluidas

pág. 34

07

Metodología de estudio

pág. 38

08

Cuadro docente

pág. 48

09

Titulación

pág. 56

01

Presentación del programa

La Calidad del *Software* se ha consolidado como un factor estratégico en el desarrollo de soluciones informáticas modernas. Más allá de la funcionalidad, los sistemas deben ser fiables, eficientes, mantenibles y seguros. En este contexto, surge la necesidad de definir métricas objetivas que permitan evaluar de forma sistemática el grado de calidad alcanzado en cada proyecto. Con esta idea en mente, TECH ha creado una pionera titulación universitaria centrada en los modelos más innovadores para garantizar la Calidad del *Software* a lo largo de todo su ciclo de vida. A su vez, se basa en una cómoda modalidad 100% online que se adapta a la agenda de profesionales ocupados.





“

Gracias a este Máster Título Propio 100% online, diseñarás estructuras de Software altamente orientadas a la Calidad, el mantenimiento y la escalabilidad”

El concepto de Calidad del *Software* ha evolucionado desde un enfoque meramente funcional hacia una visión holística del producto. Actualmente, se entiende como un conjunto de atributos que determinan su capacidad para satisfacer necesidades explícitas e implícitas bajo condiciones específicas. Frente a esto, los especialistas necesitan disponer de un conocimiento integral relativo a las metodologías más modernas para evaluar el funcionamiento de las arquitecturas distribuidas y entornos DevOps.

En este contexto, TECH presenta un vanguardista Máster Título Propio en Calidad del Software. El plan de estudios ahondará en cuestiones como la normalización de bases de datos y el desacoplamiento de componentes. Asimismo, el temario brindará al alumnado las claves para diseñar de arquitecturas escalables. También, los materiales didácticos profundizarán en el uso de métricas para evaluar la Calidad de las soluciones. Gracias a esto, los egresados desarrollarán una visión integral de los procesos de aseguramiento de Calidad, dominando desde la planificación de pruebas automatizadas hasta la implementación de estándares internacionales.

Para afianzar todos estos contenidos, TECH emplea el vanguardista método del *Relearning*, consistente en la reiteración progresiva de los conceptos clave para su correcta asimilación. Además, la titulación universitaria proporciona a los profesionales una diversidad de casos prácticos reales, permitiendo así que los alumnos se ejerciten en entornos simulados para acercarlos a la realidad de la praxis informático. En este sentido, para acceder a los recursos educativos los egresados solo requerirán un dispositivo electrónico capaz de conectarse a internet. Además, el programa contará con la colaboración de un distinguido Director Invitado Internacional, quien impartirá 10 exhaustivas *Masterclasses*.

Este **Máster Título Propio en Calidad del Software** contiene el programa universitario 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



Un prestigioso Director Invitado Internacional ofrecerá 10 intensivas Masterclasses relativas a las últimas tendencias en Calidad del Software"

“

Un plan de estudio cimentado en la revolucionaria metodología del Relearning, que te facilitará afianzar los conceptos complejos con eficiencia e inmediatez”

Incluye en su cuadro docente a profesionales pertenecientes al ámbito del desarrollo de *Software*, que vierten en este programa la experiencia de su trabajo, además de reconocidos especialistas de sociedades de referencia y universidades de prestigio.

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

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

Profundizarás en el uso de herramientas modernas de automatización para efectuar pruebas funcionales, de rendimiento y regresión.

Comprenderás los fundamentos de la Calidad del Software en el desarrollo de sistemas informáticos avanzados.



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 diez 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 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

Este plan de estudios aborda, desde una óptica crítica e interdisciplinaria, el papel de los organismos internacionales, la evolución de la regulación digital, la gobernanza de datos y el impacto de tecnologías como la inteligencia artificial, *blockchain* y la ciberseguridad. A lo largo del temario, se articula el análisis geopolítico y tecnológico con un enfoque aplicado al desarrollo y gestión de *Software*, preparando a los profesionales para intervenir en sistemas complejos y entornos digitales interconectados.



“

*Analizarás las técnicas más vanguardistas
para garantizar la seguridad y la confiabilidad
del Software”*

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 ágiles

- 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 development*). 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*



Gestionarás proyectos de Software con un enfoque basado en la mejora continua y la entrega de valor sostenible”

04

Objetivos docentes

El programa tiene como objetivo principal proporcionar una comprensión profunda e integrada del entorno global desde la perspectiva del desarrollo de *Software* y las tecnologías de la información. Para ello, se busca dotar al estudiante de las herramientas conceptuales y prácticas necesarias para analizar cómo los marcos regulatorios, las crisis tecnológicas y los organismos internacionales influyen en la innovación digital. Además, se pretende fomentar una visión crítica sobre los desafíos éticos, la seguridad informática y la evolución de la gobernanza digital. Así, se impulsa la capacidad de diseñar, adaptar y gestionar soluciones tecnológicas en escenarios cada vez más complejos.



“

Aplica metodologías ágiles y herramientas de DevOps para optimizar procesos y acelerar ciclos de desarrollo en entornos reales”



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*



Integra conocimientos de ciberseguridad, testing y control de calidad para garantizar la solidez y fiabilidad de tus desarrollos”





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*

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*

Módulo 5. TDD (*test driven development*). 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

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

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 de 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

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*

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...)
- ♦ Completar casos de simulación real, como aprendizaje continuo de la medición de la calidad

05

Salidas profesionales

Este programa impulsa perfiles preparados para irrumpir con fuerza en el sector del *Software* y la tecnología global. Más allá de los roles tradicionales, habilita caminos hacia posiciones emergentes como estratega en innovación digital, analista de entornos tecnológicos complejos o coordinador de proyectos transversales en entornos internacionales. Asimismo, el dominio de marcos regulatorios y herramientas digitales abre la puerta a colaborar con instituciones o *startups* tecnológicas. Así, quienes egresan no solo se insertan en el mercado: lo redefinen, aportando una visión crítica, global y actualizada del desarrollo tecnológico.



“

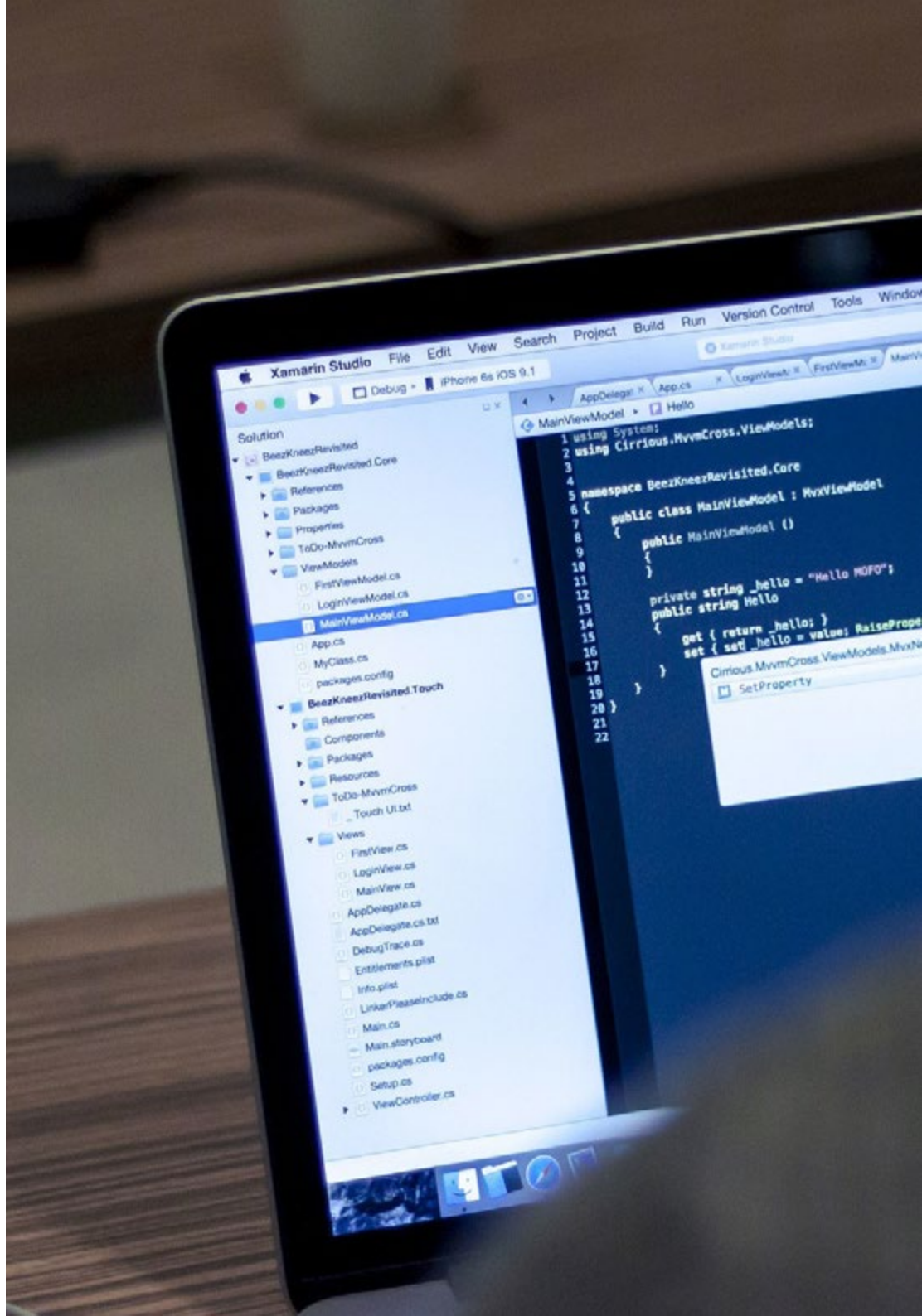
Especialízate en arquitecturas de microservicios y contenedores para liderar proyectos escalables y eficientes en la nube”

Perfil del egresado

Al finalizar este programa, el egresado se distingue por su capacidad para integrar conocimiento técnico, pensamiento analítico y visión estratégica en el ámbito del Software. Domina lenguajes, herramientas y entornos de desarrollo, pero también sabe interpretar marcos regulatorios, liderar equipos multidisciplinarios y tomar decisiones en contextos de cambio tecnológico acelerado. Gracias a una educación que combina teoría de vanguardia con aplicación práctica, adquiere habilidades para diseñar soluciones innovadoras, optimizar procesos digitales y adaptarse con agilidad a los desafíos del ecosistema tecnológico global. No solo resuelve problemas: anticipa tendencias y genera impacto.

¿Quieres ejercitarte como Ingeniero de Automatización de Pruebas? Lógralo con esta titulación universitaria en solamente meses.

- ♦ **Pensamiento crítico y analítico:** Capacidad para evaluar información compleja, identificar patrones y tomar decisiones fundamentadas en entornos tecnológicos dinámicos
- ♦ **Resolución de problemas:** Habilidad para abordar desafíos desde múltiples perspectivas, proponiendo soluciones innovadoras y sostenibles
- ♦ **Trabajo colaborativo:** Aptitud para integrarse en equipos multidisciplinarios, comunicarse de forma efectiva y contribuir al logro de objetivos comunes
- ♦ **Aprendizaje autónomo y continuo:** Disposición para actualizar conocimientos y adaptarse a nuevas herramientas, lenguajes y metodologías a lo largo de la carrera profesional





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

- 1. Ingeniero de Software:** Diseña, desarrolla y mantiene aplicaciones y sistemas informáticos eficientes y escalables.
- 2. Arquitecto de Software:** Define la estructura técnica de sistemas complejos, asegurando su rendimiento, seguridad y mantenibilidad.
- 3. Desarrollador Backend:** Crea la lógica de funcionamiento de aplicaciones, gestionando bases de datos, servidores y APIs.
- 4. Desarrollador Frontend:** Diseña e implementa interfaces interactivas, garantizando una experiencia de usuario intuitiva y responsiva.
- 5. Especialista en Integración de Sistemas:** Conecta y sincroniza plataformas y servicios diversos para lograr ecosistemas digitales interoperables.
- 6. DevOps Engineer:** Automatiza procesos de desarrollo, prueba y despliegue para mejorar la eficiencia del ciclo de vida del Software.
- 7. Consultor en Soluciones Tecnológicas:** Asesora a empresas en la elección, implementación y mejora de sistemas informáticos personalizados.
- 8. Analista de Sistemas:** Examina requerimientos tecnológicos, modela soluciones y asegura que el Software cumpla con las necesidades del usuario.

“

Accede a un amplio abanico de oportunidades profesionales como ingeniero de Software, arquitecto de sistemas, consultor técnico o líder de desarrollo”

06

Licencias de software incluidas

TECH es referencia en el mundo universitario por combinar la última tecnología con las metodologías docentes para potencial el proceso de enseñanza-aprendizaje. Para ello, ha establecido una red de alianzas que le permite tener acceso a las herramientas de software más avanzadas del mundo profesional.



“

Al matricularte recibirás, de forma completamente gratuita, las credenciales de uso académico de las siguientes aplicaciones de software profesional”

TECH ha establecido una red de alianzas profesionales en la que se encuentran los principales proveedores de software aplicado a las diferentes áreas profesionales. Estas alianzas permiten a TECH tener acceso al uso de centenares de aplicaciones informáticas y licencias de software para acercarlas a sus estudiantes.

Las licencias de software para uno académico permitirán a los estudiantes utilizar las aplicaciones informáticas más avanzadas en su área profesional, de modo que podrán conocerlas y aprender su dominio sin tener que incurrir en costes. TECH se hará cargo del procedimiento de contratación para que los alumnos puedan utilizarlas de modo ilimitado durante el tiempo que estén estudiando el programa de Máster Título Propio en Calidad del Software, y además lo podrán hacer de forma completamente gratuita.

TECH te dará acceso gratuito al uso de las siguientes aplicaciones de software:



Google Career Launchpad

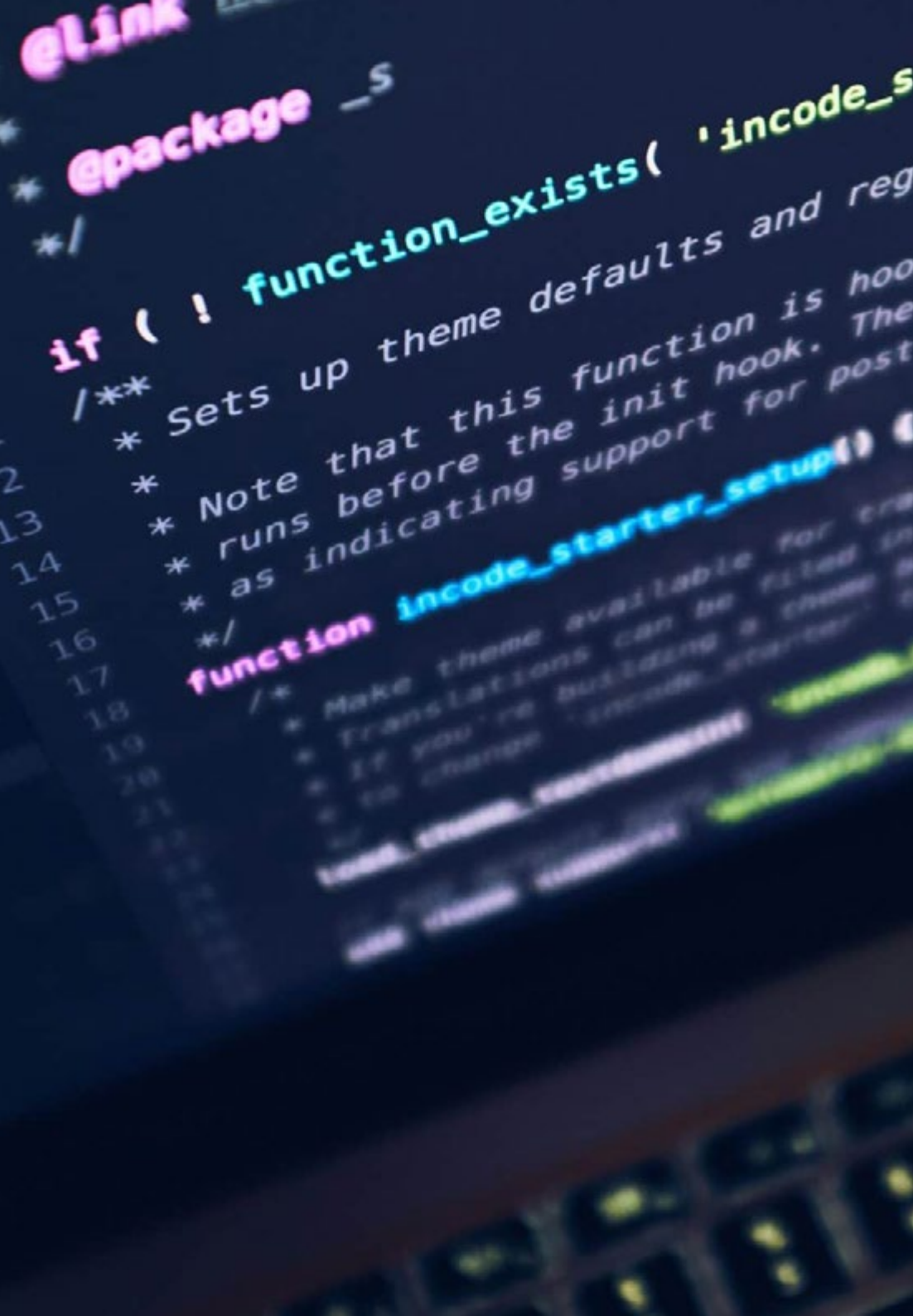
Google Career Launchpad es una solución para desarrollar habilidades digitales en tecnología y análisis de datos. Con un valor estimado de **5.000 dólares**, se incluye de forma gratuita en el programa universitario de TECH, brindando acceso a laboratorios interactivos y certificaciones reconocidas en el sector.

Esta plataforma combina capacitación técnica con casos prácticos, usando tecnologías como BigQuery y Google AI. Ofrece entornos simulados para experimentar con datos reales, junto a una red de expertos para orientación personalizada.

Funcionalidades destacadas:

- ♦ **Cursos especializados:** contenido actualizado en cloud computing, machine learning y análisis de datos
- ♦ **Laboratorios en vivo:** prácticas con herramientas reales de Google Cloud sin configuración adicional
- ♦ **Certificaciones integradas:** preparación para exámenes oficiales con validez internacional
- ♦ **Mentorías profesionales:** sesiones con expertos de Google y partners tecnológicos
- ♦ **Proyectos colaborativos:** retos basados en problemas reales de empresas líderes

En conclusión, **Google Career Launchpad** conecta a los usuarios con las últimas tecnologías del mercado, facilitando su inserción en áreas como inteligencia artificial y ciencia de datos con credenciales respaldadas por la industria.



“

Gracias a TECH podrás utilizar gratuitamente las mejores aplicaciones de software de tu área profesional”

07

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

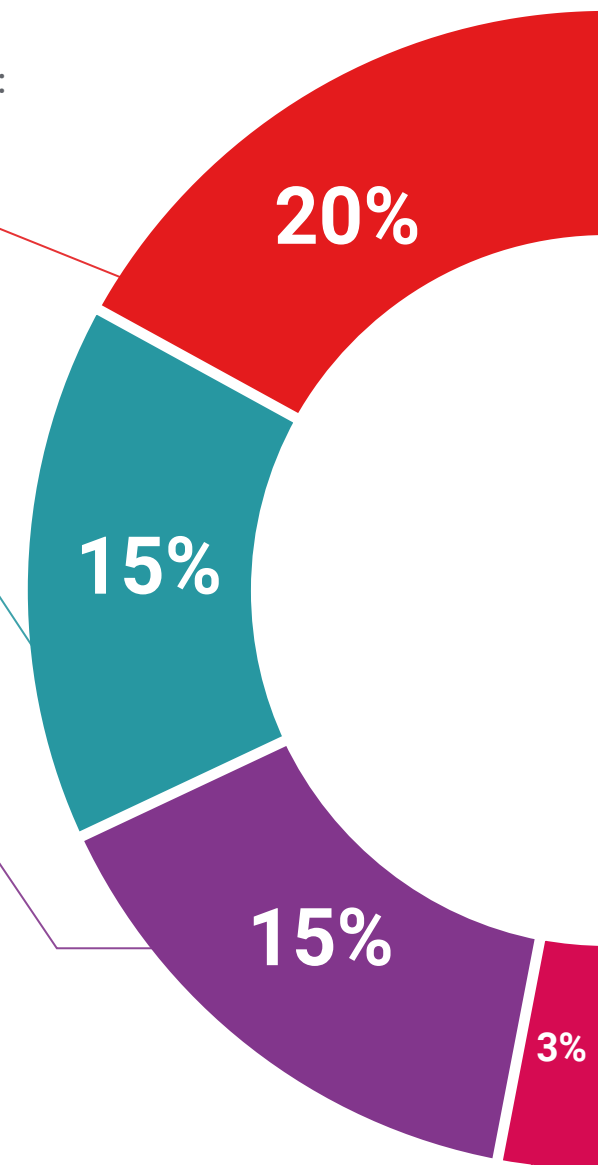
Presentamos los contenidos de manera atractiva y dinámica en píldoras multimedia que incluyen audio, vídeos, imágenes, esquemas y mapas conceptuales con el fin de afianzar el conocimiento.

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.





08

Cuadro docente

El cuadro docente de este programa representa un cruce excepcional entre experiencia práctica e innovación académica. Compuesto por profesionales que no solo dominan las últimas tecnologías en desarrollo y arquitectura de *Software*, sino que también las aplican activamente en entornos reales, este equipo garantiza una visión actualizada, dinámica y profundamente aplicada. Además, su capacidad para combinar conocimiento técnico con pensamiento estratégico permite transformar cada clase en un espacio de creación, análisis y resolución de retos concretos. Así, no solo se transmite contenido, sino que se cultiva una mentalidad preparada para liderar el futuro del *Software* desde múltiples perspectivas.



“

Un equipo docente conformado por especialistas de renombre en Calidad del Software te guiará durante el transcurso del programa académico”

Director Invitado Internacional

Con una extensa trayectoria profesional de más de 30 años en el sector tecnológico, Daniel St. John es un prestigioso **Ingeniero Informático** altamente especializado en **Calidad del Software**. En esta misma línea, se ha consolidado como un auténtico líder en este ámbito debido a su enfoque pragmático basado en la mejora continua e innovación.

A lo largo de su carrera laboral, ha formado parte de instituciones de referencia internacional como **General Electric Healthcare** en Illinois. De este modo, su labor se ha centrado en optimizar las **infraestructuras digitales** de las organizaciones con el objetivo de mejorar la **experiencia de los usuarios** significativamente. Gracias a esto, múltiples pacientes han disfrutado de una atención más personalizada y ágil, con un acceso más rápido tanto a los resultados clínicos como a los seguimientos de su salud. A su vez, ha implementado soluciones tecnológicas que han permitido a los profesionales mejorar la **toma de decisiones estratégicas** más informadas y fundamentadas en grandes volúmenes de datos.

También, ha compaginado esta labor con la creación de proyectos tecnológicos vanguardistas para maximizar la efectividad de los procesos operativos de las instituciones. Al respecto, ha liderado la **transformación digital** de numerosas compañías pertenecientes a diferentes industrias. Así pues, ha implementado instrumentos emergentes como la **Inteligencia Artificial**, el **Big Data** o **Machine Learning** para automatizar labores diarias complejas. Como resultado, dichas organizaciones han logrado adaptarse a las tendencias del mercado con inmediatez y garantizar su sostenibilidad a largo plazo.

Cabe destacar que Daniel St. John ha participado como ponente en diversos congresos científicos a escala global. De esta forma, ha compartido su vasto conocimiento en áreas como la adopción de **Metodologías Ágiles**, la realización de **Pruebas de Aplicaciones** para garantizar la fiabilidad de los sistemas o implementación de técnicas innovadoras de **Blockchain** que garantizan la protección de datos confidenciales.



D. St. John, Daniel

- Director Ingeniería de Software en General Electric Healthcare de Wisconsin, Estados Unidos
- Jefe de Ingeniería de Software en Siemens Healthineers, Illinois
- Director de Ingeniería de Software en Natus Medical Incorporated, Illinois
- Ingeniero Senior en WMS Gaming de Chicago
- Ingeniero Superior de Software en Siemens Medical Solutions, Illinois
- Máster en Estrategia y Análisis de Datos por Escuela de Postgrado en Gestión de Lake Forest
- Grado en Ciencias de la Computación por Universidad de Wisconsin-Parkside
- Miembro de la Junta Asesora del Instituto de Tecnología de Illinois
- Certificaciones en: Python para Ciencias de Datos, Inteligencia Artificial y Desarrollo, SAFe SCRUM y Project Management



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
- Introdutor de Inteligencia Artificial (ML y CV)
- Experto en Soluciones Basadas en Inteligencia Artificial en los campos de *Computer Vision*, ML/DL y NL
- 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

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

Dña. Martínez Cerrato, Yésica

- ♦ Responsable de Capacitaciones Técnicas en Securitas Seguridad España
- ♦ Especialista en Educación, Negocios y Marketing
- ♦ *Product Manager* en Seguridad Electrónica en Securitas Seguridad España
- ♦ Analista de Inteligencia Empresarial en Ricopia Technologies
- ♦ Técnico Informático y Responsable de Aulas informáticas OTEC en la Universidad de Alcalá de Henares
- ♦ Colaboradora en la Asociación ASALUMA
- ♦ Grado en Ingeniería Electrónica de Comunicaciones en la Escuela Politécnica Superior, Universidad de Alcalá de Henares



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

“

*Una experiencia de capacitación
única, clave y decisiva para
impulsar tu desarrollo profesional”*

09

Titulación

El Máster Título Propio en Calidad del Software garantiza, además de la capacitación más rigurosa y actualizada, el acceso a dos diplomas de Máster Propio, uno expedido por TECH Global University y otro expedido por Universidad FUNDEPOS.



“

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

El programa del **Máster Título Propio en Calidad del Software** es el más completo del panorama académico actual. A su egreso, el estudiante recibirá un diploma universitario emitido por TECH Global University, y otro por Universidad FUNDEPOS.

Estos títulos de formación permanente y actualización profesional de TECH Global University y Universidad FUNDEPOS garantizan la adquisición de competencias en el área de conocimiento, otorgando un alto valor curricular al estudiante que supere las evaluaciones y acredite el programa tras cursarlo en su totalidad.

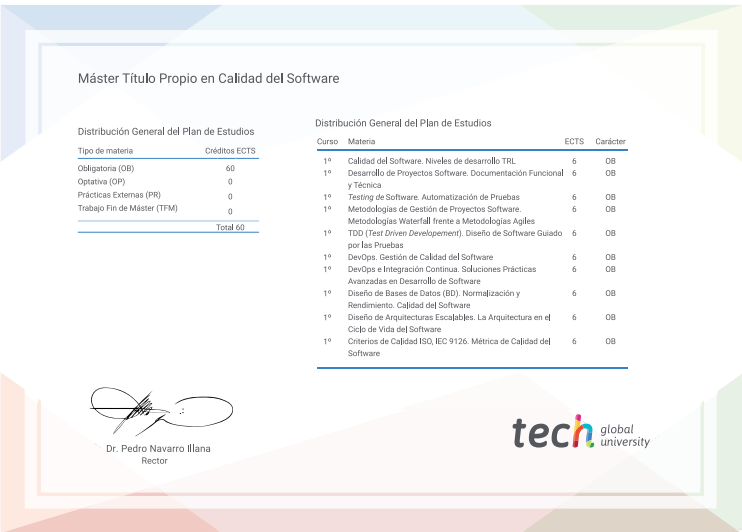
Este doble reconocimiento, de dos destacadas instituciones universitarias, suponen una doble recompensa a una formación integral y de calidad, asegurando que el estudiante obtenga una certificación reconocida tanto a nivel nacional como internacional. Este mérito académico le posicionará como un profesional altamente capacitado y preparado para enfrentar los retos y demandas en su área profesional.

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

Modalidad: **online**

Duración: **12 meses**

Acreditación: **60 ECTS**



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



Máster Título Propio Calidad del Software

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

Máster Título Propio

Calidad del Software

