

Máster Título Propio

Desarrollo de Software





Máster Título Propio Desarrollo de Software

- » Modalidad: No escolarizada (100% en línea)
- » Duración: 12 meses
- » Titulación: TECH Universidad
- » Horario: a tu ritmo
- » Exámenes: online

Acceso web: www.techtitute.com/informatica/master/master-desarrollo-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

Metodología de estudio

pág. 34

07

Cuadro docente

pág. 44

08

Titulación

pág. 48

01

Presentación del programa

La ingeniería del *Software* vive un crecimiento sin precedentes, con una demanda constante de expertos capaces de crear soluciones eficientes, escalables y seguras. Según la Organización de las Naciones Unidas, más del 60 % de las empresas enfrentan dificultades para cubrir vacantes en áreas tecnológicas, especialmente en Desarrollo de *Software*. Ante este panorama, es esencial contar con conocimientos actualizados y orientados a las necesidades reales del entorno informático. En ese sentido, TECH presenta una titulación innovadora y 100 % online en el ámbito de Desarrollo de *Software*, pensada para profesionales que deseen avanzar hacia un perfil técnico altamente competitivo y adaptado a los retos actuales del sector.



“

*Gracias a este programa 100% online,
manejarás el uso de metodologías ágiles
para optimizar el Desarrollo de Software
significativamente”*

El Desarrollo de Software es un eje estratégico para la transformación digital en todos los sectores productivos, desde la Salud o Educación hasta el Comercio. A este respecto, la creación de soluciones digitales eficientes, escalables y seguras marca la diferencia en la competitividad de empresas e instituciones. En este escenario, contar con especialistas capaces de diseñar, construir y mantener sistemas tecnológicos complejos se ha vuelto una prioridad a nivel global, impulsando la demanda de perfiles altamente cualificados en esta área.

Por eso, TECH lanza un innovador Máster Título Propio en Desarrollo de Software. Este programa permite adquirir competencias clave en programación, arquitectura de sistemas, metodologías ágiles y tecnologías emergentes, siempre con un enfoque práctico y actualizado. A través de un plan de estudios riguroso, orientado a la resolución de problemas reales, se fomenta una visión integral del Desarrollo de Software. Esta preparación no solo mejora las oportunidades laborales, sino que también facilita el acceso a roles de liderazgo técnico, consultoría y diseño de soluciones innovadoras con impacto en diversos sectores.

Cabe destacar que el formato online del programa permite avanzar sin barreras geográficas ni restricciones horarias. Asimismo, la plataforma virtual ofrece acceso continuo a contenidos de calidad, recursos interactivos y ejercicios aplicados, favoreciendo un aprendizaje flexible, autónomo y adaptado a cada ritmo. Esta modalidad facilita la conciliación con otras actividades, al tiempo que promueve una experiencia educativa alineada con las dinámicas actuales del entorno tecnológico y profesional.

En adición, el programa cuenta con la participación de un reconocido Director Invitado Internacional que ofrecerá 10 rigurosas *Masterclasses*.

Este **Máster Título Propio en Desarrollo de 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 están concebidos recogen una información científica y práctica sobre aquellas disciplinas indispensables para el ejercicio profesional
- Los ejercicios prácticos donde realizar el proceso de autoevaluación para mejorar el aprendizaje
- Su especial hincapié en metodologías innovadoras Desarrollo de Software
- Las lecciones teóricas, preguntas al experto, foros de discusión de temas controvertidos y trabajos de reflexión individual
- La disponibilidad de acceso a los contenidos desde cualquier dispositivo fijo o portátil con conexión a internet



Un prestigioso Director Invitado Internacional brindará 10 exclusivas Masterclasses sobre las últimas tendencias en el Desarrollo de Software”

“

Desarrollarás aplicaciones distribuidas y en red que respondan a los desafíos de conectividad, seguridad y rendimiento que exige la transformación digital”

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.

Integrarás bases de datos relacionales en arquitecturas complejas, optimizando la gestión de grandes volúmenes de información.

Mediante el sistema Relearning que emplea TECH, reducirás las largas horas de estudio y memorización.



02

¿Por qué estudiar en TECH?

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



“

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

La mejor universidad online del mundo según FORBES

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

El mejor claustro docente top internacional

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

La mayor universidad digital del mundo

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



Forbes
Mejor universidad
online del mundo

Plan
de estudios
más completo

Profesorado
TOP
Internacional

La metodología
más eficaz

nº1
Mundial
Mayor universidad
online del mundo

Los planes de estudio más completos del panorama universitario

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

Un método de aprendizaje único

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

La universidad online oficial de la NBA

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

Líderes en empleabilidad

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



Google Partner Premier

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



La universidad mejor valorada por sus alumnos

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



03

Plan de estudios

Este plan de estudios integra contenidos alineados con las demandas actuales del sector tecnológico, combinando teoría avanzada con aplicación práctica. Además, incorpora enfoques sobre ciberseguridad, desarrollo multiplataforma y arquitectura orientada a servicios, permitiendo una visión integral y moderna del entorno informático. Así, se ofrece una experiencia académica rigurosa y actualizada, pensada para enfrentar los retos de la transformación digital con solidez técnica y criterio innovador.

```
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
  
<?if ($send==2) {?>  
<td align="right" colspan="2">  
<input type="button" value="<?=checkLangItem("contact-atpaka1")?>" onclick="window.location='kontakti.ph  
</td>  
</tr>  
<tr>  
<td align="right" colspan="2">  
<div class="text-2">  
<?=checkLangItem("contact-paldies")?>  
</div>  
</td>
```

```
checkLangItem("contact-text-1")?></div>  
sci/index.php?sid=' + Math.r
```





“

Dominarás los fundamentos esenciales del Desarrollo de Software para construir soluciones tecnológicas sólidas, eficientes y adaptadas a las necesidades del entorno profesional”

Módulo 1. Fundamentos de programación

- 1.1. Introducción a la programación
 - 1.1.1. Estructura básica de un ordenador
 - 1.1.2. Software
 - 1.1.3. Lenguajes de programación
 - 1.1.4. Ciclo de vida de una aplicación informática
- 1.2. Diseño de algoritmos
 - 1.2.1. La resolución de problemas
 - 1.2.2. Técnicas descriptivas
 - 1.2.3. Elementos y estructura de un algoritmo
- 1.3. Elementos de un programa
 - 1.3.1. Origen y características del lenguaje C++
 - 1.3.2. El entorno de desarrollo
 - 1.3.3. Concepto de programa
 - 1.3.4. Tipos de datos fundamentales
 - 1.3.5. Operadores
 - 1.3.6. Expresiones
 - 1.3.7. Sentencias
 - 1.3.8. Entrada y salida de datos
- 1.4. Sentencias de control
 - 1.4.1. Sentencias
 - 1.4.2. Bifurcaciones
 - 1.4.3. Bucles
- 1.5. Abstracción y modularidad: funciones
 - 1.5.1. Diseño modular
 - 1.5.2. Concepto de función y utilidad
 - 1.5.3. Definición de una función
 - 1.5.4. Flujo de ejecución en la llamada de una función
 - 1.5.5. Prototipo de una función
 - 1.5.6. Devolución de resultados
 - 1.5.7. Llamada a una función: parámetros
 - 1.5.8. Paso de parámetros por referencia y por valor
 - 1.5.9. Ámbito identificador





- 1.6. Estructuras de datos estáticas
 - 1.6.1. Arrays
 - 1.6.2. Matrices. Poliedros
 - 1.6.3. Búsqueda y ordenación
 - 1.6.4. Cadenas. Funciones de E/S para cadenas
 - 1.6.5. Estructuras. Uniones
 - 1.6.6. Nuevos tipos de datos
- 1.7. Estructuras de datos dinámicas: punteros
 - 1.7.1. Concepto. Definición de puntero
 - 1.7.2. Operadores y operaciones con punteros
 - 1.7.3. Arrays de punteros
 - 1.7.4. Punteros y arrays
 - 1.7.5. Punteros a cadenas
 - 1.7.6. Punteros a estructuras
 - 1.7.7. Indirección múltiple
 - 1.7.8. Punteros a funciones
 - 1.7.9. Paso de funciones, estructuras y arrays como parámetros de funciones
- 1.8. Ficheros
 - 1.8.1. Conceptos básicos
 - 1.8.2. Operaciones con ficheros
 - 1.8.3. Tipos de ficheros
 - 1.8.4. Organización de los ficheros
 - 1.8.5. Introducción a los ficheros C++
 - 1.8.6. Manejo de ficheros
- 1.9. Recursividad
 - 1.9.1. Definición de recursividad
 - 1.9.2. Tipos de recursión
 - 1.9.3. Ventajas e inconvenientes
 - 1.9.4. Consideraciones
 - 1.9.5. Conversión recursivo iterativa
 - 1.9.6. La pila de recursión

- 1.10. Prueba y documentación
 - 1.10.1. Pruebas de programas
 - 1.10.2. Prueba de la caja blanca
 - 1.10.3. Prueba de la caja negra
 - 1.10.4. Herramientas para realizar las pruebas
 - 1.10.5. Documentación de programas

Módulo 2. Estructura de datos

- 2.1. Introducción a la programación en C++
 - 2.1.1. Clases, constructores, métodos y atributos
 - 2.1.2. Variables
 - 2.1.3. Expresiones condicionales y bucles
 - 2.1.4. Objetos
- 2.2. Tipos abstractos de datos (TAD)
 - 2.2.1. Tipos de datos
 - 2.2.2. Estructuras básicas y TAD
 - 2.2.3. Vectores y Arrays
- 2.3. Estructuras de datos lineales
 - 2.3.1. TAD Lista definición
 - 2.3.2. Listas enlazadas y doblemente enlazadas
 - 2.3.3. Listas ordenadas
 - 2.3.4. Listas en C++
 - 2.3.5. TAD Pila
 - 2.3.6. TAD Cola
 - 2.3.7. Pila y Cola en C++
- 2.4. Estructuras de datos jerárquicas
 - 2.4.1. TAD Árbol
 - 2.4.2. Recorridos
 - 2.4.3. Árboles n-arios
 - 2.4.4. Árboles binarios
 - 2.4.5. Árboles binarios de búsqueda
- 2.5. Estructuras de datos jerárquicas: árboles complejos
 - 2.5.1. Árboles perfectamente equilibrados o de altura mínima
 - 2.5.2. Árboles multicamino

- 2.5.3. Referencias bibliográficas
- 2.6. Montículos y Cola de prioridad
 - 2.6.1. TAD Montículos
 - 2.6.2. TAD Cola de prioridad
- 2.7. Tablas Hash
 - 2.7.1. TAD Tabla Hash
 - 2.7.2. Funciones Hash
 - 2.7.3. Función Hash en tablas Hash
 - 2.7.4. Redispersión
 - 2.7.5. Tablas Hash abiertas
- 2.8. Grafos
 - 2.8.1. TAD Grafo
 - 2.8.2. Tipos de Grafo
 - 2.8.3. Representación gráfica y operaciones básicas
 - 2.8.4. Diseño de Grafo
- 2.9. Algoritmos y conceptos avanzados sobre Grafos
 - 2.9.1. Problemas sobre Grafos
 - 2.9.2. Algoritmos sobre caminos
 - 2.9.3. Algoritmos de búsqueda o recorridos
 - 2.9.4. Otros algoritmos
- 2.10. Otras estructuras de datos
 - 2.10.1. Conjuntos
 - 2.10.2. Arrays paralelos
 - 2.10.3. Tablas de símbolos
 - 2.10.4. Tries

Módulo 3. Algoritmia y complejidad

- 3.1. Introducción a las estrategias de diseño de algoritmos
 - 3.1.1. Recursividad
 - 3.1.2. Divide y conquista
 - 3.1.3. Otras estrategias
- 3.2. Eficiencia y análisis de los algoritmos
 - 3.2.1. Medidas de eficiencia
 - 3.2.2. Medir el tamaño de la entrada

- 3.2.3. Medir el tiempo de ejecución
- 3.2.4. Caso peor, mejor y medio
- 3.2.5. Notación asintótica
- 3.2.6. Criterios de análisis matemático de algoritmos no recursivos
- 3.2.7. Análisis matemático de algoritmos recursivos
- 3.2.8. Análisis empírico de algoritmos
- 3.3. Algoritmos de ordenación
 - 3.3.1. Concepto de ordenación
 - 3.3.2. Ordenación de la burbuja
 - 3.3.3. Ordenación por selección
 - 3.3.4. Ordenación por inserción
 - 3.3.5. Ordenación por mezcla (Merge Sort)
 - 3.3.6. Ordenación rápida (QuickSort)
- 3.4. Algoritmos con árboles
 - 3.4.1. Concepto de árbol
 - 3.4.2. Árboles binarios
 - 3.4.3. Recorridos de árbol
 - 3.4.4. Representar expresiones
 - 3.4.5. Árboles binarios ordenados
 - 3.4.6. Árboles binarios balanceados
- 3.5. Algoritmos con Heaps
 - 3.5.1. Los Heaps
 - 3.5.2. El algoritmo HeapSort
 - 3.5.3. Las colas de prioridad
- 3.6. Algoritmos con grafos
 - 3.6.1. Representación
 - 3.6.2. Recorrido en anchura
 - 3.6.3. Recorrido en profundidad
 - 3.6.4. Ordenación topológica
- 3.7. Algoritmos Greedy
 - 3.7.1. La estrategia Greedy
 - 3.7.2. Elementos de la estrategia Greedy
 - 3.7.3. Cambio de monedas
 - 3.7.4. Problema del viajante

- 3.7.5. Problema de la mochila
- 3.8. Búsqueda de caminos mínimos
 - 3.8.1. El problema del camino mínimo
 - 3.8.2. Arcos negativos y ciclos
 - 3.8.3. Algoritmo de Dijkstra
- 3.9. Algoritmos Greedy sobre Grafos
 - 3.9.1. El árbol de recubrimiento mínimo
 - 3.9.2. El algoritmo de Prim
 - 3.9.3. El algoritmo de Kruskal
 - 3.9.4. Análisis de complejidad
- 3.10. Backtracking
 - 3.10.1. El Backtracking
 - 3.10.2. Técnicas alternativas

Módulo 4. Bases de datos

- 4.1. Aplicaciones y propósitos de los sistemas de base de datos
 - 4.1.1. Aplicaciones de los diferentes sistemas de base de datos
 - 4.1.2. Propósito en los diferentes sistemas de base de datos
 - 4.1.3. Visión de los datos
- 4.2. Base de datos y arquitectura
 - 4.2.1. Base de datos relacionales
 - 4.2.2. El diseño de base de datos
 - 4.2.3. Bases de datos basadas en objetos y semiestructuradas
 - 4.2.4. Almacenamiento de datos y consultas
 - 4.2.5. Gestión de transacciones
 - 4.2.6. Minería y análisis de datos
 - 4.2.7. Arquitectura de las bases de datos
- 4.3. El modelo relacional: estructura, operaciones y álgebra relacional extendida
 - 4.3.1. La estructura de las BD relacionales
 - 4.3.2. Operaciones fundamentales en el álgebra relacional
 - 4.3.3. Otras operaciones del álgebra relacional
 - 4.3.4. Operaciones del álgebra relacional extendida
 - 4.3.5. Valores nulos
 - 4.3.6. Modificación de la base de datos

- 4.4. SQL (I)
 - 4.4.1. ¿Qué es SQL?
 - 4.4.2. La definición de datos
 - 4.4.3. Estructura básica de las consultas SQL
 - 4.4.4. Operaciones sobre conjuntos
 - 4.4.5. Funciones de agregación
 - 4.4.6. Valores nulos
- 4.5. SQL (II)
 - 4.5.1. Subconsultas anidadas
 - 4.5.2. Consultas complejas
 - 4.5.3. Vistas
 - 4.5.4. Cursores
 - 4.5.5. Consultas complejas
 - 4.5.6. Disparadores
- 4.6. Diseño de base de datos y el modelo E-R
 - 4.6.1. Visión general del proceso de diseño
 - 4.6.2. El modelo entidad relación
 - 4.6.3. Restricciones
- 4.7. Diagramas entidad relación
 - 4.7.1. Diagramas entidad relación
 - 4.7.2. Aspectos del diseño entidad relación
 - 4.7.3. Conjuntos de entidades débiles
- 4.8. El modelo entidad relación extendido
 - 4.8.1. Características del modelo E-R extendido
 - 4.8.2. Diseño de una base de datos
 - 4.8.3. Reducción a esquemas relacionales
- 4.9. Diseño de bases de datos relacionales
 - 4.9.1. Características de los buenos diseños relacionales
 - 4.9.2. Dominios atómicos y la primera forma normal (1FN)
 - 4.9.3. Descomposición mediante dependencias funcionales
 - 4.9.4. Teoría de las dependencias funcionales
 - 4.9.5. Algoritmos de descomposición
 - 4.9.6. Descomposición mediante dependencias multivaloradas

- 4.9.7. Más formas normales
- 4.9.8. Proceso de diseño de las base de datos
- 4.10. Bases de datos NoSQL
 - 4.10.1. ¿Qué son las bases de datos NoSQL?
 - 4.10.2. Análisis de las diferentes opciones de NoSQL y sus características
 - 4.10.3. MongoDB

Módulo 5. Bases de datos avanzadas

- 5.1. Introducción a los diferentes sistemas de bases de datos
 - 5.1.1. Repaso histórico
 - 5.1.2. Bases de datos jerárquicas
 - 5.1.3. Bases de datos red
 - 5.1.4. Bases de datos relacionales
 - 5.1.5. Bases de datos no relacionales
- 5.2. XML y bases de datos para la web
 - 5.2.1. Validación de documentos XML
 - 5.2.2. Transformaciones de documentos XML
 - 5.2.3. Almacenamiento de datos XML
 - 5.2.4. Bases de datos relacionales XML
 - 5.2.5. SQL/XML
 - 5.2.6. Bases de datos nativas XML
- 5.3. Bases de datos paralelas
 - 5.3.1. Sistemas paralelos
 - 5.3.2. Arquitecturas paralelas de bases de datos
 - 5.3.3. Paralelismo en consultas
 - 5.3.4. Paralelismo entre consultas
 - 5.3.5. Diseño de sistemas paralelos
 - 5.3.6. Procesamiento paralelo en SQL
- 5.4. Bases de datos distribuidas
 - 5.4.1. Sistemas distribuidos
 - 5.4.2. Almacenamiento distribuido
 - 5.4.3. Disponibilidad
 - 5.4.4. Procesamiento distribuido de consultas
 - 5.4.5. Proveedores de bases de datos distribuidas

- 5.5. Indexación y asociación
 - 5.5.1. Índices ordenados
 - 5.5.2. Índices densos y dispersos
 - 5.5.3. Índices multinivel
 - 5.5.4. Actualización del índice
 - 5.5.5. Asociación estática
 - 5.5.6. Cómo usar índices en bases de datos
- 5.6. Introducción al procesamiento transaccional
 - 5.6.1. Estados de una transacción
 - 5.6.2. Implementación de la atomicidad y durabilidad
 - 5.6.3. Secuencialidad
 - 5.6.4. Recuperabilidad
 - 5.6.5. Implementación del aislamiento
- 5.7. Sistemas de recuperación
 - 5.7.1. Clasificación de fallos
 - 5.7.2. Estructuras de almacenamiento
 - 5.7.3. Recuperación y atomicidad
 - 5.7.4. Recuperación basada en registro histórico
 - 5.7.5. Transacciones concurrentes y recuperación
 - 5.7.6. Alta disponibilidad en bases de datos
- 5.8. Ejecución y procesamiento de consultas
 - 5.8.1. Coste de una consulta
 - 5.8.2. Operación de selección
 - 5.8.3. Ordenación
 - 5.8.4. Introducción a la optimización de consultas
 - 5.8.5. Monitorización del rendimiento
- 5.9. Bases de datos no relacionales
 - 5.9.1. Bases de datos orientadas a documentos
 - 5.9.2. Bases de datos orientadas a Grafos
 - 5.9.3. Bases de datos clave-valor
- 5.10. Data Warehouse, OLAP y minería de datos
 - 5.10.1. Componentes de los almacenes de datos
 - 5.10.2. Arquitectura de un data Warehouse
 - 5.10.3. OLAP

- 5.10.4. Funcionalidades de la minería de datos
- 5.10.5. Otros tipos de minería

Módulo 6. Diseño avanzado de algoritmos

- 6.1. Análisis de algoritmos recursivos y tipo divide y conquista
 - 6.1.1. Planteamiento y resolución de ecuaciones de recurrencia homogéneas y no homogéneas
 - 6.1.2. Descripción general de la estrategia divide y conquista
- 6.2. Análisis amortizado
 - 6.2.1. El análisis agregado
 - 6.2.2. El método de contabilidad
 - 6.2.3. El método del potencial
- 6.3. Programación dinámica y algoritmos para problemas NP
 - 6.3.1. Características de la programación dinámica
 - 6.3.2. Vuelta atrás: Backtracking
 - 6.3.3. Ramificación y poda
- 6.4. Optimización combinatoria
 - 6.4.1. Representación de problemas
 - 6.4.2. Optimización en 1D
- 6.5. Algoritmos de aleatorización
 - 6.5.1. Ejemplos de algoritmos de aleatorización
 - 6.5.2. El teorema Buffon
 - 6.5.3. Algoritmo de Monte Carlo
 - 6.5.4. Algoritmo Las Vegas
- 6.6. Búsqueda local y con candidatos
 - 6.6.1. Gradient Ascent
 - 6.6.2. Hill Climbing
 - 6.6.3. Simulated Annealing
 - 6.6.4. Tabu Search
 - 6.6.5. Búsqueda con candidatos
- 6.7. Verificación formal de programas
 - 6.7.1. Especificación de abstracciones funcionales
 - 6.7.2. El lenguaje de la lógica de primer orden
 - 6.7.3. El sistema formal de Hoare

- 6.8. Verificación de programas iterativos
 - 6.8.1. Reglas del sistema formal de Hoare
 - 6.8.2. Concepto de invariante de iteraciones
- 6.9. Métodos numéricos
 - 6.9.1. El método de la bisección
 - 6.9.2. El método de Newton Raphson
 - 6.9.3. El método de la secante
- 6.10. Algoritmos paralelos
 - 6.10.1. Operaciones binarias paralelas
 - 6.10.2. Operaciones paralelas con grafos
 - 6.10.3. Paralelismo en divide y vencerás
 - 6.10.4. Paralelismo en programación dinámica

Módulo 7. Interacción persona ordenador

- 7.1. Introducción a la interacción persona ordenador
 - 7.1.1. Qué es la interacción persona ordenador
 - 7.1.2. Relación de la interacción persona ordenador con otras disciplinas
 - 7.1.3. La interfaz de usuario
 - 7.1.4. Usabilidad y accesibilidad
 - 7.1.5. Experiencia de usuario y diseño centrado en el usuario
- 7.2. El ordenador y la interacción: interfaz de usuario y paradigmas de interacción
 - 7.2.1. La interacción
 - 7.2.2. Paradigmas y estilos de interacción
 - 7.2.3. Evolución de las interfaces de usuario
 - 7.2.4. Interfaces de usuario clásicas: WIMP/GUI, comandos, voz, realidad virtual
 - 7.2.5. Interfaces de usuario innovadoras: móviles, portátiles, colaborativas, BCI
- 7.3. El factor humano: aspectos psicológicos y cognitivos
 - 7.3.1. La importancia del factor humano en la interacción
 - 7.3.2. El procesamiento humano de información
 - 7.3.3. La entrada y salida de la información: visual, auditiva y táctil
 - 7.3.4. Percepción y atención
 - 7.3.5. Conocimiento y modelos mentales: representación, organización y adquisición
- 7.4. El factor humano: limitaciones sensoriales y físicas
 - 7.4.1. Diversidad funcional, discapacidad y deficiencia
 - 7.4.2. Diversidad visual
 - 7.4.3. Diversidad auditiva
 - 7.4.4. Diversidad cognitiva
 - 7.4.5. Diversidad motórica
 - 7.4.6. El caso de los inmigrantes digitales
- 7.5. El proceso de diseño (I): análisis de requisitos para el diseño de la interfaz de usuario
 - 7.5.1. Diseño centrado en el usuario
 - 7.5.2. Qué es el análisis de requisitos
 - 7.5.3. La recogida de información
 - 7.5.4. Análisis e interpretación de la información
 - 7.5.5. Análisis de la usabilidad y la accesibilidad
- 7.6. El proceso de diseño (II): prototipado y análisis de tareas
 - 7.6.1. Diseño conceptual
 - 7.6.2. Prototipado
 - 7.6.3. Análisis jerárquico de tareas
- 7.7. El proceso de diseño (III): la evaluación
 - 7.7.1. Evaluación en el proceso de diseño: objetivos y métodos
 - 7.7.2. Métodos de evaluación sin usuarios
 - 7.7.3. Métodos de evaluación con usuarios
 - 7.7.4. Estándares y normas de evaluación
- 7.8. Accesibilidad: definición y pautas
 - 7.8.1. Accesibilidad y diseño universal
 - 7.8.2. La iniciativa WAI y las pautas WCAG
 - 7.8.3. Pautas WCAG 2.0 y 2.1
- 7.9. Accesibilidad: evaluación y diversidad funcional
 - 7.9.1. Herramientas de evaluación de la accesibilidad en la web
 - 7.9.2. Accesibilidad y diversidad funcional
- 7.10. El ordenador y la interacción: periféricos y dispositivos
 - 7.10.1. Dispositivos y periféricos tradicionales
 - 7.10.2. Dispositivos y periféricos alternativos
 - 7.10.3. Móviles y tabletas
 - 7.10.4. Diversidad funcional, interacción y periféricos

Módulo 8. Programación avanzada

- 8.1. Introducción a la programación orientada a objetos
 - 8.1.1. Introducción a la programación orientada a objetos
 - 8.1.2. Diseño de clases
 - 8.1.3. Introducción a UML para el modelado de los problemas
- 8.2. Relaciones entre clases
 - 8.2.1. Abstracción y herencia
 - 8.2.2. Conceptos avanzados de herencia
 - 8.2.3. Polimorfismo
 - 8.2.4. Composición y agregación
- 8.3. Introducción a los patrones de diseño para problemas orientados a objetos
 - 8.3.1. Qué son los patrones de diseño
 - 8.3.2. Patrón Factory
 - 8.3.3. Patrón Singleton
 - 8.3.4. Patrón Observer
 - 8.3.5. Patrón Composite
- 8.4. Excepciones
 - 8.4.1. ¿Qué son las excepciones?
 - 8.4.2. Captura y gestión de excepciones
 - 8.4.3. Lanzamiento de excepciones
 - 8.4.4. Creación de excepciones
- 8.5. Interfaces de usuarios
 - 8.5.1. Introducción a Qt
 - 8.5.2. Posicionamiento
 - 8.5.3. ¿Qué son los eventos?
 - 8.5.4. Eventos: definición y captura
 - 8.5.5. Desarrollo de interfaces de usuario
- 8.6. Introducción a la programación concurrente
 - 8.6.1. Introducción a la programación concurrente
 - 8.6.2. El concepto de proceso e hilo
 - 8.6.3. Interacción entre procesos o hilos
 - 8.6.4. Los hilos en C++
 - 8.6.5. Ventajas e inconvenientes de la programación concurrente

- 8.7. Gestión de hilos y sincronización
 - 8.7.1. Ciclo de vida de un hilo
 - 8.7.2. La clase Thread
 - 8.7.3. Planificación de hilos
 - 8.7.4. Grupos hilos
 - 8.7.5. Hilos de tipo demonio
 - 8.7.6. Sincronización
 - 8.7.7. Mecanismos de bloqueo
 - 8.7.8. Mecanismos de comunicación
 - 8.7.9. Monitores
- 8.8. Problemas comunes dentro de la programación concurrente
 - 8.8.1. El problema de los productores consumidores
 - 8.8.2. El problema de los lectores y escritores
 - 8.8.3. El problema de la cena de los filósofos
- 8.9. Documentación y pruebas de software
 - 8.9.1. ¿Por qué es importante documentar el software?
 - 8.9.2. Documentación de diseño
 - 8.9.3. Uso de herramientas para la documentación
- 8.10. Pruebas de software
 - 8.10.1. Introducción a las pruebas del software
 - 8.10.2. Tipos de pruebas
 - 8.10.3. Prueba de unidad
 - 8.10.4. Prueba de integración
 - 8.10.5. Prueba de validación
 - 8.10.6. Prueba del sistema

Módulo 9. Desarrollo de aplicaciones en red

- 9.1. Lenguajes de marcado HTML5
 - 9.1.1. Conceptos básicos de HTML
 - 9.1.2. Nuevos elementos HTML 5
 - 9.1.3. Formularios: nuevos controles
- 9.2. Introducción a hojas de estilo CSS
 - 9.2.1. Primeros pasos con CSS
 - 9.2.2. Introducción a CSS3

- 9.3. Lenguaje script de navegador: JavaScript
 - 9.3.1. Conceptos básicos de JavaScript
 - 9.3.2. DOM
 - 9.3.3. Eventos
 - 9.3.4. JQuery
 - 9.3.5. Ajax
- 9.4. Concepto de la programación orientada a componentes
 - 9.4.1. Contexto
 - 9.4.2. Componentes e interfaces
 - 9.4.3. Estados de un componente
- 9.5. Arquitectura de componentes
 - 9.5.1. Arquitecturas actuales
 - 9.5.2. Integración y despliegue de componentes
- 9.6. Framework Frontend: Bootstrap
 - 9.6.1. Diseño con rejilla
 - 9.6.2. Formularios
 - 9.6.3. Componentes
- 9.7. Modelo vista controlador
 - 9.7.1. Métodos de desarrollo Web
 - 9.7.2. Patrón de diseño: MVC
- 9.8. Tecnologías Grid de la información
 - 9.8.1. Incremento de recursos en computación
 - 9.8.2. Concepto de tecnología Grid
- 9.9. Arquitectura orientada a servicios
 - 9.9.1. SOA y servicios web
 - 9.9.2. Topología de un servicio web
 - 9.9.3. Plataformas para los servicios web
- 9.10. Protocolo HTTP
 - 9.10.1. Mensajes
 - 9.10.2. Sesiones persistentes
 - 9.10.3. Sistema criptográfico
 - 9.10.4. Funcionamiento del protocolo HTTPS

Módulo 10. Ingeniería del Software

- 10.1. Introducción a la ingeniería del software y al modelado
 - 10.1.1. La naturaleza del software
 - 10.1.2. La naturaleza única de las WebApps
 - 10.1.3. Ingeniería del software
 - 10.1.4. El proceso del software
 - 10.1.5. La práctica de la ingeniería del software
 - 10.1.6. Mitos del software
 - 10.1.7. Cómo comienza todo
 - 10.1.8. Conceptos orientados a objetos
 - 10.1.9. Introducción a UML
- 10.2. El proceso del software
 - 10.2.1. Un modelo general de proceso
 - 10.2.2. Modelos de proceso prescriptivos
 - 10.2.3. Modelos de proceso especializado
 - 10.2.4. El proceso unificado
 - 10.2.5. Modelos del proceso personal y del equipo
 - 10.2.6. ¿Qué es la agilidad?
 - 10.2.7. ¿Qué es un proceso ágil?
 - 10.2.8. Scrum
 - 10.2.9. Conjunto de herramientas para el proceso ágil
- 10.3. Principios que guían la práctica de la ingeniería del software
 - 10.3.1. Principios que guían el proceso
 - 10.3.2. Principios que guían la práctica
 - 10.3.3. Principios de comunicación
 - 10.3.4. Principios de planificación
 - 10.3.5. Principios de modelado
 - 10.3.6. Principios de construcción
 - 10.3.7. Principios de despliegue
- 10.4. Comprensión de los requisitos
 - 10.4.1. Ingeniería de requisitos
 - 10.4.2. Establecer las bases
 - 10.4.3. Indagación de los requisitos

- 10.4.4. Desarrollo de casos de uso
- 10.4.5. Elaboración del modelo de los requisitos
- 10.4.6. Negociación de los requisitos
- 10.4.7. Validación de los requisitos
- 10.5. Modelado de los requisitos: escenarios, información y clases de análisis
 - 10.5.1. Análisis de los requisitos
 - 10.5.2. Modelado basado en escenarios
 - 10.5.3. Modelos UML que proporcionan el caso de uso
 - 10.5.4. Conceptos de modelado de datos
 - 10.5.5. Modelado basado en clases
 - 10.5.6. Diagramas de clases
- 10.6. Modelado de los requisitos: flujo, comportamiento y patrones
 - 10.6.1. Requisitos que modelan las estrategias
 - 10.6.2. Modelado orientado al flujo
 - 10.6.3. Diagramas de estado
 - 10.6.4. Creación de un modelo de comportamiento
 - 10.6.5. Diagramas de secuencia
 - 10.6.6. Diagramas de comunicación
 - 10.6.7. Patrones para el modelado de requisitos
- 10.7. Conceptos de diseño
 - 10.7.1. Diseño en el contexto de la ingeniería del software
 - 10.7.2. El proceso de diseño
 - 10.7.3. Conceptos de diseño
 - 10.7.4. Conceptos de diseño orientado a objetos
 - 10.7.5. El modelo del diseño
- 10.8. Diseño de la arquitectura
 - 10.8.1. Arquitectura del software
 - 10.8.2. Géneros arquitectónicos
 - 10.8.3. Estilos arquitectónicos
 - 10.8.4. Diseño arquitectónico
 - 10.8.5. Evolución de los diseños alternativos para la arquitectura
 - 10.8.6. Mapeo de la arquitectura con el uso del flujo de datos
- 10.9. Diseño en el nivel de componentes y basado en patrones
 - 10.9.1. ¿Qué es un componente?
 - 10.9.2. Diseño de componentes basados en clase
 - 10.9.3. Realización del diseño en el nivel de componentes
 - 10.9.4. Diseño de componentes tradicionales
 - 10.9.5. Desarrollo basado en componentes
 - 10.9.6. Patrones de diseño
 - 10.9.7. Diseño de software basado en patrones
 - 10.9.8. Patrones arquitectónicos
 - 10.9.9. Patrones de diseño en el nivel de componentes
 - 10.9.10. Patrones de diseño de la interfaz de usuario
- 10.10. Calidad del software y administración de proyectos
 - 10.10.1. Calidad del software
 - 10.10.2. El dilema de la calidad del software
 - 10.10.3. Lograr la calidad del software
 - 10.10.4. Aseguramiento de la calidad del software
 - 10.10.5. El espectro administrativo
 - 10.10.6. El personal
 - 10.10.7. El producto
 - 10.10.8. El proceso
 - 10.10.9. El proyecto
 - 10.10.10. Principios y prácticas



Aplicarás criterios de calidad, rendimiento y sostenibilidad en todo el ciclo de vida del Software”

04

Objetivos docentes

Los objetivos docentes de este programa responden a las exigencias de un ecosistema digital en constante evolución. No solo se busca desarrollar competencias técnicas avanzadas, sino también fomentar el pensamiento crítico, la resolución ágil de problemas y la capacidad de innovación. A través de un enfoque multidisciplinar y actualizado, se pretende formar profesionales capaces de liderar proyectos tecnológicos, adaptarse a nuevas herramientas y aplicar metodologías emergentes con eficiencia. Además, se promueve una comprensión profunda del ciclo completo de desarrollo, integrando aspectos clave como la seguridad, la escalabilidad y la sostenibilidad del software en entornos altamente competitivos y dinámicos.



```
!!$_GET[type]) echo "current";  
type=1{text_margin}>  
</div>  
ang'] == 'rus') ed
```

“

Especialízate en el diseño de experiencias digitales centradas en el usuario, empleando técnicas de interacción persona-ordenador y accesibilidad”



Objetivos generales

- ♦ Desarrollar competencias técnicas avanzadas en programación, diseño, arquitectura de *Software* y metodologías asociadas
- ♦ Aplicar metodologías ágiles y enfoques de desarrollo colaborativo en proyectos tecnológicos de distinta complejidad
- ♦ Integrar herramientas de automatización, control de versiones y pruebas en entornos de desarrollo profesional modernos
- ♦ Analizar y diseñar soluciones escalables, seguras, eficientes y sostenibles en diversos contextos y sectores digitales
- ♦ Dominar entornos de desarrollo multiplataforma y tecnologías emergentes aplicadas al diseño de soluciones digitales
- ♦ Comprender el ciclo completo de vida del *Software* y su impacto en procesos empresariales actuales
- ♦ Evaluar la viabilidad técnica y económica de proyectos de desarrollo tecnológico en distintos entornos profesionales
- ♦ Potenciar la capacidad de innovación aplicada al diseño de productos, servicios y sistemas digitales complejos
- ♦ Fomentar el pensamiento crítico, la adaptabilidad y la toma de decisiones basada en datos objetivos





Objetivos específicos

Módulo 1. Fundamentos de programación

- ◆ Comprender los principios básicos de la programación estructurada y orientada a objetos
- ◆ Identificar y aplicar estructuras de control, variables y operadores en distintos lenguajes
- ◆ Desarrollar algoritmos simples utilizando buenas prácticas de codificación
- ◆ Interpretar errores comunes y depurar programas de forma eficiente

Módulo 2. Estructura de datos

- ◆ Analizar la utilidad de las estructuras de datos en la resolución de problemas computacionales
- ◆ Implementar listas, pilas, colas, árboles y grafos en contextos específicos
- ◆ Evaluar la eficiencia de diferentes estructuras según el uso requerido
- ◆ Aplicar estructuras de datos adecuadas al diseño de algoritmos

Módulo 3. Algoritmia y complejidad

- ◆ Diseñar algoritmos eficientes considerando su costo computacional
- ◆ Comparar el rendimiento de algoritmos mediante notación Big O
- ◆ Aplicar técnicas de análisis para identificar cuellos de botella
- ◆ Optimizar algoritmos para mejorar su comportamiento en grandes volúmenes de datos

Módulo 4. Bases de datos

- ◆ Diseñar modelos relacionales aplicando principios de normalización
- ◆ Implementar bases de datos utilizando sistemas gestores convencionales
- ◆ Elaborar consultas SQL para extraer, insertar y modificar información
- ◆ Comprender los fundamentos de integridad, transacciones y concurrencia

Módulo 5. Bases de datos avanzadas

- ♦ Aplicar técnicas de optimización de consultas en entornos complejos
- ♦ Gestionar bases de datos distribuidas y no relacionales
- ♦ Emplear procedimientos almacenados, *triggers* y vistas en escenarios reales
- ♦ Analizar soluciones de bases de datos orientadas a Big Data

Módulo 6. Diseño avanzado de algoritmos

- ♦ Aplicar técnicas como divide y vencerás, programación dinámica y *backtracking*
- ♦ Resolver problemas complejos mediante algoritmos clásicos y heurísticos
- ♦ Identificar problemas NP-completos y evaluar posibles soluciones
- ♦ Implementar algoritmos eficientes adaptados a necesidades concretas

Módulo 7. Interacción persona ordenador

- ♦ Diseñar interfaces centradas en la experiencia del usuario
- ♦ Aplicar principios de usabilidad y accesibilidad en sistemas interactivos
- ♦ Evaluar prototipos mediante pruebas con usuarios y métricas de interacción
- ♦ Integrar patrones de diseño de interfaz en aplicaciones reales

Módulo 8. Programación avanzada

- ♦ Implementar aplicaciones complejas utilizando técnicas modernas de programación
- ♦ Emplear conceptos como hilos, memoria dinámica y excepciones controladas
- ♦ Desarrollar software modular, reutilizable y escalable
- ♦ Integrar bibliotecas y herramientas externas en proyectos de desarrollo





Módulo 9. Desarrollo de aplicaciones en red

- ♦ Diseñar e implementar aplicaciones cliente-servidor utilizando protocolos estándar
- ♦ Gestionar conexiones, concurrencia y seguridad en entornos distribuidos
- ♦ Aplicar técnicas de serialización y comunicación entre procesos remotos
- ♦ Emplear herramientas para el desarrollo y prueba de servicios en red

Módulo 10. Ingeniería del Software

- ♦ Aplicar metodologías de desarrollo como *Scrum*, *XP* y *DevOps*
- ♦ Gestionar el ciclo de vida del software desde la planificación hasta el mantenimiento
- ♦ Elaborar documentación técnica clara y estructurada para proyectos de software
- ♦ Coordinar equipos y herramientas para el desarrollo colaborativo de sistemas complejos



Destacarás por tu compromiso con la calidad, la seguridad del código y el desarrollo ético del Software”

05

Salidas profesionales

El Desarrollo de Software se ha consolidado como uno de los pilares de la economía digital, impulsando la demanda de perfiles especializados en múltiples sectores. Así, este programa abre la puerta a oportunidades profesionales en áreas como la ingeniería de aplicaciones, la arquitectura de sistemas, la ciberseguridad o el análisis de datos. Además, gracias a la constante evolución tecnológica y la transformación digital global, las competencias adquiridas permiten acceder a roles estratégicos en compañías tecnológicas, startups e instituciones internacionales. Por tanto, se trata de un campo con alta empleabilidad, proyección internacional y capacidad real de impacto en la innovación empresarial.

“

*Impulsa tu perfil profesional
hacia roles estratégicos como
Ingeniero DevOps, Arquitecto de
Software o Analista de Sistemas”*

Perfil del egresado

El perfil del egresado de este programa universitario de TECH se distingue por una combinación sólida de habilidades técnicas, pensamiento analítico y capacidad para adaptarse a entornos tecnológicos cambiantes. Gracias a una visión integral del Desarrollo de Software, está preparado para diseñar, implementar y optimizar soluciones digitales eficientes, escalables y seguras. Además, domina herramientas y lenguajes actuales, trabaja con metodologías ágiles y comprende los principios de la ingeniería del software moderno. Esta versatilidad le permite integrarse con éxito en equipos multidisciplinares, liderar proyectos innovadores y responder con eficacia a los desafíos de un mercado global altamente competitivo.

¿Quieres desempeñarte como Arquitecto de Software? Este plan de estudios te dará las claves para lograrlo en solamente meses.

- ♦ **Pensamiento lógico y estructurado:** Clave para el diseño y optimización de algoritmos complejos
- ♦ **Capacidad de abstracción:** Aptitud para modelar soluciones tecnológicas a partir de problemas reales
- ♦ **Innovación aplicada:** Competencia orientada a la creación de productos digitales con valor diferencial
- ♦ **Responsabilidad digital y ética profesional:** Destreza en el uso y desarrollo de tecnologías de impacto social





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

- 1. Desarrollador de Software:** Diseña, codifica y prueba aplicaciones adaptadas a las necesidades de empresas y usuarios en distintos entornos tecnológicos.
- 2. Ingeniero de Software:** Planifica, construye y mantiene sistemas complejos, aplicando principios de calidad, escalabilidad y seguridad.
- 3. Arquitecto de Software:** Define la estructura técnica de aplicaciones y coordina decisiones clave para garantizar su eficiencia y sostenibilidad.
- 4. Analista de Sistemas:** Evalúa requerimientos, diseña soluciones digitales y actúa como enlace entre equipos técnicos y objetivos de negocio.
- 5. Ingeniero DevOps:** Automatiza procesos de desarrollo e integración continua, optimizando el ciclo de vida del Software.
- 6. Desarrollador Backend:** Se especializa en la lógica interna de sistemas, gestionando bases de datos, servidores y servicios.
- 7. Desarrollador Full Stack:** Domina tanto el desarrollo *frontend* como *backend*, participando en todas las fases de un proyecto digital.
- 8. Especialista en QA y Testing:** Garantiza la calidad del software mediante pruebas funcionales, automatizadas y de rendimiento.



Crearás pruebas automatizadas para la validación de Software en entornos de integración, garantizando una óptima entrega”

06

Metodología de estudio

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

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



“

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

El alumno: la prioridad de todos los programas de TECH

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

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

“

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



Los planes de estudios más exhaustivos a nivel internacional

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

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

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

“

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

Case studies o Método del caso

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

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

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



Método Relearning

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

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

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

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



Un Campus Virtual 100% online con los mejores recursos didácticos

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

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

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

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



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

La eficacia del método se justifica con cuatro logros fundamentales:

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

La metodología universitaria mejor valorada por sus alumnos

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

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

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

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



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



Material de estudio

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

Estos contenidos son aplicados después al formato audiovisual que creará nuestra manera de trabajo online, con las técnicas más novedosas que nos permiten ofrecerte una gran calidad, en cada una de las piezas que pondremos a tu servicio.



Prácticas de habilidades y competencias

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



Resúmenes interactivos

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

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



Lecturas complementarias

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





Case Studies

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



Testing & Retesting

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



Clases magistrales

Existe evidencia científica sobre la utilidad de la observación de terceros expertos. El denominado *Learning from an expert* afianza el conocimiento y el recuerdo, y genera seguridad en nuestras futuras decisiones difíciles.



Guías rápidas de actuación

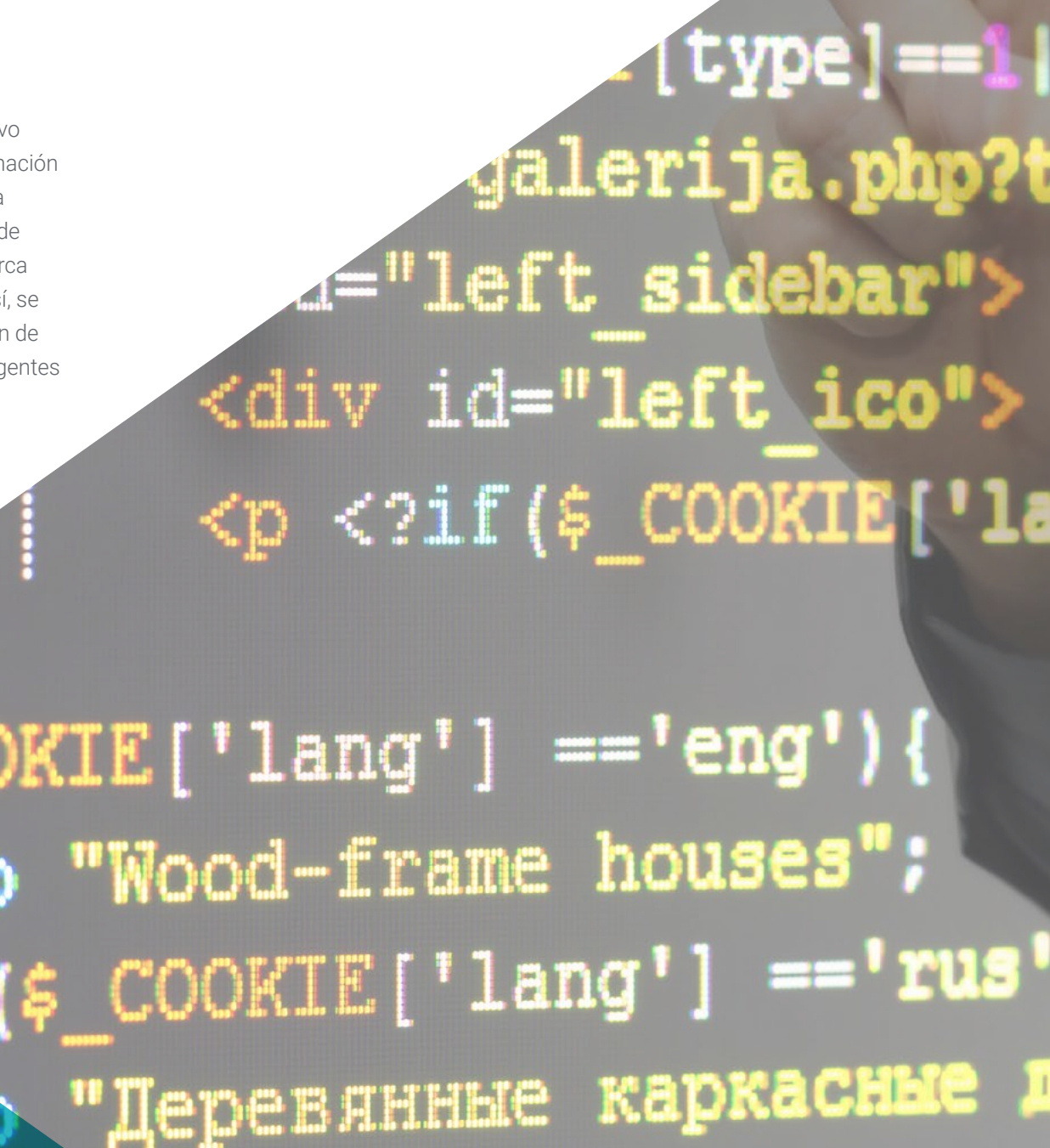
TECH ofrece los contenidos más relevantes del curso en forma de fichas o guías rápidas de actuación. Una manera sintética, práctica y eficaz de ayudar al estudiante a progresar en su aprendizaje.



07

Cuadro docente

El cuadro docente de este programa está conformado por profesionales en activo con experiencia directa en empresas líderes del sector tecnológico. Esta combinación de conocimiento académico y práctica real permite una transmisión actualizada de contenidos, alineada con las exigencias del mercado. Además, la diversidad de especialidades dentro del equipo garantiza un enfoque multidisciplinar, que abarca desde la ingeniería de software hasta la innovación en sistemas distribuidos. Así, se favorece una comprensión integral del entorno digital, permitiendo la adquisición de competencias que responden a los desafíos actuales y a las necesidades emergentes de la industria global.

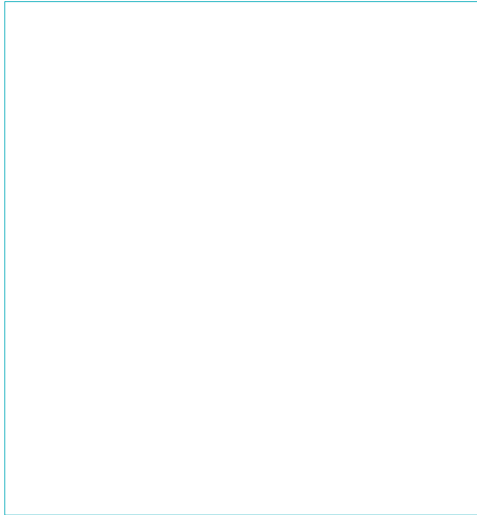




“

Accederás a un plan de estudios diseñado por un reputado equipo docente altamente cualificado en Desarrollo de Software, que te garantizará un aprendizaje eficiente”

Dirección



D. Patil, Amey

- Director Ingeniería de Software de Google en California, Estados Unidos
- Director de Ingeniería en Dell Technologies, California
- Desarrollador de Software en Mealpit, California
- Ingeniero Garantía de Calidad en Amdocs, Chipre
- Especialista en Google Ads y Google Analytics
- Máster en Ciencias de la Ingeniería por Universidad de Maryland
- Licenciatura en Telecomunicaciones por Universidad de Mumbai



08

Titulación

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



“

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

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

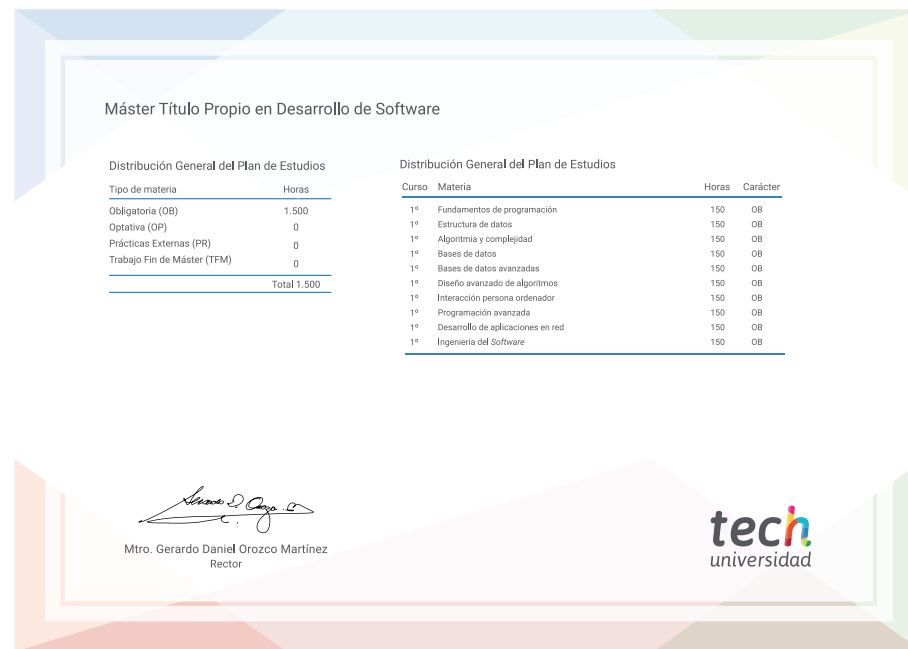
Tras la superación de la evaluación, el alumno recibirá por correo postal* con acuse de recibo su correspondiente título de **Máster Propio** emitido por **TECH Universidad**.

Este título expedido por **TECH Universidad** 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 Desarrollo de Software**

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

Duración: **12 meses**



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



Máster Título Propio Desarrollo de Software

- » Modalidad: No escolarizada (100% en línea)
- » Duración: 12 meses
- » Titulación: TECH Universidad
- » Horario: a tu ritmo
- » Exámenes: online

Máster Título Propio Desarrollo de Software